Package ‘ggstatsplot’

April 14, 2023

Type Package

Title 'ggplot2' Based Plots with Statistical Details

Version 0.11.1

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 syntax 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 the most common types of statistical approaches and tests: parametric, nonparametric, robust, and Bayesian versions of t-test/ANOVA, correlation analyses, contingency table analysis, meta-analysis, and regression analyses. References: Patil (2021) <doi:10.21105/joss.03236>.

License GPL-3 | file LICENSE


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

Depends R (>= 4.1.0)

Imports correlation (>= 0.8.4), datawizard (>= 0.7.1), dplyr (>= 1.1.1), ggplot2 (>= 3.4.2), ggrepel (>= 0.9.3), ggsignif, glue, insight (>= 0.19.1), paletteer, parameters (>= 0.20.3), patchwork, performance (>= 0.10.2), purrr (>= 1.0.1), rlang, stats, statsExpressions (>= 1.5.0), tidyr, utils

Suggests afex, BayesFactor (>= 0.9.12-4.4), gapminder, ggcorrplot (>= 0.1.4), ggside (>= 0.2.2), knitr, lme4, MASS, metaBMA, metafor, metaplus, PMCMRplus, psych, rmarkdown, survival, testthat (>= 3.1.7), tibble, vdiffr (>= 1.0.5), WRS2

VignetteBuilder knitr

Encoding UTF-8

Language en-US
**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**


**Examples**

```r
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 frighteningness and low disgustingness; LFHD: low frighteningness and high disgustingness; HFHD: high frighteningness and low disgustingness; HFHD: high frighteningness and high disgustingness).

Details
This data set, "Bugs", provides the extent to which men and women want to kill arthropods that vary in frighteningness (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

Examples
```
dim(bugs_wide)
head(bugs_wide)
dplyr::glimpse(bugs_wide)
```
**Description**

Wrapper around `patchwork::wrap_plots()` that will return a combined grid of plots with annotations. In case you want to create a grid of plots, it is **highly recommended** that you use `{patchwork}` package directly and not this wrapper around it which is mostly useful with `{ggstatsplot}` plots. It is exported only for backward compatibility.

**Usage**

```r
combine_plots(
  plotlist,
  plotgrid.args = list(),
  annotation.args = list(),
  guides = "collect",
  ...
)
```

**Arguments**

- `plotlist` A list containing `ggplot` objects.
- `plotgrid.args` A list of additional arguments passed to `patchwork::wrap_plots()`, except for guides argument which is already separately specified here.
- `annotation.args` A list of additional arguments passed to `patchwork::plot_annotation()`.
- `guides` A string specifying how guides should be treated in the layout. ‘collect’ will collect guides below to the given nesting level, removing duplicates. ‘keep’ will stop collection at this level and let guides be placed alongside their plot. `auto` will allow guides to be collected if a upper level tries, but place them alongside the plot if not. If you modify default guide "position" with `theme(legend.position=...)` while also collecting guides you must apply that change to the overall `patchwork` (see example).
- `...` Currently ignored.

**Value**

Combined plot with annotation labels

**Examples**

```r
library(ggplot2)

# preparing the first plot
p1 <- ggplot(
  data = subset(iris, iris$Species == "setosa"),
```


# preparing the second plot
p2 <- 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(
  plotlist = list(p1, p2),
  plotgrid.args = list(nrow = 1),
  annotation.args = list(
    tag_levels = "a",
    title = "Dataset: Iris Flower dataset",
    subtitle = "Edgar Anderson collected this data",
    caption = "Note: Only two species of flower are displayed",
    theme = theme(
      plot.subtitle = element_text(size = 20),
      plot.title = element_text(size = 30)
    )
  )
)

---

**extract_stats**

*Extracting data frames or expressions from* `{ggstatsplot}` *plots*

**Description**

Extracting data frames or expressions from `{ggstatsplot}` plots

**Usage**

```r
eXtract_stats(p, ...

extract_subtitle(p)

extract_caption(p)
```

**Arguments**

- `p`: A plot from `{ggstatsplot}` package
- `...`: Ignored
Details

These are convenience functions to extract data frames or expressions with statistical details that are used to create expressions displayed in \{ggstatsplot\} plots as subtitle, caption, etc. Note that all of this analysis is carried out by the \{statsExpressions\} package. And so if you are using these functions only to extract data frames, you are better off using that package.

The only exception is the \texttt{ggcorrmat()} function. But, if a data frame is what you want, you shouldn't be using \texttt{ggcorrmat()} anyway. You can use \texttt{correlation::correlation()} function which provides tidy data frames by default.

Value

A list of tibbles containing summaries of various statistical analyses. The exact details included will depend on the function.

Examples

```r
set.seed(123)

# non-grouped plot
p1 <- ggbetweenstats(mtcars, cyl, mpg)

# grouped plot
p2 <- grouped_ggbarstats(Titanic_full, Survived, Sex, grouping.var = Age)

# extracting expressions -----------------------------
extract_subtitle(p1)
extract_caption(p1)
extract_subtitle(p2)
extract_caption(p2)

# extracting data frames -----------------------------
extract_stats(p1)
extract_stats(p2[[1L]])
extract_stats(p2[[2L]])
```

---

**ggbarstats**

*Stacked bar charts with statistical tests*

**Description**

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

```r
ggbarstats(
  data,
  x,
  y,
  counts = NULL,
  type = "parametric",
  paired = FALSE,
  results.subtitle = TRUE,
  label = "percentage",
  label.args = list(alpha = 1, fill = "white"),
  k = 2L,
  proportion.test = results.subtitle,
  perc.k = 0L,
  bf.message = TRUE,
  ratio = NULL,
  conf.level = 0.95,
  sampling.plan = "indepMulti",
  fixed.margin = "rows",
  prior.concentration = 1,
  title = NULL,
  subtitle = NULL,
  caption = NULL,
  legend.title = NULL,
  xlab = NULL,
  ylab = NULL,
  ggtheme = ggstatsplot::theme_ggstatsplot(),
  package = "RColorBrewer",
  palette = "Dark2",
  ggplot.component = NULL,
  ...
)
```

Arguments

- **data** A data frame (or a tibble) from which variables specified are to be taken. Other data types (e.g., matrix, table, array, etc.) will not be accepted. Additionally, grouped data frames from `{dplyr}` should be ungrouped before they are entered as data.

- **x** The variable to use as the rows in the contingency table. Please note that if there are empty factor levels in your variable, they will be dropped.

- **y**  The variable to use as the columns in the contingency table. Please note that if there are empty factor levels in your variable, they will be dropped. 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** The variable in data containing counts, or NULL if each row represents a single observation.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>type</strong></td>
<td>A character specifying the type of statistical approach:</td>
</tr>
<tr>
<td></td>
<td>• &quot;parametric&quot;</td>
</tr>
<tr>
<td></td>
<td>• &quot;nonparametric&quot;</td>
</tr>
<tr>
<td></td>
<td>• &quot;robust&quot;</td>
</tr>
<tr>
<td></td>
<td>• &quot;bayes&quot;</td>
</tr>
<tr>
<td></td>
<td>You can specify just the initial letter.</td>
</tr>
<tr>
<td><strong>paired</strong></td>
<td>Logical indicating whether data came from a within-subjects or repeated measures design study (Default: FALSE). If TRUE, McNemar’s test expression will be returned. If FALSE, Pearson’s chi-square test will be returned.</td>
</tr>
<tr>
<td><strong>results.subtitle</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>label</strong></td>
<td>Character decides what information needs to be displayed on the label in each pie slice. Possible options are &quot;percentage&quot; (default), &quot;counts&quot;, &quot;both&quot;.</td>
</tr>
<tr>
<td><strong>label.args</strong></td>
<td>Additional aesthetic arguments that will be passed to geom_label.</td>
</tr>
<tr>
<td><strong>k</strong></td>
<td>Number of digits after decimal point (should be an integer) (Default: k = 2L).</td>
</tr>
<tr>
<td><strong>proportion.test</strong></td>
<td>Decides whether proportion test for x variable is to be carried out for each level of y. Defaults to results.subtitle. In ggbarstats, only p-values from this test will be displayed.</td>
</tr>
<tr>
<td><strong>perc.k</strong></td>
<td>Numeric that decides number of decimal places for percentage labels (Default: 0L).</td>
</tr>
<tr>
<td><strong>bf.message</strong></td>
<td>Logical that decides whether to display Bayes Factor in favor of the null hypothesis. This argument is relevant only for parametric test (Default: TRUE).</td>
</tr>
<tr>
<td><strong>ratio</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>conf.level</strong></td>
<td>Scalar between 0 and 1 (default: 95% confidence/credible intervals, 0.95). If NULL, no confidence intervals will be computed.</td>
</tr>
<tr>
<td><strong>sampling.plan</strong></td>
<td>Character describing the sampling plan. Possible options are &quot;indepMulti&quot; (independent multinomial; default), &quot;poisson&quot;, &quot;jointMulti&quot; (joint multinomial), &quot;hypergeom&quot; (hypergeometric). For more, see ?BayesFactor::contingencyTableBF().</td>
</tr>
<tr>
<td><strong>fixed.margin</strong></td>
<td>For the independent multinomial sampling plan, which margin is fixed (&quot;rows&quot; or &quot;cols&quot;). Defaults to &quot;rows&quot;.</td>
</tr>
<tr>
<td><strong>prior.concentration</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>title</strong></td>
<td>The text for the plot title.</td>
</tr>
<tr>
<td><strong>subtitle</strong></td>
<td>The text for the plot subtitle. Will work only if results.subtitle = FALSE.</td>
</tr>
</tbody>
</table>
caption
The text for the plot caption. This argument is relevant only if \texttt{bf.message} = \texttt{FALSE}.

legend.title
Title text for the legend.

xlab
Label for x axis variable. If \texttt{NULL} (default), variable name for x will be used.

ylab
Labels for y axis variable. If \texttt{NULL} (default), variable name for y will be used.

ggtheme
A \texttt{ggplot2} theme. Default value is \texttt{ggstatsplot::theme_ggstatsplot()}. Any of the \texttt{ggplot2} themes (e.g., \texttt{theme_bw()}), or themes from extension packages are allowed (e.g., \texttt{ggthemes::theme_fivethirtyeight()}, \texttt{hrbrthemes::theme_ipsum_ps()}, etc.). But note that sometimes these themes will remove some of the details that \texttt{ggstatsplot} plots typically contains. For example, if relevant, \texttt{ggbetweenstats()} shows details about multiple comparison test as a label on the secondary Y-axis. Some themes (e.g. \texttt{ggthemes::theme_fivethirtyeight()}) will remove the secondary Y-axis and thus the details as well.

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 \texttt{View(paletteer::palettes_d_names)}.

ggplot.component
A \texttt{ggplot} component to be added to the plot prepared by \texttt{ggstatsplot}. This argument is primarily helpful for \texttt{grouped_} variants of all primary functions. Default is \texttt{NULL}. The argument should be entered as a \texttt{ggplot2} function or a list of \texttt{ggplot2} functions.

\ldots
Currently ignored.

Details
For details, see: \url{https://indrajeetpatil.github.io/ggstatsplot/articles/web_only/ggpiestats.html}

Summary of graphics

<table>
<thead>
<tr>
<th>graphical element</th>
<th>geom used</th>
<th>argument for further modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>bars</td>
<td>\texttt{ggplot2::geom_bar()}</td>
<td>NA</td>
</tr>
<tr>
<td>descriptive labels</td>
<td>\texttt{ggplot2::geom_label()}</td>
<td>\texttt{label.args}</td>
</tr>
</tbody>
</table>

Contingency table analyses

The table below provides summary about:

- statistical test carried out for inferential statistics
- type of effect size estimate and a measure of uncertainty for this estimate
- functions used internally to compute these details

two-way table:
Hypothesis testing
### Type Design Test Function used

<table>
<thead>
<tr>
<th>Type</th>
<th>Design</th>
<th>Test</th>
<th>Function used</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parametric/Non-parametric</td>
<td>Unpaired</td>
<td>Pearson’s chi-squared test</td>
<td>stats::chisq.test()</td>
</tr>
<tr>
<td>Bayesian</td>
<td>Unpaired</td>
<td>Bayesian Pearson’s chi-squared test</td>
<td>BayesFactor::contingencyTableBF()</td>
</tr>
<tr>
<td>Parametric/Non-parametric</td>
<td>Paired</td>
<td>McNemar’s chi-squared test</td>
<td>stats::mcnemar.test()</td>
</tr>
<tr>
<td>Bayesian</td>
<td>Paired</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>

### Effect size estimation

<table>
<thead>
<tr>
<th>Type</th>
<th>Design</th>
<th>Effect size</th>
<th>CI available?</th>
<th>Function used</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parametric/Non-parametric</td>
<td>Unpaired</td>
<td>Cramer’s V</td>
<td>Yes</td>
<td>effectsize::cramers_v()</td>
</tr>
<tr>
<td>Bayesian</td>
<td>Unpaired</td>
<td>Cramer’s V</td>
<td>Yes</td>
<td>effectsize::cramers_v()</td>
</tr>
<tr>
<td>Parametric/Non-parametric</td>
<td>Paired</td>
<td>Cohen’s g</td>
<td>Yes</td>
<td>effectsize::cohens_g()</td>
</tr>
<tr>
<td>Bayesian</td>
<td>Paired</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>

### one-way table:

#### Hypothesis testing

<table>
<thead>
<tr>
<th>Type</th>
<th>Test</th>
<th>Function used</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parametric/Non-parametric</td>
<td>Goodness of fit chi-squared test</td>
<td>stats::chisq.test()</td>
</tr>
<tr>
<td>Bayesian</td>
<td>Bayesian Goodness of fit chi-squared test</td>
<td>(custom)</td>
</tr>
</tbody>
</table>

#### Effect size estimation

<table>
<thead>
<tr>
<th>Type</th>
<th>Effect size</th>
<th>CI available?</th>
<th>Function used</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parametric/Non-parametric</td>
<td>Pearson’s C</td>
<td>Yes</td>
<td>effectsize::pearsons_c()</td>
</tr>
<tr>
<td>Bayesian</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>

### See Also

- `grouped_ggbarstats`, `ggpiestats`, `grouped_ggpiestats`

### Examples

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

# creating a plot
p <- ggbarstats(mtcars, x = vs, y = cyl)

# looking at the plot
p

# extracting details from statistical tests
extract_stats(p)
```
**ggbetweenstats**

*Box/Violin plots for between-subjects comparisons*

**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**

```r
# A combination of box and violin plots along with statistical details
# included in the plot as a subtitle.
ggbetweenstats(
  data,  # Required
  x,      # Required
  y,      # Required
  plot.type = "boxviolin",  # default
  type = "parametric",      # default
  pairwise.comparisons = TRUE,  # default
  pairwise.display = "significant",  # default
  p.adjust.method = "holm",  # default
  effsize.type = "unbiased",  # default
  bf.prior = 0.707,  # default
  bf.message = TRUE,  # default
  results.subtitle = TRUE,  # default
  xlab = NULL,  # default
  ylab = NULL,  # default
  caption = NULL,  # default
  title = NULL,  # default
  subtitle = NULL,  # default
  k = 2L,  # default
  var.equal = FALSE,  # default
  conf.level = 0.95,  # default
  nboot = 100L,  # default
  tr = 0.2,  # default
  centrality.plotting = TRUE,  # default
  centrality.type = type,  # default
  centrality.point.args = list(size = 5, color = "darkred"),  # default
  centrality.label.args = list(size = 3, nudge_x = 0.4, segment.linetype = 4, min.segment.length = 0),  # default
  point.args = list(position = ggplot2::position_jitterdodge(dodge.width = 0.6), alpha = 0.4, size = 3, stroke = 0, na.rm = TRUE),  # default
  violin.args = list(width = 0.5, alpha = 0.2, na.rm = TRUE),  # default
  ggsignif.args = list(textsize = 3, tip_length = 0.01, na.rm = TRUE),  # default
  ggtesttheme = ggstatsplot::theme_ggtesttheme(),  # default
  package = "RColorBrewer",  # default
  palette = "Dark2",  # default
  ggplot.component = NULL,  # default
  ...  # default
)```

...
Arguments

A data frame (or a tibble) from which variables specified are to be taken. Other data types (e.g., matrix, table, array, etc.) will not be accepted. Additionally, grouped data frames from {dplyr} should be ungrouped before they are entered as data.

The grouping (or independent) variable from data. In case of a repeated measures or within-subjects design, if subject.id argument is not available or not explicitly specified, the function assumes that the data has already been sorted by such an id by the user and creates an internal identifier. So if your data is not sorted, the results can be inaccurate when there are more than two levels in x and there are NAs present. The data is expected to be sorted by user in subject-1, subject-2, ..., pattern.

The response (or outcome or dependent) variable from data.

Character describing the type of plot. Currently supported plots are "box" (for only boxplots), "violin" (for only violin plots), and "boxviolin" (for a combination of box and violin plots; default).

A character specifying the type of statistical approach:

- "parametric"
- "nonparametric"
- "robust"
- "bayes"

You can specify just the initial letter.

Logical that decides whether pairwise comparisons are to be displayed (default: TRUE). 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 pairwise_comparisons function. For more details about pairwise comparisons, see the documentation for that function.

Decides which pairwise comparisons to display. Available options are:

- "significant" (abbreviation accepted: "s")
- "non-significant" (abbreviation accepted: "ns")
- "all"

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.

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 "eta" (partial eta-squared) or "omega" (partial omega-squared).

**bf.prior**  
A number between 0.5 and 2 (default 0.707), the prior width to use in calculating Bayes factors and posterior estimates. In addition to numeric arguments, several named values are also recognized: "medium", "wide", and "ultrawide", corresponding to $r$ scale values of $1/2$, $\sqrt{2}/2$, and 1, respectively. In case of an ANOVA, this value corresponds to scale for fixed effects.

**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**  
Label for x axis variable. If NULL (default), variable name for x will be used.

**ylab**  
Labels for y axis variable. If NULL (default), variable name for y will be used.

**caption**  
The text for the plot caption. This argument is relevant only if bf.message = FALSE.

**title**  
The text for the plot title.

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

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

**var.equal**  
a logical variable indicating whether to treat the two variances as being equal.  
If TRUE then the pooled variance is used to estimate the variance otherwise the Welch (or Satterthwaite) approximation to the degrees of freedom is used.

**conf.level**  
Scalar between 0 and 1 (default: 95% confidence/credible intervals, 0.95). If NULL, no confidence intervals will be computed.

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

**tr**  
Trim level for the mean when carrying out robust tests. In case of an error, try reducing the value of tr, which is by default set to 0.2. Lowering the value might help.

**centrality.plotting**  
Logical that decides whether centrality tendency measure is to be displayed as a point with a label (Default: TRUE). Function decides which central tendency measure to show depending on the type argument.

- **mean** for parametric statistics
- **median** for non-parametric statistics
- **trimmed mean** for robust statistics
- **MAP estimator** for Bayesian statistics

If you want default centrality parameter, you can specify this using centrality.type argument.

**centrality.type**  
Decides which centrality parameter is to be displayed. The default is to choose the same as type argument. You can specify this to be:  
- "parametric" (for mean)
• "nonparametric" (for **median**)
• robust (for **trimmed mean**)
• bayes (for **MAP estimator**)

Just as type argument, abbreviations are also accepted.

centrality.point.args, centrality.label.args
A list of additional aesthetic arguments to be passed to `geom_point` and `ggrepel::geom_label_repel` geoms, which are involved in mean 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 `{ggplot2}` theme. Default value is `ggstatsplot::theme_ggstatsplot()`. Any of the `{ggplot2}` themes (e.g., `theme_bw()`), or themes from extension packages are allowed (e.g., `ggthemes::theme_fivethirtyeight()`, `hrbrthemes::theme_ipsum_ps()`), etc.). But note that sometimes these themes will remove some of the details that `{ggstatsplot}` plots typically contains. For example, irrelevant, `ggbetweenstats()` shows details about multiple comparison test as a label on the secondary Y-axis. Some themes (e.g. `ggthemes::theme_fivethirtyeight()`) will remove the secondary Y-axis and thus the details as well.

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_ variants of all primary functions. Default is NULL. The argument should be entered as a `{ggplot2}` function or a list of `{ggplot2}` functions.

... Currently ignored.

Details
For details, see: [https://indrajeetpatil.github.io/ggstatsplot/articles/web_only/ggbetweenstats.html](https://indrajeetpatil.github.io/ggstatsplot/articles/web_only/ggbetweenstats.html)

Summary of graphics

<table>
<thead>
<tr>
<th>graphical element</th>
<th>geom used</th>
<th>argument for further modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>raw data</td>
<td>ggplot2::geom_point()</td>
<td>point.args</td>
</tr>
<tr>
<td>box plot</td>
<td>ggplot2::geom_boxplot()</td>
<td>NA</td>
</tr>
<tr>
<td>density plot</td>
<td>ggplot2::geom_violin()</td>
<td>violin.args</td>
</tr>
<tr>
<td>centrality measure point</td>
<td>ggplot2::geom_point()</td>
<td>centrality.point.args</td>
</tr>
<tr>
<td>centrality measure label</td>
<td>ggrepel::geom_label_repel()</td>
<td>centrality.label.args</td>
</tr>
<tr>
<td>pairwise comparisons</td>
<td>ggsignif::geom_signif()</td>
<td>ggsignif.args</td>
</tr>
</tbody>
</table>
# Centrality measures

The table below provides summary about:

- statistical test carried out for inferential statistics
- type of effect size estimate and a measure of uncertainty for this estimate
- functions used internally to compute these details

<table>
<thead>
<tr>
<th>Type</th>
<th>Measure</th>
<th>Function used</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parametric</td>
<td>mean</td>
<td>datawizard::describe_distribution()</td>
</tr>
<tr>
<td>Non-parametric</td>
<td>median</td>
<td>datawizard::describe_distribution()</td>
</tr>
<tr>
<td>Robust</td>
<td>trimmed mean</td>
<td>datawizard::describe_distribution()</td>
</tr>
<tr>
<td>Bayesian</td>
<td>MAP</td>
<td>datawizard::describe_distribution()</td>
</tr>
</tbody>
</table>

# Two-sample tests

The table below provides summary about:

- statistical test carried out for inferential statistics
- type of effect size estimate and a measure of uncertainty for this estimate
- functions used internally to compute these details

### between-subjects:

#### Hypothesis testing

<table>
<thead>
<tr>
<th>Type</th>
<th>No. of groups</th>
<th>Test</th>
<th>Function used</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parametric</td>
<td>2</td>
<td>Student’s or Welch’s $t$-test</td>
<td>stats::t.test()</td>
</tr>
<tr>
<td>Non-parametric</td>
<td>2</td>
<td>Mann-Whitney $U$ test</td>
<td>stats::wilcox.test()</td>
</tr>
<tr>
<td>Robust</td>
<td>2</td>
<td>Yuen’s test for trimmed means</td>
<td>WRS2::yuen()</td>
</tr>
<tr>
<td>Bayesian</td>
<td>2</td>
<td>Student’s $t$-test</td>
<td>BayesFactor::ttestBF()</td>
</tr>
</tbody>
</table>

### Effect size estimation

<table>
<thead>
<tr>
<th>Type</th>
<th>No. of groups</th>
<th>Effect size</th>
<th>CI available?</th>
<th>Function used</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parametric</td>
<td>2</td>
<td>Cohen’s $d$, Hedge’s $g$</td>
<td>Yes</td>
<td>effectsize:</td>
</tr>
<tr>
<td>Non-parametric</td>
<td>2</td>
<td>$r$ (rank-biserial correlation)</td>
<td>Yes</td>
<td>effectsize:</td>
</tr>
<tr>
<td>Robust</td>
<td>2</td>
<td>Algina-Keselman-Penfield robust standardized difference</td>
<td>Yes</td>
<td>WRS2::akp.e</td>
</tr>
<tr>
<td>Bayesian</td>
<td>2</td>
<td>difference</td>
<td>Yes</td>
<td>bayestestR:</td>
</tr>
</tbody>
</table>

### within-subjects:

#### Hypothesis testing

<table>
<thead>
<tr>
<th>Type</th>
<th>No. of groups</th>
<th>Test</th>
<th>Function used</th>
</tr>
</thead>
</table>

---
**Effect size estimation**

<table>
<thead>
<tr>
<th>Type</th>
<th>No. of groups</th>
<th>Effect size</th>
<th>CI available?</th>
<th>Function used</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parametric</td>
<td>2</td>
<td>Cohen's $d$, Hedge's $g$</td>
<td>Yes</td>
<td>effectsize::cohens_d()</td>
</tr>
<tr>
<td>Non-parametric</td>
<td>2</td>
<td>$r$ (rank-biserial correlation)</td>
<td>Yes</td>
<td>effectsize::rank_biserial()</td>
</tr>
<tr>
<td>Robust</td>
<td>2</td>
<td>Algina-Keselman-Penfield robust standardized difference</td>
<td>Yes</td>
<td>WRS2::wmcpAKP()</td>
</tr>
<tr>
<td>Bayesian</td>
<td>2</td>
<td>difference</td>
<td>Yes</td>
<td>bayestestR::effectsize()</td>
</tr>
</tbody>
</table>

**One-way ANOVA**

The table below provides summary about:

- statistical test carried out for inferential statistics
- type of effect size estimate and a measure of uncertainty for this estimate
- functions used internally to compute these details

**between-subjects:**

**Hypothesis testing**

<table>
<thead>
<tr>
<th>Type</th>
<th>No. of groups</th>
<th>Test</th>
<th>Function used</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parametric</td>
<td>&gt; 2</td>
<td>Fisher’s or Welch’s one-way ANOVA</td>
<td>stats::oneway.test()</td>
</tr>
<tr>
<td>Non-parametric</td>
<td>&gt; 2</td>
<td>Kruskal-Wallis one-way ANOVA</td>
<td>stats::kruskal.test()</td>
</tr>
<tr>
<td>Robust</td>
<td>&gt; 2</td>
<td>Heteroscedastic one-way ANOVA for trimmed means</td>
<td>WRS2::t1way()</td>
</tr>
<tr>
<td>Bayes Factor</td>
<td>&gt; 2</td>
<td>Fisher’s ANOVA</td>
<td>BayesFactor::anovaBF()</td>
</tr>
</tbody>
</table>

**Effect size estimation**

<table>
<thead>
<tr>
<th>Type</th>
<th>No. of groups</th>
<th>Effect size</th>
<th>CI available?</th>
<th>Function used</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parametric</td>
<td>&gt; 2</td>
<td>partial eta-squared, partial omega-squared</td>
<td>Yes</td>
<td>effectsize::omega_squared()</td>
</tr>
<tr>
<td>Non-parametric</td>
<td>&gt; 2</td>
<td>rank epsilon squared</td>
<td>Yes</td>
<td>effectsize::rank_epsilon()</td>
</tr>
<tr>
<td>Robust</td>
<td>&gt; 2</td>
<td>Explanatory measure of effect size</td>
<td>Yes</td>
<td>WRS2::t1way()</td>
</tr>
<tr>
<td>Bayes Factor</td>
<td>&gt; 2</td>
<td>Bayesian R-squared</td>
<td>Yes</td>
<td>performance::r2_bayes()</td>
</tr>
</tbody>
</table>

**within-subjects:**

**Hypothesis testing**

<table>
<thead>
<tr>
<th>Type</th>
<th>No. of groups</th>
<th>Test</th>
<th>Function used</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parametric</td>
<td>&gt; 2</td>
<td>One-way repeated measures ANOVA</td>
<td>afex::aov_ez()</td>
</tr>
<tr>
<td>Non-parametric</td>
<td>&gt; 2</td>
<td>Friedman rank sum test</td>
<td>stats::friedman()</td>
</tr>
</tbody>
</table>
Robust > 2 Heteroscedastic one-way repeated measures ANOVA for trimmed means WRS2::rmanova()
Bayes Factor > 2 One-way repeated measures ANOVA BayesFactor::anovaBF()

Effect size estimation

<table>
<thead>
<tr>
<th>Type</th>
<th>No. of groups</th>
<th>Effect size</th>
<th>CI available?</th>
<th>Function used</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parametric</td>
<td>&gt; 2</td>
<td>partial eta-squared, partial omega-squared</td>
<td>Yes</td>
<td>effectsize::omega_squared(), effectsize::eta_squared()</td>
</tr>
<tr>
<td>Non-parametric</td>
<td>&gt; 2</td>
<td>Kendall’s coefficient of concordance</td>
<td>Yes</td>
<td>effectsize::kendalls_w()</td>
</tr>
<tr>
<td>Robust</td>
<td>&gt; 2</td>
<td>Algina-Keselman-Penfield robust standardized difference average</td>
<td>Yes</td>
<td>WRS2::wmcpAKP()</td>
</tr>
<tr>
<td>Bayes Factor</td>
<td>&gt; 2</td>
<td>Bayesian R-squared</td>
<td>Yes</td>
<td>performance::r2_bayes()</td>
</tr>
</tbody>
</table>

Pairwise comparison tests

The table below provides summary about:

- statistical test carried out for inferential statistics
- type of effect size estimate and a measure of uncertainty for this estimate
- functions used internally to compute these details

between-subjects:

Hypothesis testing

<table>
<thead>
<tr>
<th>Type</th>
<th>Equal variance?</th>
<th>Test</th>
<th>p-value adjustment?</th>
<th>Function used</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parametric</td>
<td>No</td>
<td>Games-Howell test</td>
<td>Yes</td>
<td>PMCMRplus::gamesHowellTest()</td>
</tr>
<tr>
<td>Parametric</td>
<td>Yes</td>
<td>Student’s t-test</td>
<td>Yes</td>
<td>stats::pairwise.t.test()</td>
</tr>
<tr>
<td>Non-parametric</td>
<td>No</td>
<td>Dunn test</td>
<td>Yes</td>
<td>PMCMRplus::kwAllPairsDunnTest()</td>
</tr>
<tr>
<td>Robust</td>
<td>No</td>
<td>Yuen’s trimmed means test</td>
<td>Yes</td>
<td>WRS2::lincon()</td>
</tr>
<tr>
<td>Bayesian</td>
<td>NA</td>
<td>Student’s t-test</td>
<td>NA</td>
<td>BayesFactor::ttestBF()</td>
</tr>
</tbody>
</table>

Effect size estimation

Not supported.

within-subjects:

Hypothesis testing

<table>
<thead>
<tr>
<th>Type</th>
<th>Test</th>
<th>p-value adjustment?</th>
<th>Function used</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parametric</td>
<td>Student’s t-test</td>
<td>Yes</td>
<td>stats::pairwise.t.test()</td>
</tr>
<tr>
<td>Non-parametric</td>
<td>Durbin-Conover test</td>
<td>Yes</td>
<td>PMCMRplus::durbinAllPairsTest()</td>
</tr>
<tr>
<td>Robust</td>
<td>Yuen’s trimmed means test</td>
<td>Yes</td>
<td>WRS2::rmmcp()</td>
</tr>
<tr>
<td>Bayesian</td>
<td>Student’s t-test</td>
<td>NA</td>
<td>BayesFactor::ttestBF()</td>
</tr>
</tbody>
</table>

Effect size estimation

Not supported.
ggcoefstats

Dot-and-whisker plots for regression analyses

Description

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

Usage

ggcoefstats(
  x,
  statistic = NULL,
  conf.int = TRUE,
  conf.level = 0.95,
  k = 2L,
  exclude.intercept = FALSE,
  effectsize.type = "eta",
  meta.analytic.effect = FALSE,
Arguments

x A model object to be tidied, or a tidy data frame from a regression model. Function internally uses parameters::model_parameters() to get a tidy data frame. If a data frame, it must contain at least the minimum two columns named term (names of predictors) and estimate (corresponding estimates of coefficients or other quantities of interest).

statistic Relevant statistic for the model ("t", "f", "z", or "chi") in the label. Relevant only if x is a data frame.

conf.int Logical. Decides whether to display confidence intervals as error bars (Default: TRUE).

conf.level Numeric deciding level of confidence or credible intervals (Default: 0.95).

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: FALSE).

effectsize.type This is the same as effectsize_type argument of parameters::model_parameters(). Defaults to "eta", and relevant for ANOVA-like objects.

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.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>meta.type</td>
<td>Type of statistics used to carry out random-effects meta-analysis. If &quot;parametric&quot; (default), metafor::rma function will be used. If &quot;robust&quot;, metaplus::metaplus function will be used. If &quot;bayes&quot;, metaBMA::meta_random function will be used.</td>
</tr>
<tr>
<td>bf.message</td>
<td>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.</td>
</tr>
<tr>
<td>sort</td>
<td>If &quot;none&quot; (default) do not sort, &quot;ascending&quot; sort by increasing coefficient value, or &quot;descending&quot; sort by decreasing coefficient value.</td>
</tr>
<tr>
<td>xlab</td>
<td>Label for x axis variable. If NULL (default), variable name for x will be used.</td>
</tr>
<tr>
<td>ylab</td>
<td>Labels for y axis variable. If NULL (default), variable name for y will be used.</td>
</tr>
<tr>
<td>title</td>
<td>The text for the plot title.</td>
</tr>
<tr>
<td>subtitle</td>
<td>The text for the plot subtitle. The input to this argument will be ignored if meta.analytic.effect is set to TRUE.</td>
</tr>
<tr>
<td>caption</td>
<td>The text for the plot caption. This argument is relevant only if bf.message = FALSE.</td>
</tr>
<tr>
<td>only.significant</td>
<td>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.</td>
</tr>
<tr>
<td>point.args</td>
<td>Additional arguments that will be passed to geom_point geom. Please see documentation for that function to know more about these arguments.</td>
</tr>
<tr>
<td>errorbar.args</td>
<td>Additional arguments that will be passed to geom_errorbarh geom. Please see documentation for that function to know more about these arguments.</td>
</tr>
<tr>
<td>vline</td>
<td>Decides whether to display a vertical line (Default: &quot;TRUE&quot;).</td>
</tr>
<tr>
<td>vline.args</td>
<td>Additional arguments that will be passed to geom_vline geom. Please see documentation for that function to know more about these arguments.</td>
</tr>
<tr>
<td>stats.labels</td>
<td>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).</td>
</tr>
<tr>
<td>stats.label.color</td>
<td>Color for the labels. If set to NULL, colors will be chosen from the specified package (Default: &quot;RColorBrewer&quot;) and palette (Default: &quot;Dark2&quot;).</td>
</tr>
<tr>
<td>stats.label.args</td>
<td>Additional arguments that will be passed to ggrepel::geom_label_repel(). Please see documentation for that function to know more about these arguments.</td>
</tr>
<tr>
<td>package, palette</td>
<td>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).</td>
</tr>
<tr>
<td>gggtheme</td>
<td>A (ggplot2) theme. Default value is ggstatsplot::theme_ggstatsplot(). Any of the (ggplot2) themes (e.g., theme_bw()), or themes from extension packages are allowed (e.g., ggthemes::theme_fivethirtyeight(), hrbrthemes::theme_ipsum_ps(), etc.). But note that sometimes these themes will remove some of the details that ggstatsplot plots typically contains. For example, if relevant, ggbetweenstats() shows details about multiple comparison test as a label on the secondary Y-axis. Some themes (e.g. ggthemes::theme_fivethirtyeight()) will remove the secondary Y-axis and thus the details as well.</td>
</tr>
</tbody>
</table>
Additional arguments to tidying method. For more, see `parameters::model_parameters`.

**Details**

For details, see: [https://indrajeetpatil.github.io/ggstatsplot/articles/web_only/ggcoefstats.html](https://indrajeetpatil.github.io/ggstatsplot/articles/web_only/ggcoefstats.html)

**Summary of graphics**

<table>
<thead>
<tr>
<th>graphical element</th>
<th>geom used</th>
<th>argument for further modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>regression estimate</td>
<td>ggplot2::geom_point()</td>
<td>point.args</td>
</tr>
<tr>
<td>error bars</td>
<td>ggplot2::geom_errorbarh()</td>
<td>errorbar.args</td>
</tr>
<tr>
<td>vertical line</td>
<td>ggplot2::geom_vline()</td>
<td>vline.args</td>
</tr>
<tr>
<td>label with statistical details</td>
<td>ggrepel::geom_label_repel()</td>
<td>stats.label.args</td>
</tr>
</tbody>
</table>

**Random-effects meta-analysis**

The table below provides summary about:

- statistical test carried out for inferential statistics
- type of effect size estimate and a measure of uncertainty for this estimate
- functions used internally to compute these details

**Hypothesis testing and Effect size estimation**

<table>
<thead>
<tr>
<th>Type</th>
<th>Test</th>
<th>CI available?</th>
<th>Function used</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parametric</td>
<td>Pearson’s correlation coefficient</td>
<td>Yes</td>
<td><code>correlation::correlation()</code></td>
</tr>
<tr>
<td>Non-parametric</td>
<td>Spearman’s rank correlation coefficient</td>
<td>Yes</td>
<td><code>correlation::correlation()</code></td>
</tr>
<tr>
<td>Robust</td>
<td>Winsorized Pearson’s correlation coefficient</td>
<td>Yes</td>
<td><code>correlation::correlation()</code></td>
</tr>
<tr>
<td>Bayesian</td>
<td>Bayesian Pearson’s correlation coefficient</td>
<td>Yes</td>
<td><code>correlation::correlation()</code></td>
</tr>
</tbody>
</table>

**Note**

1. In case you want to carry out meta-analysis, you will be asked to install the needed packages (`{metafor}`, `{metaplus}`, or `{metaBMA}`) if they are unavailable.

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

3. Given the rapid pace at which new methods are added to these packages, it is recommended that you install development versions of `{easystats}` packages using the `install_latest()` function from `{easystats}`.

**Examples**
ggcorrmat

Visualization of a correlation matrix

Description

Correlation matrix containing results from pairwise correlation tests. If you want a data frame of (grouped) correlation matrix, use correlation::correlation() instead. It can also do grouped analysis when used with output from dplyr::group_by().

Usage

ggcorrmat(
data, cor.vars = NULL, cor.vars.names = NULL, matrix.type = "upper", type = "parametric", tr = 0.2, partial = FALSE, k = 2L, sig.level = 0.05, conf.level = 0.95, bf.prior = 0.707, p.adjust.method = "holm", pch = "cross", ggcorrplot.args = list(method = "square", outline.color = "black", pch.cex = 14), package = "RColorBrewer", palette = "Dark2", # for reproducibility set.seed(123) library(lme4) # model object mod <- lm(formula = mpg ~ cyl * am, data = mtcars) # creating a plot p <- ggcoefstats(mod) # looking at the plot p # extracting details from statistical tests extract_stats(p) # further arguments can be passed to 'parameters::model_parameters()' ggcoefstats(lmer(Reaction ~ Days + (Days | Subject), sleepstudy), effects = "fixed")
ggcorrmat

```r
colors = c("#E69F00", "white", "#009E73"),
ggtheme = ggstatsplot::theme_ggstatsplot(),
ggplot.component = NULL,
title = NULL,
subtitle = NULL,
caption = NULL,
...)
```

**Arguments**

- `data` A data frame from which variables specified are 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.
- `matrix.type` Character, "upper" (default), "lower", or "full", display full matrix, lower triangular or upper triangular matrix.
- `type` A character specifying the type of statistical approach:
  - "parametric"
  - "nonparametric"
  - "robust"
  - "bayes"
  You can specify just the initial letter.
- `tr` Trim level for the mean when carrying out robust tests. In case of an error, try reducing the value of `tr`, which is by default set to 0.2. Lowering the value might help.
- `partial` Can be `TRUE` for partial correlations. For Bayesian partial correlations, "full" instead of pseudo-Bayesian partial correlations (i.e., Bayesian correlation based on frequentist partialization) are returned.
- `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.
- `conf.level` Scalar between 0 and 1 (default: 95% confidence/credible intervals, 0.95). If `NULL`, no confidence intervals will be computed.
- `bf.prior` A number between 0.5 and 2 (default `0.707`), the prior width to use in calculating Bayes factors and posterior estimates. In addition to numeric arguments, several named values are also recognized: "medium", "wide", and "ultrawide", corresponding to `r` scale values of 1/2, `sqrt(2)/2`, and 1, respectively. In case of an ANOVA, this value corresponds to scale for fixed effects.
- `p.adjust.method` Adjustment method for `p-values` for multiple comparisons. Possible methods are: "holm" (default), "hochberg", "hommel", "bonferroni", "BH", "BY", "fdr", "none".
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: corr, method, p.mat, sig.level, ggtheme, colors, lab, pch, legend.title, digits.

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).

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 ggplot2 theme. Default value is ggstatsplot::theme_ggstatsplot(). Any of the ggplot2 themes (e.g., theme_bw()), or themes from extension packages are allowed (e.g., ggthemes::theme_fivethirtyeight(), hrbrthemes::theme_ipsum_ps(), etc.). But note that sometimes these themes will remove some of the details that ggstatsplot plots typically contains. For example, if relevant, ggbetweenstats() shows details about multiple comparison test as a label on the secondary Y-axis. Some themes (e.g. ggthemes::theme_fivethirtyeight()) will remove the secondary Y-axis and thus the details as well.

ggplot.component
A ggplot component to be added to the plot prepared by ggstatsplot. This argument is primarily helpful for grouped_ variants of all primary functions. Default is NULL. The argument should be entered as a ggplot2 function or a list of ggplot2 functions.

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. This argument is relevant only if bf.message = FALSE.

Details
For details, see: https://indrajeetpatil.github.io/ggstatsplot/articles/web_only/ggcorrmat.html

Summary of graphics

<table>
<thead>
<tr>
<th>graphical element</th>
<th>geom used</th>
<th>argument for further modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>correlation matrix</td>
<td>ggcorrplot::ggcorrplot()</td>
<td>ggcorrplot.args</td>
</tr>
</tbody>
</table>
Correlation analyses

The table below provides summary about:

- statistical test carried out for inferential statistics
- type of effect size estimate and a measure of uncertainty for this estimate
- functions used internally to compute these details

Hypothesis testing and Effect size estimation

<table>
<thead>
<tr>
<th>Type</th>
<th>Test</th>
<th>CI available?</th>
<th>Function used</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parametric</td>
<td>Pearson’s correlation coefficient</td>
<td>Yes</td>
<td>correlation::correlation()</td>
</tr>
<tr>
<td>Non-parametric</td>
<td>Spearman’s rank correlation coefficient</td>
<td>Yes</td>
<td>correlation::correlation()</td>
</tr>
<tr>
<td>Robust</td>
<td>Winsorized Pearson’s correlation coefficient</td>
<td>Yes</td>
<td>correlation::correlation()</td>
</tr>
<tr>
<td>Bayesian</td>
<td>Bayesian Pearson’s correlation coefficient</td>
<td>Yes</td>
<td>correlation::correlation()</td>
</tr>
</tbody>
</table>

See Also
grouped_ggcorrmat ggscatterstats grouped_ggscatterstats

Examples

```r
# for reproducibility
set.seed(123)
library(ggcorrplot) # for plot

ggcormat(iris)
```

**gdotplotstats**

*Dot plot/chart for labeled numeric data.*

Description

A dot chart (as described by William S. Cleveland) with statistical details from one-sample test.

Usage

```r
gdotplotstats(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,
tr = 0.2,
k = 2L,
results.subtitle = TRUE,
point.args = list(color = "black", size = 3, shape = 16),
centrality.plotting = TRUE,
centrality.type = type,
centrality.line.args = list(color = "blue", linwidth = 1, linetype = "dashed"),
ggplot.component = NULL,
ggtheme = ggstatsplot::theme_ggstatsplot(),
...
)

Arguments

data A data frame (or a tibble) from which variables specified are to be taken. Other
data types (e.g., matrix, table, array, etc.) will not be accepted. Additionally, grouped data frames from {dplyr} should be ungrouped before they are entered as data.

x A numeric variable from the data frame data.

y Label or grouping variable.

xlab Label for x axis variable. If NULL (default), variable name for x will be used.

ylab Labels for y axis variable. If NULL (default), variable name for 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. This argument is relevant only if bf.message = FALSE.
type A character specifying the type of statistical approach:
  
  • "parametric"
  • "nonparametric"
  • "robust"
  • "bayes"

You can specify just the initial letter.

test.value A number indicating the true value of the mean (Default: 0).

bf.prior A number between 0.5 and 2 (default 0.707), the prior width to use in calculating Bayes factors and posterior estimates. In addition to numeric arguments, several named values are also recognized: "medium", "wide", and "ultrawide", corresponding to r scale values of 1/2, sqrt(2)/2, and 1, respectively. In case of an ANOVA, this value corresponds to scale for fixed effects.
### 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 (default: 95% confidence/credible intervals, 0.95). If NULL, no confidence intervals will be computed.

### tr
Trim level for the mean when carrying out robust tests. In case of an error, try reducing the value of tr, which is by default set to 0.2. Lowering the value might help.

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

### 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.

### centrality.plotting
Logical that decides whether centrality tendency measure is to be displayed as a point with a label (Default: TRUE). Function decides which central tendency measure to show depending on the type argument.

- **mean** for parametric statistics
- **median** for non-parametric statistics
- **trimmed mean** for robust statistics
- **MAP estimator** for Bayesian statistics

If you want default centrality parameter, you can specify this using centrality.type argument.

### centrality.type
Decides which centrality parameter is to be displayed. The default is to choose the same as type argument. You can specify this to be:

- "parametric" (for mean)
- "nonparametric" (for median)
- robust (for trimmed mean)
- bayes (for MAP estimator)

Just as type argument, abbreviations are also accepted.

### 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.

### ggplot.component
A ggplot component to be added to the plot prepared by `ggstatsplot`. This argument is primarily helpful for grouped variants of all primary functions. Default is NULL. The argument should be entered as a `ggplot2` function or a list of `ggplot2` functions.

### ggtheme
A `ggplot2` theme. Default value is `ggstatsplot::theme_ggstatsplot()`. Any of the `ggplot2` themes (e.g., `theme_bw()`), or themes from extension packages are allowed (e.g., `ggthemes::theme_fivethirtyeight()`, `hrbrthemes::theme_ipsum_ps()`).
etc.). But note that sometimes these themes will remove some of the details that
(ggstatsplot) plots typically contains. For example, if relevant, ggbetweenstats()
shows details about multiple comparison test as a label on the secondary Y-axis.
Some themes (e.g. ggthemes::theme_fivethirtyeight()) will remove
the secondary Y-axis and thus the details as well.

... Currently ignored.

Details

For details, see: https://indrajeetpatil.github.io/ggstatsplot/articles/web_only/ggdotplotstats.html

Summary of graphics

<table>
<thead>
<tr>
<th>graphical element</th>
<th>geom used</th>
<th>argument for further modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>histogram bin</td>
<td>ggplot2::stat_bin()</td>
<td>bin.args</td>
</tr>
<tr>
<td>centrality measure line</td>
<td>ggplot2::geom_vline()</td>
<td>centrality.line.args</td>
</tr>
<tr>
<td>normality curve</td>
<td>ggplot2::stat_function()</td>
<td>normal.curve.args</td>
</tr>
</tbody>
</table>

One-sample tests

The table below provides summary about:

- statistical test carried out for inferential statistics
- type of effect size estimate and a measure of uncertainty for this estimate
- functions used internally to compute these details

Hypothesis testing

<table>
<thead>
<tr>
<th>Type</th>
<th>Test</th>
<th>Function used</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parametric</td>
<td>One-sample Student’s t-test</td>
<td>stats::t.test()</td>
</tr>
<tr>
<td>Non-parametric</td>
<td>One-sample Wilcoxon test</td>
<td>stats::wilcox.test()</td>
</tr>
<tr>
<td>Robust</td>
<td>Bootstrap-t method for one-sample test</td>
<td>WRS2::trimcibt()</td>
</tr>
<tr>
<td>Bayesian</td>
<td>One-sample Student’s t-test</td>
<td>BayesFactor::ttestBF()</td>
</tr>
</tbody>
</table>

Effect size estimation

<table>
<thead>
<tr>
<th>Type</th>
<th>Effect size</th>
<th>CI available?</th>
<th>Function used</th>
</tr>
</thead>
</table>
| Parametric     | Cohen’s $d$, Hedge’s $g$           | Yes           | effectsize::cohens_d(),
effectsize::hedges_g() |
| Non-parametric | $r$ (rank-biserial correlation)    | Yes           | effectsize::rank_biserial()        |
| Robust         | trimmed mean                       | Yes           | WRS2::trimcibt()                   |
| Bayes Factor   | difference                         | Yes           | bayestestR::describe_posterior()   |
See Also

grouped_gghistostats, gghistostats, grouped_ggdotplotstats

Examples

# for reproducibility
set.seed(123)

# creating a plot
p <- ggdotplotstats(
data = ggplot2::mpg,
x = cty,
y = manufacturer,
title = "Fuel economy data",
xlab = "city miles per gallon"
)

# looking at the plot
p

# extracting details from statistical tests
extract_stats(p)

---

**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**

```r
gghistostats(
data,  
x,  
binwidth = NULL,  
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,
```
tr = 0.2,
k = 2L,
ggtheme = ggstatsplot::theme_ggstatsplot(),
results.subtitle = TRUE,
bin.args = list(color = "black", fill = "grey50", alpha = 0.7),
centrality.plotting = TRUE,
centrality.type = type,
centrality.line.args = list(color = "blue", linewidth = 1, linetype = "dashed"),
normal.curve = FALSE,
normal.curve.args = list(linewidth = 2),
ggplot.component = NULL,
... )

Arguments

data A data frame (or a tibble) from which variables specified are to be taken. Other data types (e.g., matrix, table, array, etc.) will not be accepted. Additionally, grouped data frames from \{dplyr\} should be ungrouped before they are entered as data.

x A numeric variable from the data frame 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.

xlab Label for x axis variable. If NULL (default), variable name for x 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. This argument is relevant only if bf.message = FALSE.

type A character specifying the type of statistical approach:
- "parametric"
- "nonparametric"
- "robust"
- "bayes"

You can specify just the initial letter.

test.value A number indicating the true value of the mean (Default: 0).

bf.prior A number between 0.5 and 2 (default 0.707), the prior width to use in calculating Bayes factors and posterior estimates. In addition to numeric arguments, several named values are also recognized: "medium", "wide", and "ultrawide", corresponding to \( r \) scale values of 1/2, sqrt(2)/2, and 1, respectively. In case of an ANOVA, this value corresponds to scale for fixed effects.

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 (default: 95% confidence/credible intervals, 0.95). If NULL, no confidence intervals will be computed.

tr  Trim level for the mean when carrying out robust tests. In case of an error, try reducing the value of tr, which is by default set to 0.2. Lowering the value might help.

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

ggtheme  A ggplot2 theme. Default value is ggstatsplot::theme_ggstatsplot(). Any of the ggplot2 themes (e.g., theme_bw()), or themes from extension packages are allowed (e.g., ggthemes::theme_fivethirtyeight(), hrbrthemes::theme_ipsum_ps(), etc.). But note that sometimes these themes will remove some of the details that ggstatsplot plots typically contains. For example, if relevant, gggbetweenstats() shows details about multiple comparison test as a label on the secondary Y-axis. Some themes (e.g. ggthemes::theme_fivethirtyeight()) will remove the secondary Y-axis and thus the details as well.

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.

bin.args  A list of additional aesthetic arguments to be passed to the stat_bin used to display the bins. Do not specify binwidth argument in this list since it has already been specified using the dedicated argument.

centrality.plotting  Logical that decides whether centrality tendency measure is to be displayed as a point with a label (Default: TRUE). Function decides which central tendency measure to show depending on the type argument.

• mean for parametric statistics
• median for non-parametric statistics
• trimmed mean for robust statistics
• MAP estimator for Bayesian statistics

If you want default centrality parameter, you can specify this using centrality.type argument.

centrality.type  Decides which centrality parameter is to be displayed. The default is to choose the same as type argument. You can specify this to be:

• "parametric" (for mean)
• "nonparametric" (for median)
• robust (for trimmed mean)
• bayes (for MAP estimator)

Just as type argument, abbreviations are also accepted.

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.
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_ variants of all primary functions. Default is NULL. The argument should be entered as a `ggplot2` function or a list of `ggplot2` functions.

... Currently ignored.

Details

For details, see: https://indrajeetpatil.github.io/ggstatsplot/articles/web_only/gghistostats.html

Summary of graphics

<table>
<thead>
<tr>
<th>graphical element</th>
<th>geom used</th>
<th>argument for further modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>histogram bin</td>
<td>ggplot2::stat_bin()</td>
<td>bin.args</td>
</tr>
<tr>
<td>centrality measure line</td>
<td>ggplot2::geom_vline()</td>
<td>centrality.line.args</td>
</tr>
<tr>
<td>normality curve</td>
<td>ggplot2::stat_function()</td>
<td>normal.curve.args</td>
</tr>
</tbody>
</table>

One-sample tests

The table below provides summary about:

- statistical test carried out for inferential statistics
- type of effect size estimate and a measure of uncertainty for this estimate
- functions used internally to compute these details

Hypothesis testing

<table>
<thead>
<tr>
<th>Type</th>
<th>Test</th>
<th>Function used</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parametric</td>
<td>One-sample Student’s t-test</td>
<td><code>stats::t.test()</code></td>
</tr>
<tr>
<td>Non-parametric</td>
<td>One-sample Wilcoxon test</td>
<td><code>stats::wilcox.test()</code></td>
</tr>
<tr>
<td>Robust</td>
<td>Bootstrap-t method for one-sample test</td>
<td><code>WRS2::trimcibt()</code></td>
</tr>
<tr>
<td>Bayesian</td>
<td>One-sample Student’s t-test</td>
<td><code>BayesFactor::ttestBF()</code></td>
</tr>
</tbody>
</table>

Effect size estimation

<table>
<thead>
<tr>
<th>Type</th>
<th>Effect size</th>
<th>CI available?</th>
<th>Function used</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parametric</td>
<td>Cohen’s d, Hedge’s g</td>
<td>Yes</td>
<td><code>effectsize::cohens_d()</code>, <code>effectsize::hedges_g()</code></td>
</tr>
<tr>
<td>Non-parametric</td>
<td>r (rank-biserial correlation)</td>
<td>Yes</td>
<td><code>effectsize::rank_biserial()</code></td>
</tr>
</tbody>
</table>
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,
  x,
  y = NULL,
  counts = NULL,
  type = "parametric",
  paired = FALSE,
  results.subtitle = TRUE,
  label = "percentage",
  label.args = list(direction = "both"),
)

Examples

# for reproducibility
set.seed(123)

# creating a plot
p <- gghistostats(
  data = ToothGrowth,
  x = len,
  xlab = "Tooth length",
  centrality.type = "np"
)

# looking at the plot
p

# extracting details from statistical tests
extract_stats(p)
label.repel = FALSE,
k = 2L,
proportion.test = results.subtitle,
perc.k = 0L,
bf.message = TRUE,
ratio = NULL,
conf.level = 0.95,
sampling.plan = "indepMulti",
fixed.margin = "rows",
prior.concentration = 1,
title = NULL,
subtitle = NULL,
caption = NULL,
legend.title = NULL,
ggtheme = ggstatsplot::theme_ggstatsplot(),
package = "RColorBrewer",
palette = "Dark2",
ggplot.component = NULL,
...)

Arguments

data A data frame (or a tibble) from which variables specified are to be taken. Other
data types (e.g., matrix, table, array, etc.) will not be accepted. Additionally,
grouped data frames from {dplyr} should be ungrouped before they are entered
as data.

x The variable to use as the rows in the contingency table. Please note that if there
are empty factor levels in your variable, they will be dropped.

y The variable to use as the columns in the contingency table. Please note that
if there are empty factor levels in your variable, they will be dropped. 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 The variable in data containing counts, or NULL if each row represents a single
observation.

type A character specifying the type of statistical approach:

• "parametric"
• "nonparametric"
• "robust"
• "bayes"

You can specify just the initial letter.

paired Logical indicating whether data came from a within-subjects or repeated mea-
sures design study (Default: FALSE). If TRUE, McNemar's test expression 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.

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

proportion.test
Decides whether proportion test for x variable is to be carried out for each level of y. Defaults to results.subtitle. In ggbarstats, only p-values from this test will be displayed.

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

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).

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.

conf.level
Scalar between 0 and 1 (default: 95% confidence/credible intervals, 0.95). If NULL, no confidence intervals will be computed.

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. This argument is relevant only if bf.message = FALSE.

legend.title
Title text for the legend.

ggtheme
A {ggplot2} theme. Default value is ggstatsplot::theme_ggstatsplot(). Any of the {ggplot2} themes (e.g., theme_bw()), or themes from extension packages are allowed (e.g., ggthemes::theme_fivethirtyeight(), hrbrthemes::theme_ipsum_ps(), etc.). But note that sometimes these themes will remove some of the details that ggstatsplot plots typically contains. For example, if relevant, ggbetweenstats() shows details about multiple comparison test as a label on the secondary Y-axis.
Some themes (e.g., `ggthemes::theme_fivethirtyeight()`) will remove the secondary Y-axis and thus the details as well.

**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_ variants of all primary functions. Default is `NULL`. The argument should be entered as a `{ggplot2}` function or a list of `{ggplot2}` functions.

... Currently ignored.

**Details**

For details, see: [https://indrajeetpatil.github.io/ggstatsplot/articles/web_only/ggpiestats.html](https://indrajeetpatil.github.io/ggstatsplot/articles/web_only/ggpiestats.html)

**Summary of graphics**

<table>
<thead>
<tr>
<th>graphical element</th>
<th>geom used</th>
<th>argument for further modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>pie slices</td>
<td><code>ggplot2::geom_col()</code></td>
<td>NA</td>
</tr>
<tr>
<td>labels</td>
<td><code>ggplot2::geom_label()</code>/<code>ggrepel::geom_label_repel()</code></td>
<td><code>label.args</code></td>
</tr>
</tbody>
</table>

**Contingency table analyses**

The table below provides summary about:

- statistical test carried out for inferential statistics
- type of effect size estimate and a measure of uncertainty for this estimate
- functions used internally to compute these details

**two-way table:**

**Hypothesis testing**

<table>
<thead>
<tr>
<th>Type</th>
<th>Design</th>
<th>Test</th>
<th>Function used</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parametric/Non-parametric</td>
<td>Unpaired</td>
<td>Pearson’s chi-squared test</td>
<td><code>stats::chisq.test()</code></td>
</tr>
<tr>
<td>Bayesian</td>
<td>Unpaired</td>
<td>Bayesian Pearson’s chi-squared test</td>
<td><code>BayesFactor::contingencyTableBF()</code></td>
</tr>
<tr>
<td>Parametric/Non-parametric</td>
<td>Paired</td>
<td>McNemar’s chi-squared test</td>
<td><code>stats::mcnemar.test()</code></td>
</tr>
<tr>
<td>Bayesian</td>
<td>Paired</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>

**Effect size estimation**

<table>
<thead>
<tr>
<th>Type</th>
<th>Design</th>
<th>Effect size</th>
<th>CI available?</th>
<th>Function used</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parametric/Non-parametric</td>
<td>Unpaired</td>
<td>Cramer’s V</td>
<td>Yes</td>
<td><code>effectsize::cramers_v()</code></td>
</tr>
</tbody>
</table>
### Bayesian Unpaired Cramer’s V

<table>
<thead>
<tr>
<th>Type</th>
<th>Test</th>
<th>CI available?</th>
<th>Function used</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parametric/Non-parametric</td>
<td>Goodness of fit chi-squared test</td>
<td>Yes</td>
<td>stats::chisq.test()</td>
</tr>
<tr>
<td>Bayesian</td>
<td>Bayesian Goodness of fit chi-squared test</td>
<td>No</td>
<td>(custom)</td>
</tr>
</tbody>
</table>

### Effect size estimation

<table>
<thead>
<tr>
<th>Type</th>
<th>Effect size</th>
<th>CI available?</th>
<th>Function used</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parametric/Non-parametric</td>
<td>Pearson’s C</td>
<td>Yes</td>
<td>effectsize::pearsons_c()</td>
</tr>
<tr>
<td>Bayesian</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>

See Also

grouped_ggpiestats, ggbarstats, grouped_ggbarstats

Examples

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

# one sample goodness of fit proportion test
p <- ggpiestats(mtcars, vs)

# looking at the plot
p

# extracting details from statistical tests
extract_stats(p)

# association test (or contingency table analysis)
ggpiestats(mtcars, vs, cyl)
```
Description

Scatterplots from \{ggplot2\} combined with marginal densigram (density + histogram) plots with statistical details.

Usage

```r
ggscatterstats(
  data,
  x,
  y,
  type = "parametric",
  conf.level = 0.95,
  bf.prior = 0.707,
  bf.message = TRUE,
  tr = 0.2,
  k = 2L,
  results.subtitle = TRUE,
  label.var = NULL,
  label.expression = NULL,
  marginal = TRUE,
  point.args = list(size = 3, alpha = 0.4, stroke = 0),
  point.width.jitter = 0,
  point.height.jitter = 0,
  point.label.args = list(size = 3, max.overlaps = 1e+06),
  smooth.line.args = list(linewidth = 1.5, color = "blue", method = "lm", formula = y ~
    x),
  xsidehistogram.args = list(fill = "#009E73", color = "black", na.rm = TRUE),
  ysidehistogram.args = list(fill = "#D55E00", color = "black", na.rm = TRUE),
  xlab = NULL,
  ylab = NULL,
  title = NULL,
  subtitle = NULL,
  caption = NULL,
  ggtheme = ggstatsplot::theme_ggstatsplot(),
  ggplot.component = NULL,
  ...
)
```

Arguments

- **data** A data frame (or a tibble) from which variables specified are to be taken. Other data types (e.g., matrix, table, array, etc.) will **not** be accepted. Additionally, grouped data frames from \{dplyr\} should be ungrouped before they are entered as data.

- **x** The column in data containing the explanatory variable to be plotted on the x-axis.

- **y** The column in data containing the response (outcome) variable to be plotted on the y-axis.
type

A character specifying the type of statistical approach:
- "parametric"
- "nonparametric"
- "robust"
- "bayes"

You can specify just the initial letter.

conf.level

Scalar between 0 and 1 (default: 95% confidence/credible intervals, 0.95). If NULL, no confidence intervals will be computed.

bf.prior

A number between 0.5 and 2 (default 0.707), the prior width to use in calculating Bayes factors and posterior estimates. In addition to numeric arguments, several named values are also recognized: "medium", "wide", and "ultrawide", corresponding to r scale values of 1/2, sqrt(2)/2, and 1, respectively. In case of an ANOVA, this value corresponds to scale for fixed effects.

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).

tr

Trim level for the mean when carrying out robust tests. In case of an error, try reducing the value of tr, which is by default set to 0.2. Lowering the value might help.

k

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

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.var

Variable to use for points labels entered as a symbol (e.g. var1).

label.expression

An expression evaluating to a logical vector that determines the subset of data points to label (e.g. y < 4 & z < 20). While using this argument with purrr::map(), you will have to provide a quoted expression (e.g. quote(y < 4 & z < 20)).

marginal

Decides whether marginal distributions will be plotted on axes using ggside functions. The default is TRUE. The package ggside must already be installed by the user.

point.args

A list of additional aesthetic arguments to be passed to 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.

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 geom_smooth geom used to display the regression line.
xsidehistogram.args, ysidehistogram.args
A list of arguments passed to respective geom_s from the {ggside} package to change the marginal distribution histograms plots.

xlab
Label for x axis variable. If NULL (default), variable name for x will be used.

ylab
Labels for y axis variable. If NULL (default), variable name for 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. This argument is relevant only if bf.message = FALSE.

ggtheme
A {ggplot2} theme. Default value is ggstatsplot::theme_ggstatsplot(). Any of the {ggplot2} themes (e.g., theme_bw()), or themes from extension packages are allowed (e.g., ggthemes::theme_fivethirtyeight(), hrbrthemes::theme_ipsum_ps(), etc.). But note that sometimes these themes will remove some of the details that ggstatsplot plots typically contains. For example, if relevant, ggbetweenstats() shows details about multiple comparison test as a label on the secondary Y-axis. Some themes (e.g., ggthemes::theme_fivethirtyeight()) will remove the secondary Y-axis and thus the details as well.

ggplot.component
A ggplot component to be added to the plot prepared by ggstatsplot. This argument is primarily helpful for grouped_ variants of all primary functions. Default is NULL. The argument should be entered as a {ggplot2} function or a list of {ggplot2} functions.

... Currently ignored.

Details
For details, see: https://indrajeetpatil.github.io/ggstatsplot/articles/web_only/ggscatterstats.html

Summary of graphics

<table>
<thead>
<tr>
<th>graphical element</th>
<th>geom used</th>
<th>argument for further modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>histogram bin</td>
<td>ggplot2::stat_bin()</td>
<td>bin.args</td>
</tr>
<tr>
<td>centrality measure line</td>
<td>ggplot2::geom_vline()</td>
<td>centrality.line.args</td>
</tr>
<tr>
<td>normality curve</td>
<td>ggplot2::stat_function()</td>
<td>normal.curve.args</td>
</tr>
</tbody>
</table>

Correlation analyses
The table below provides summary about:

- statistical test carried out for inferential statistics
- type of effect size estimate and a measure of uncertainty for this estimate
- functions used internally to compute these details
Hypothesis testing and Effect size estimation

<table>
<thead>
<tr>
<th>Type</th>
<th>Test</th>
<th>CI available?</th>
<th>Function used</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parametric</td>
<td>Pearson’s correlation coefficient</td>
<td>Yes</td>
<td>correlation::correlation()</td>
</tr>
<tr>
<td>Non-parametric</td>
<td>Spearman’s rank correlation coefficient</td>
<td>Yes</td>
<td>correlation::correlation()</td>
</tr>
<tr>
<td>Robust</td>
<td>Winsorized Pearson’s correlation coefficient</td>
<td>Yes</td>
<td>correlation::correlation()</td>
</tr>
<tr>
<td>Bayesian</td>
<td>Bayesian Pearson’s correlation coefficient</td>
<td>Yes</td>
<td>correlation::correlation()</td>
</tr>
</tbody>
</table>

Note

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.

See Also

grouped_ggscatterstats, ggcorrmat, grouped_ggcorrmat

Examples

```r
code here
```

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",
  bf.prior = 0.707,
  bf.message = TRUE,
  results.subtitle = TRUE,
  xlab = NULL,
  ylab = NULL,
  caption = NULL,
  title = NULL,
  subtitle = NULL,
  k = 2L,
  conf.level = 0.95,
  nboot = 100L,
  tr = 0.2,
  centrality.plotting = TRUE,
  centrality.type = type,
  centrality.point.args = list(size = 5, color = "darkred"),
  centrality.label.args = list(size = 3, nudge_x = 0.4, segment.linetype = 4),
  centrality.path = TRUE,
  centrality.path.args = list(linewidth = 1, color = "red", alpha = 0.5),
  point.args = list(size = 3, alpha = 0.5, na.rm = TRUE),
  point.path = TRUE,
  point.path.args = list(alpha = 0.5, linetype = "dashed"),
  boxplot.args = list(width = 0.2, alpha = 0.5, na.rm = TRUE),
  violin.args = list(width = 0.5, alpha = 0.2, na.rm = TRUE),
  ggsignif.args = list(textsize = 3, tip_length = 0.01, na.rm = TRUE),
  ggtheme = ggstatsplot::theme_ggstatsplot(),
  package = "RColorBrewer",
  palette = "Dark2",
  ggplot.component = NULL,
  ...
)

Arguments

data
A data frame (or a tibble) from which variables specified are to be taken. Other
data types (e.g., matrix, table, array, etc.) will not be accepted. Additionally,
grouped data frames from {dplyr} should be ungrouped before they are entered
as data.

x
The grouping (or independent) variable from data. In case of a repeated mea-
sures or within-subjects design, if subject.id argument is not available or not explicitly specified, the function assumes that the data has already been sorted by such an id by the user and creates an internal identifier. So if your data is not sorted, the results can be inaccurate when there are more than two levels in x and there are NAs present. The data is expected to be sorted by user in subject-1,subject-2, ..., pattern.

y
The response (or outcome or dependent) variable from data.

type
A character specifying the type of statistical approach:
- "parametric"
- "nonparametric"
- "robust"
- "bayes"

You can specify just the initial letter.

pairwise.comparisons
Logical that decides whether pairwise comparisons are to be displayed (default: TRUE). 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 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")
- "non-significant" (abbreviation accepted: "ns")
- "all"

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 "eta" (partial eta-squared) or "omega" (partial omega-squared).

bf.prior
A number between 0.5 and 2 (default 0.707), the prior width to use in calculating Bayes factors and posterior estimates. In addition to numeric arguments, several named values are also recognized: "medium", "wide", and "ultrawide", corresponding to r scale values of 1/2, sqrt(2)/2, and 1, respectively. In case of an ANOVA, this value corresponds to scale for fixed effects.

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  Label for x axis variable. If NULL (default), variable name for x will be used.

ylab  Labels for y axis variable. If NULL (default), variable name for y will be used.

caption  The text for the plot caption. This argument is relevant only if bf.message = FALSE.

title  The text for the plot title.

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

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

cconf.level  Scalar between 0 and 1 (default: 95% confidence/credible intervals, 0.95). If NULL, no confidence intervals will be computed.

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

tr  Trim level for the mean when carrying out robust tests. In case of an error, try reducing the value of tr, which is by default set to 0.2. Lowering the value might help.

centrality.plotting  Logical that decides whether centrality tendency measure is to be displayed as a point with a label (Default: TRUE). Function decides which central tendency measure to show depending on the type argument.

• mean for parametric statistics
• median for non-parametric statistics
• trimmed mean for robust statistics
• MAP estimator for Bayesian statistics

If you want default centrality parameter, you can specify this using centrality.type argument.

centrality.type  Decides which centrality parameter is to be displayed. The default is to choose the same as type argument. You can specify this to be:

• "parametric" (for mean)
• "nonparametric" (for median)
• robust (for trimmed mean)
• bayes (for MAP estimator)

Just as type argument, abbreviations are also accepted.

centrality.point.args, centrality.label.args  A list of additional aesthetic arguments to be passed to geom_point and ggrepel::geom_label_repel geoms, which are involved in mean plotting.

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

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

- **boxplot.args**: A list of additional aesthetic arguments passed on to `geom_boxplot`.
- **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 `{ggplot2}` theme. Default value is `ggstatsplot::theme_ggstatsplot()`.
  - Any of the `{ggplot2}` themes (e.g., `theme_bw()`), or themes from extension packages are allowed (e.g. `ggthemes::theme_fivethirtyeight()`, `hrbrthemes::theme_ipsum_ps()` etc.). But note that sometimes these themes will remove some of the details that `{ggstatsplot}` plots typically contains. For example, if relevant, `ggbetweenstats()` shows details about multiple comparison test as a label on the secondary Y-axis. Some themes (e.g. `ggthemes::theme_fivethirtyeight()`) will remove the secondary Y-axis and thus the details as well.

- **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 variants of all primary functions. Default is `NULL`. The argument should be entered as a `{ggplot2}` function or a list of `{ggplot2}` functions.
- **...**: Currently ignored.

### Details

For details, see: [https://indrajeetpatil.github.io/ggstatsplot/articles/web_only/ggwithinstats.html](https://indrajeetpatil.github.io/ggstatsplot/articles/web_only/ggwithinstats.html)

### Summary of graphics

<table>
<thead>
<tr>
<th>graphical element</th>
<th>geom used</th>
<th>argument for further modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>raw data</td>
<td>ggplot2::geom_point()</td>
<td>point.args</td>
</tr>
<tr>
<td>point path</td>
<td>ggplot2::geom_path()</td>
<td>point.path.args</td>
</tr>
<tr>
<td>box plot</td>
<td>ggplot2::geom_boxplot()</td>
<td>boxplot.args</td>
</tr>
<tr>
<td>density plot</td>
<td>ggplot2::geom_violin()</td>
<td>violin.args</td>
</tr>
<tr>
<td>centrality measure point</td>
<td>ggplot2::geom_point()</td>
<td>centrality.point.args</td>
</tr>
<tr>
<td>centrality measure point path</td>
<td>ggplot2::geom_path()</td>
<td>centrality.path.args</td>
</tr>
<tr>
<td>centrality measure label</td>
<td>ggrepel::geom_label_repel()</td>
<td>centrality.label.args</td>
</tr>
<tr>
<td>pairwise comparisons</td>
<td>ggsignif::geom_signif()</td>
<td>ggsignif.args</td>
</tr>
</tbody>
</table>
Centrality measures

The table below provides summary about:

- statistical test carried out for inferential statistics
- type of effect size estimate and a measure of uncertainty for this estimate
- functions used internally to compute these details

<table>
<thead>
<tr>
<th>Type</th>
<th>Measure</th>
<th>Function used</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parametric</td>
<td>mean</td>
<td>datawizard::describe_distribution()</td>
</tr>
<tr>
<td>Non-parametric</td>
<td>median</td>
<td>datawizard::describe_distribution()</td>
</tr>
<tr>
<td>Robust</td>
<td>trimmed mean</td>
<td>datawizard::describe_distribution()</td>
</tr>
<tr>
<td>Bayesian</td>
<td>MAP</td>
<td>datawizard::describe_distribution()</td>
</tr>
</tbody>
</table>

Two-sample tests

The table below provides summary about:

- statistical test carried out for inferential statistics
- type of effect size estimate and a measure of uncertainty for this estimate
- functions used internally to compute these details

### between-subjects:
#### Hypothesis testing

<table>
<thead>
<tr>
<th>Type</th>
<th>No. of groups</th>
<th>Test</th>
<th>Function used</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parametric</td>
<td>2</td>
<td>Student’s or Welch’s t-test</td>
<td>stats::t.test()</td>
</tr>
<tr>
<td>Non-parametric</td>
<td>2</td>
<td>Mann-Whitney U test</td>
<td>stats::wilcox.test()</td>
</tr>
<tr>
<td>Robust</td>
<td>2</td>
<td>Yuen’s test for trimmed means</td>
<td>WRS2::yuen()</td>
</tr>
<tr>
<td>Bayesian</td>
<td>2</td>
<td>Student’s t-test</td>
<td>BayesFactor::ttestBF()</td>
</tr>
</tbody>
</table>

#### Effect size estimation

<table>
<thead>
<tr>
<th>Type</th>
<th>No. of groups</th>
<th>Effect size</th>
<th>CI available?</th>
<th>Function used</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parametric</td>
<td>2</td>
<td>Cohen’s d, Hedge’s g</td>
<td>Yes</td>
<td>effectsize::cohens_d()</td>
</tr>
<tr>
<td>Non-parametric</td>
<td>2</td>
<td>r (rank-biserial correlation)</td>
<td>Yes</td>
<td>effectsize::hedges_g()</td>
</tr>
<tr>
<td>Robust</td>
<td>2</td>
<td>Algina-Keselman-Penfield robust standardized difference</td>
<td>Yes</td>
<td>WRS2::akp.e()</td>
</tr>
<tr>
<td>Bayesian</td>
<td>2</td>
<td>difference</td>
<td>Yes</td>
<td>bayestestR::describe_posterior()</td>
</tr>
</tbody>
</table>

### within-subjects:
#### Hypothesis testing

<table>
<thead>
<tr>
<th>Type</th>
<th>No. of groups</th>
<th>Test</th>
<th>Function used</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parametric</td>
<td>2</td>
<td>Student’s t-test</td>
<td>stats::t.test()</td>
</tr>
<tr>
<td>Non-parametric</td>
<td>2</td>
<td>Wilcoxon signed-rank test</td>
<td>stats::wilcox.test()</td>
</tr>
</tbody>
</table>
Robust 2 Yuen’s test on trimmed means for dependent samples  \texttt{WRS2::yuend()}
Bayesian 2 Student’s \( t \)-test  \texttt{BayesFactor::ttestBF()}

**Effect size estimation**

<table>
<thead>
<tr>
<th>Type</th>
<th>No. of groups</th>
<th>Effect size</th>
<th>CI available?</th>
<th>Function used</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parametric</td>
<td>2</td>
<td>Cohen’s ( d ), Hedge’s ( g )</td>
<td>Yes</td>
<td>\texttt{effectsize::cohens_d(), effectsize::hedges_g()}</td>
</tr>
<tr>
<td>Non-parametric</td>
<td>2</td>
<td>( r ) (rank-biserial correlation)</td>
<td>Yes</td>
<td>\texttt{effectsize::cohens_d(), effectsize::hedges_g()}</td>
</tr>
<tr>
<td>Robust</td>
<td>2</td>
<td>Algina-Keselman-Penfield robust standardized difference</td>
<td>Yes</td>
<td>\texttt{WRS2::wmcpAKP()}</td>
</tr>
<tr>
<td>Bayesian</td>
<td>2</td>
<td>difference</td>
<td>Yes</td>
<td>\texttt{BayesFactor::ttestBF()}</td>
</tr>
</tbody>
</table>

**One-way ANOVA**

The table below provides summary about:

- statistical test carried out for inferential statistics
- type of effect size estimate and a measure of uncertainty for this estimate
- functions used internally to compute these details

**between-subjects:**

**Hypothesis testing**

<table>
<thead>
<tr>
<th>Type</th>
<th>No. of groups</th>
<th>Test</th>
<th>Function used</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parametric</td>
<td>&gt; 2</td>
<td>Fisher’s or Welch’s one-way ANOVA</td>
<td>\texttt{stats::oneway.test()}</td>
</tr>
<tr>
<td>Non-parametric</td>
<td>&gt; 2</td>
<td>Kruskal-Wallis one-way ANOVA</td>
<td>\texttt{stats::kruskal.test()}</td>
</tr>
<tr>
<td>Robust</td>
<td>&gt; 2</td>
<td>Heteroscedastic one-way ANOVA for trimmed means</td>
<td>\texttt{WRS2::t1way()}</td>
</tr>
<tr>
<td>Bayes Factor</td>
<td>&gt; 2</td>
<td>Fisher’s ANOVA</td>
<td>\texttt{BayesFactor::anovaBF()}</td>
</tr>
</tbody>
</table>

**Effect size estimation**

<table>
<thead>
<tr>
<th>Type</th>
<th>No. of groups</th>
<th>Effect size</th>
<th>CI available?</th>
<th>Function used</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parametric</td>
<td>&gt; 2</td>
<td>partial eta-squared, partial omega-squared</td>
<td>Yes</td>
<td>\texttt{effectsize::omega_squared(), effectsize::eta_squared()}</td>
</tr>
<tr>
<td>Non-parametric</td>
<td>&gt; 2</td>
<td>rank epsilon squared</td>
<td>Yes</td>
<td>\texttt{effectsize::omega_squared(), effectsize::eta_squared()}</td>
</tr>
<tr>
<td>Robust</td>
<td>&gt; 2</td>
<td>Explanatory measure of effect size</td>
<td>Yes</td>
<td>\texttt{WRS2::t1way()}</td>
</tr>
<tr>
<td>Bayes Factor</td>
<td>&gt; 2</td>
<td>Bayesian R-squared</td>
<td>Yes</td>
<td>\texttt{performance::r2_bayes()}</td>
</tr>
</tbody>
</table>

**within-subjects:**

**Hypothesis testing**

<table>
<thead>
<tr>
<th>Type</th>
<th>No. of groups</th>
<th>Test</th>
<th>Function used</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parametric</td>
<td>&gt; 2</td>
<td>One-way repeated measures ANOVA</td>
<td>\texttt{afex::aov_ez()}</td>
</tr>
<tr>
<td>Non-parametric</td>
<td>&gt; 2</td>
<td>Friedman rank sum test</td>
<td>\texttt{stats::friedman.test()}</td>
</tr>
<tr>
<td>Robust</td>
<td>&gt; 2</td>
<td>Heteroscedastic one-way repeated measures ANOVA for trimmed means</td>
<td>\texttt{WRS2::rmanova()}</td>
</tr>
<tr>
<td>Bayes Factor</td>
<td>&gt; 2</td>
<td>One-way repeated measures ANOVA</td>
<td>\texttt{BayesFactor::anovaBF()}</td>
</tr>
</tbody>
</table>
Effect size estimation

<table>
<thead>
<tr>
<th>Type</th>
<th>No. of groups</th>
<th>Effect size</th>
<th>CI available?</th>
<th>Function used</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parametric</td>
<td>&gt; 2</td>
<td>partial eta-squared, partial omega-squared</td>
<td>Yes</td>
<td>effectsize::omega_squared()</td>
</tr>
<tr>
<td>Non-parametric</td>
<td>&gt; 2</td>
<td>Kendall’s coefficient of concordance</td>
<td>Yes</td>
<td>effectsize::eta_squared()</td>
</tr>
<tr>
<td>Robust</td>
<td>&gt; 2</td>
<td>Algina-Keselman-Penfield robust standardized difference average</td>
<td>Yes</td>
<td>WRS2::wmcpAKP()</td>
</tr>
<tr>
<td>Bayes Factor</td>
<td>&gt; 2</td>
<td>Bayesian R-squared</td>
<td>Yes</td>
<td>performance::r2_bayes()</td>
</tr>
</tbody>
</table>

Pairwise comparison tests

The table below provides summary about:
- statistical test carried out for inferential statistics
- type of effect size estimate and a measure of uncertainty for this estimate
- functions used internally to compute these details

between-subjects:

Hypothesis testing

<table>
<thead>
<tr>
<th>Type</th>
<th>Equal variance?</th>
<th>Test</th>
<th>p-value adjustment?</th>
<th>Function used</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parametric</td>
<td>No</td>
<td>Games-Howell test</td>
<td>Yes</td>
<td>PMCMRplus::gamesHowellTest()</td>
</tr>
<tr>
<td>Parametric</td>
<td>Yes</td>
<td>Student’s t-test</td>
<td>Yes</td>
<td>stats::pairwise.t.test()</td>
</tr>
<tr>
<td>Non-parametric</td>
<td>No</td>
<td>Dunn test</td>
<td>Yes</td>
<td>PMCMRplus::kwAllPairsDunnTest()</td>
</tr>
<tr>
<td>Robust</td>
<td>No</td>
<td>Yuen’s trimmed means test</td>
<td>Yes</td>
<td>WRS2::rmmcp()</td>
</tr>
<tr>
<td>Bayesian</td>
<td>NA</td>
<td>Student’s t-test</td>
<td>NA</td>
<td>BayesFactor::ttestBF()</td>
</tr>
</tbody>
</table>

Effect size estimation
Not supported.

within-subjects:

Hypothesis testing

<table>
<thead>
<tr>
<th>Type</th>
<th>Test</th>
<th>p-value adjustment?</th>
<th>Function used</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parametric</td>
<td>Student’s t-test</td>
<td>Yes</td>
<td>stats::pairwise.t.test()</td>
</tr>
<tr>
<td>Non-parametric</td>
<td>Durbin-Conover test</td>
<td>Yes</td>
<td>PMCMRplus::durbinAllPairsTest()</td>
</tr>
<tr>
<td>Robust</td>
<td>Yuen’s trimmed means test</td>
<td>Yes</td>
<td>WRS2::rmmcp()</td>
</tr>
<tr>
<td>Bayesian</td>
<td>Student’s t-test</td>
<td>NA</td>
<td>BayesFactor::ttestBF()</td>
</tr>
</tbody>
</table>

Effect size estimation
Not supported.

See Also

grouped_ggbetweenstats, ggbetweenstats, grouped_ggwithinstats
Examples

# for reproducibility
set.seed(123)
library(dplyr, warn.conflicts = FALSE)

# create a plot
p <- ggwithinstats(
  data = filter(bugs_long, condition %in% c("HDHF", "HDLF")),
  x = condition,
  y = desire,
  type = "np"
)

# looking at the plot
p

# extracting details from statistical tests
extract_stats(p)

# modifying defaults
ggwithinstats(
  data = bugs_long,
  x = condition,
  y = desire,
  type = "robust"
)

---

**grouped_ggbarstats**  
*Grouped bar 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,
  
  ..., 
  grouping.var, 
  plotgrid.args = list(), 
  annotation.args = list() 
)
Arguments

**data**
A data frame (or a tibble) from which variables specified are to be taken. Other data types (e.g., matrix, table, array, etc.) will **not** be accepted. Additionally, grouped data frames from `{dplyr}` should be ungrouped before they are entered as data.

Arguments passed on to `ggbarstats`

**x** The variable to use as the **rows** in the contingency table. Please note that if there are empty factor levels in your variable, they will be dropped.

**y** The variable to use as the **columns** in the contingency table. Please note that if there are empty factor levels in your variable, they will be dropped. 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.

**proportion.test** Decides whether proportion test for `x` variable is to be carried out for each level of `y`. Defaults to `results.subtitle` in `ggbarstats`, only p-values from this test will be displayed.

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

**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`.

**legend.title** Title text for the legend.

**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. This argument is relevant only if `bf.message = FALSE`.

**ggplot.component** A `ggplot` component to be added to the plot prepared by `{ggstatsplot}`. This argument is primarily helpful for grouped_ variants of all primary functions. Default is `NULL`. The argument should be entered as a `{ggplot2}` function or a list of `{ggplot2}` functions.

**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)`.

**ggtheme** A `{ggplot2}` theme. Default value is `{ggstatsplot}::theme_ggstatsplot()`. Any of the `{ggplot2}` themes (e.g., `theme_bw()`), or themes from extension packages are allowed (e.g., `ggthemes::theme_fivethirtyeight()`, `hrbrthemes::theme_ipsum_ps()`, etc.). But note that sometimes these themes will remove some of the details that `{ggstatsplot}` plots typically
contains. For example, if relevant, `ggbetweenstats()` shows details about multiple comparison test as a label on the secondary Y-axis. Some themes (e.g. ggthemes::theme_fivethirtyeight()) will remove the secondary Y-axis and thus the details as well.

type  A character specifying the type of statistical approach:
  • "parametric"
  • "nonparametric"
  • "robust"
  • "bayes"

You can specify just the initial letter.

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

conf.level  Scalar between 0 and 1 (default: 95% confidence/credible intervals, 0.95). If NULL, no confidence intervals will be computed.

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

counts  The variable in data containing counts, or NULL if each row represents a single observation.

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::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.

xlab  Label for x axis variable. If NULL (default), variable name for x will be used.

ylab  Labels for y axis variable. If NULL (default), variable name for y will be used.

grouping.var  A single grouping variable.

plotgrid.args  A list of additional arguments passed to patchwork::wrap_plots(), except for guides argument which is already separately specified here.

annotation.args  A list of additional arguments passed to patchwork::plot_annotation().

Details

For details, see: https://indrajeetpatil.github.io/ggstatsplot/articles/web_only/ggpiestats.html
grouped_ggbetweenstats

See Also
ggbarstats, ggpiestats, grouped_ggpiestats

Examples

```r
# for reproducibility
set.seed(123)
library(dplyr, warn.conflicts = FALSE)

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

# plot
grouped_ggbarstats(
  data = diamonds_short,
  x = color,
  y = clarity,
  grouping.var = cut,
  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

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

Usage

```
grouped_ggbetweenstats(
  data, ...
  grouping.var,
  plotgrid.args = list(),
  annotation.args = list()
)
```
Arguments

data  A data frame (or a tibble) from which variables specified are to be taken. Other data types (e.g., matrix, table, array, etc.) will not be accepted. Additionally, grouped data frames from \{dplyr\} should be ungrouped before they are entered as data.

Arguments passed on to \texttt{ggbetweenstats}

plot.type  Character describing the type of plot. Currently supported plots are "box" (for only boxplots), "violin" (for only violin plots), and "boxviolin" (for a combination of box and violin plots; default).

xlab  Label for x axis variable. If NULL (default), variable name for x will be used.

ylab  Labels for y axis variable. If NULL (default), variable name for y will be used.

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

p.adjust.method  Adjustment method for \textit{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")
  • "non-significant" (abbreviation accepted: "ns")
  • "all"

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

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

subtitle  The text for the plot subtitle. Will work only if \texttt{results.subtitle} = \texttt{FALSE}.

caption  The text for the plot caption. This argument is relevant only if \texttt{bf.message} = \texttt{FALSE}.

centrality.plotting  Logical that decides whether centrality tendency measure is to be displayed as a point with a label (Default: \texttt{TRUE}). Function decides which central tendency measure to show depending on the type argument.
• **mean** for parametric statistics
• **median** for non-parametric statistics
• **trimmed mean** for robust statistics
• **MAP estimator** for Bayesian statistics

If you want default centrality parameter, you can specify this using `centrality.type` argument.

`centrality.type` Decides which centrality parameter is to be displayed. The default is to choose the same as `type` argument. You can specify this to be:

• "parametric" (for **mean**)  
• "nonparametric" (for **median**)  
• robust (for **trimmed mean**)  
• bayes (for **MAP estimator**)  

Just as `type` argument, abbreviations are also accepted.

`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_ variants of all primary functions. Default is NULL. The argument should be entered as a `{ggplot2}` function or a list of `{ggplot2}` functions.

`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)`.

`centrality.point.args,centrality.label.args` A list of additional aesthetic arguments to be passed to `geom_point` and `ggrepel::geom_label_repel` geoms, which are involved in mean plotting.

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

`ggtheme` A `{ggplot2}` theme. Default value is `ggstatsplot::theme_ggstatsplot()`. Any of the `{ggplot2}` themes (e.g., `theme_bw()`), or themes from extension packages are allowed (e.g., `ggthemes::theme_fivethirtyeight()`, `hrbrthemes::theme_ipsum_ps()`, etc.). But note that sometimes these themes will remove some of the details that `{ggstatsplot}` plots typically contains. For example, if relevant, `ggbetweenstats()` shows details about multiple comparison test as a label on the secondary Y-axis. Some themes (e.g. `ggthemes::theme_fivethirtyeight()`) will remove the secondary Y-axis and thus the details as well.

`x` The grouping (or independent) variable from data. In case of a repeated measures or within-subjects design, if `subject.id` argument is not available or not explicitly specified, the function assumes that the data has already been sorted by such an id by the user and creates an internal identifier. So if your data is not sorted, the results can be inaccurate when there are more than two levels in `x` and there are NAs present. The data is expected to be sorted by user in `subject-1,subject-2,...,pattern`.

`y` The response (or outcome or dependent) variable from data.

`type` A character specifying the type of statistical approach:
• "parametric"
• "nonparametric"
• "robust"
• "bayes"

You can specify just the initial letter.

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

conf.level Scalar between 0 and 1 (default: 95% confidence/credible intervals, \( 0.95 \)). If NULL, no confidence intervals will be computed.

effsize.type Type of effect size needed for parametric tests. The argument can be "eta" (partial eta-squared) or "omega" (partial omega-squared).

var.equal a logical variable indicating whether to treat the two variances as being equal. If TRUE then the pooled variance is used to estimate the variance otherwise the Welch (or Satterthwaite) approximation to the degrees of freedom is used.

bf.prior A number between 0.5 and 2 (default 0.707), the prior width to use in calculating Bayes factors and posterior estimates. In addition to numeric arguments, several named values are also recognized: "medium", "wide", and "ultrawide", corresponding to \( r \) scale values of 1/2, \( \sqrt{2}/2 \), and 1, respectively. In case of an ANOVA, this value corresponds to scale for fixed effects.

tr Trim level for the mean when carrying out robust tests. In case of an error, try reducing the value of \( tr \), which is by default set to 0.2. Lowering the value might help.

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

grouping.var A single grouping variable.

plotgrid.args A list of additional arguments passed to patchwork::wrap_plots(), except for guides argument which is already separately specified here.

annotation.args A list of additional arguments passed to patchwork::plot_annotation().

See Also
ggbetweenstats, ggwithinstats, grouped_ggwithinstats

Examples

# for reproducibility
set.seed(123)
library(PMCMRplus) # for pairwise comparisons
library(dplyr, warn.conflicts = FALSE)
library(ggplot2)

# the most basic function call
grouped_ggbetweenstats(
  data = filter(ggplot2::mpg, drv != "4"),


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,
  ...,
  grouping.var,
  plotgrid.args = list(),
  annotation.args = list()
)

Arguments

data
  A data frame from which variables specified are to be taken.

...  
  Arguments passed on to ggcorrmat
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.

partial Can be TRUE for partial correlations. For Bayesian partial correlations, "full" instead of pseudo-Bayesian partial correlations (i.e., Bayesian correlation based on frequentist partialization) are returned.

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

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.

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 gggcorrplot::ggcorrplot() function. The list should avoid any of the following arguments since they are already internally being used: corr.method, p.mat, sig.level, ggtheme, colors, lab, pch, legend.title, digits.

type A character specifying the type of statistical approach:

- "parametric"
- "nonparametric"
- "robust"
- "bayes"

You can specify just the initial letter.

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

conf.level Scalar between 0 and 1 (default: 95% confidence/credible intervals, 0.95). If NULL, no confidence intervals will be computed.

tr Trim level for the mean when carrying out robust tests. In case of an error, try reducing the value of tr, which is by default set to 0.2. Lowering the value might help.

bf.prior A number between 0.5 and 2 (default 0.707), the prior width to use in calculating Bayes factors and posterior estimates. In addition to numeric arguments, several named values are also recognized: "medium", "wide", and "ultrawide", corresponding to r scale values of 1/2, sqrt(2)/2, and 1, respectively. In case of an ANOVA, this value corresponds to scale for fixed effects.

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

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

caption The text for the plot caption. This argument is relevant only if bf.message = FALSE.
ggplot.component A ggplot component to be added to the plot prepared by \{ggstatsplot\}. This argument is primarily helpful for \texttt{grouped_} variants of all primary functions. Default is NULL. The argument should be entered as a \{ggplot2\} function or a list of \{ggplot2\} functions.

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 \texttt{View(paletted::palettes_d_names)}.

ggtheme A \{ggplot2\} theme. Default value is \texttt{ggstatsplot::theme_ggstatsplot()}. Any of the \{ggplot2\} themes (e.g., \texttt{theme_bw()}) or themes from extension packages are allowed (e.g., \texttt{ggthemes::theme_fivethirtyeight()}, \texttt{hrbrthemes::theme_ipsum_ps()}, etc.). But note that sometimes these themes will remove some of the details that \{ggstatsplot\} plots typically contains. For example, if relevant, \texttt{ggbetweenstats()} shows details about multiple comparison test as a label on the secondary Y-axis. Some themes (e.g. \texttt{ggthemes::theme_fivethirtyeight()}) will remove the secondary Y-axis and thus the details as well.

grouping.var A single grouping variable.

plotgrid.args A list of additional arguments passed to \texttt{patchwork::wrap_plots()}, except for \texttt{guides} argument which is already separately specified here.

annotation.args A list of additional arguments passed to \texttt{patchwork::plot_annotation()}.

Details

For details, see: https://indrajeetpatil.github.io/ggstatsplot/articles/web_only/ggcorrmat.html

See Also

ggcorrmat, ggscatterstats, grouped_ggscatterstats

Examples

# for reproducibility
set.seed(123)
library(ggcorrplot) # for plot

grouped_ggcorrmat(
  data = iris,
  grouping.var = Species,
  type = "robust",
  p.adjust.method = "holm",
  plotgrid.args = list(ncol = 1),
  annotation.args = list(tag_levels = "i")
)
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

```r
grouped_ggdotplotstats(
  data,
  ...,  
  grouping.var,
  plotgrid.args = list(),
  annotation.args = list()
)
```

Arguments

data A data frame (or a tibble) from which variables specified are to be taken. Other data types (e.g., matrix, table, array, etc.) will not be accepted. Additionally, grouped data frames from {dplyr} should be ungrouped before they are entered as data.

... Arguments passed on to ggdotplotstats

y Label or grouping variable.

point.args A list of additional aesthetic arguments passed to geom_point.

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.

x A numeric variable from the data frame data.

type A character specifying the type of statistical approach:

- "parametric"
- "nonparametric"
- "robust"
- "bayes"

You can specify just the initial letter.

test.value A number indicating the true value of the mean (Default: 0).

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

conf.level Scalar between 0 and 1 (default: 95% confidence/credible intervals, 0.95). If NULL, no confidence intervals will be computed.

tr Trim level for the mean when carrying out robust tests. In case of an error, try reducing the value of tr, which is by default set to 0.2. Lowering the value might help.
bf.prior  A number between 0.5 and 2 (default 0.707), the prior width to use in calculating Bayes factors and posterior estimates. In addition to numeric arguments, several named values are also recognized: "medium", "wide", and "ultrawide", corresponding to \( r \) scale values of 1/2, \( \sqrt{2}/2 \), and 1, respectively. In case of an ANOVA, this value corresponds to scale for fixed effects.

effsize.type Type of effect size needed for parametric tests. The argument can be "d" (for Cohen's \( d \)) or "g" (for Hedge's \( g \)).
xlab Label for x axis variable. If NULL (default), variable name for x will be used.

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. This argument is relevant only if bf.message = FALSE.

centrality.plotting Logical that decides whether centrality tendency measure is to be displayed as a point with a label (Default: TRUE). Function decides which central tendency measure to show depending on the type argument.

- mean for parametric statistics
- median for non-parametric statistics
- trimmed mean for robust statistics
- MAP estimator for Bayesian statistics

If you want default centrality parameter, you can specify this using centrality.type argument.

centrality.type Decides which centrality parameter is to be displayed. The default is to choose the same as type argument. You can specify this to be:

- "parametric" (for mean)
- "nonparametric" (for median)
- robust (for trimmed mean)
- bayes (for MAP estimator)

Just as type argument, abbreviations are also accepted.

ggplot.component A ggplot component to be added to the plot prepared by \{ggstatsplot\}. This argument is primarily helpful for grouped_ variants of all primary functions. Default is NULL. The argument should be entered as a \{ggplot2\} function or a list of \{ggplot2\} functions.

ggtheme A \{ggplot2\} theme. Default value is \{ggstatsplot\}::theme_ggstatsplot(). Any of the \{ggplot2\} themes (e.g., theme_bw()), or themes from extension packages are allowed (e.g., ggthemes::theme_fivethirtyeight(), hrbrthemes::theme_ipsum_ps(), etc.). But note that sometimes these themes will remove some of the details that \{ggstatsplot\} plots typically
contains. For example, if relevant, `ggbetweenstats()` shows details about multiple comparison test as a label on the secondary Y-axis. Some themes (e.g. `ggthemes::theme_fivethirtyeight()`) will remove the secondary Y-axis and thus the details as well.

- `ylab` Labels for y axis variable. If NULL (default), variable name for y will be used.

- `grouping.var` A single grouping variable.

- `plotgrid.args` A list of additional arguments passed to `patchwork::wrap_plots()`, except for `guides` argument which is already separately specified here.

- `annotation.args` A list of additional arguments passed to `patchwork::plot_annotation()`.

**Details**

For details, see: https://indrajeetpatil.github.io/ggstatsplot/articles/web_only/ggdotplotstats.html

**See Also**

- `grouped_gghistostats`, `ggdotplotstats`, `gghistostats`

**Examples**

```r
# for reproducibility
set.seed(123)
library(dplyr, warn.conflicts = FALSE)

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

# plot
grouped_ggdotplotstats(
  data = df,
  x = cty,
  y = manufacturer,
  grouping.var = cyl,
  test.value = 15.5
)
```

---

**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

```r
grouped_gghistostats(
  data,
  x,
  grouping.var,
  binwidth = NULL,
  plotgrid.args = list(),
  annotation.args = list(),
  ...
)
```

Arguments

- **data**: A data frame (or a tibble) from which variables specified are to be taken. Other data types (e.g., matrix, table, array, etc.) will not be accepted. Additionally, grouped data frames from `{dplyr}` should be ungrouped before they are entered as `data`.
- **x**: A numeric variable from the data frame `data`.
- **grouping.var**: A single grouping variable.
- **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 `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.
- **plotgrid.args**: A list of additional arguments passed to `patchwork::wrap_plots()`, except for the `guides` argument which is already separately specified here.
- **annotation.args**: A list of additional arguments passed to `patchwork::plot_annotation()`.
- **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.
- **bin.args**: A list of additional aesthetic arguments to be passed to the `stat_bin` used to display the bins. Do not specify `binwidth` argument in this list since it has already been specified using the dedicated argument.
- **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.
- **type**: A character specifying the type of statistical approach:
  - "parametric"
  - "nonparametric"
  - "robust"
  - "bayes"
  You can specify just the initial letter.
- **test.value**: A number indicating the true value of the mean (Default: 0).
k  Number of digits after decimal point (should be an integer) (Default: k = 2L).
conf.level Scalar between 0 and 1 (default: 95% confidence/credible intervals, 0.95). If NULL, no confidence intervals will be computed.
tr  Trim level for the mean when carrying out robust tests. In case of an error, try reducing the value of tr, which is by default set to 0.2. Lowering the value might help.
bf.prior  A number between 0.5 and 2 (default: 0.707), the prior width to use in calculating Bayes factors and posterior estimates. In addition to numeric arguments, several named values are also recognized: "medium", "wide", and "ultrawide", corresponding to $r$ scale values of 1/2, sqrt(2)/2, and 1, respectively. In case of an ANOVA, this value corresponds to scale for fixed effects.
effsize.type  Type of effect size needed for parametric tests. The argument can be "d" (for Cohen’s $d$) or "g" (for Hedge’s $g$).
xlab  Label for x axis variable. If NULL (default), variable name for x will be used.
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. This argument is relevant only if bf.message = FALSE.
centrality.plotting  Logical that decides whether centrality tendency measure is to be displayed as a point with a label (Default: TRUE). Function decides which central tendency measure to show depending on the type argument.
  • mean for parametric statistics
  • median for non-parametric statistics
  • trimmed mean for robust statistics
  • MAP estimator for Bayesian statistics
If you want default centrality parameter, you can specify this using centrality.type argument.
centrality.type  Decides which centrality parameter is to be displayed. The default is to choose the same as type argument. You can specify this to be:
  • "parametric" (for mean)
  • "nonparametric" (for median)
  • robust (for trimmed mean)
  • bayes (for MAP estimator)
Just as type argument, abbreviations are also accepted.
ggplot.component  A ggplot component to be added to the plot prepared by {ggstatsplot}. This argument is primarily helpful for grouped_ variants
of all primary functions. Default is NULL. The argument should be entered as a \{ggplot2\} function or a list of \{ggplot2\} functions.

`ggtheme` A \{ggplot2\} theme. Default value is `ggstatsplot::theme_ggstatsplot()`. Any of the \{ggplot2\} themes (e.g., `theme_bw()`), or themes from extension packages are allowed (e.g., `ggthemes::theme_fivethirtyeight()`, `hrbrthemes::theme_ipsum_ps()`, etc.). But note that sometimes these themes will remove some of the details that \{ggstatsplot\} plots typically contains. For example, if relevant, `ggbetweenstats()` shows details about multiple comparison test as a label on the secondary Y-axis. Some themes (e.g. `ggthemes::theme_fivethirtyeight()`) will remove the secondary Y-axis and thus the details as well.

**Details**

For details, see: [https://indrajeetpatil.github.io/ggstatsplot/articles/web_only/ghistostats.html](https://indrajeetpatil.github.io/ggstatsplot/articles/web_only/ghistostats.html)

**See Also**

`gghistostats, ggdotplotstats, grouped_ggdotplotstats`

**Examples**

# for reproducibility
set.seed(123)

# plot
grouped_gghistostats(
data = iris, 
  x = Sepal.Length, 
test.value = 5, 
grouping.var = Species, 
plotgrid.args = list(nrow = 1), 
annotation.args = list(tag_levels = "i")
)

---

**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,
  ..., 
  grouping.var,
  plotgrid.args = list(),
  annotation.args = list()
)

Arguments

data A data frame (or a tibble) from which variables specified are to be taken. Other data types (e.g., matrix, table, array, etc.) will not be accepted. Additionally, grouped data frames from {dplyr} should be ungrouped before they are entered as data.

... Arguments passed on to ggpiestats

x The variable to use as the rows in the contingency table. Please note that if there are empty factor levels in your variable, they will be dropped.

y The variable to use as the columns in the contingency table. Please note that if there are empty factor levels in your variable, they will be dropped. 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.

proportion.test Decides whether proportion test for x variable is to be carried out for each level of y. Defaults to results.subtitle. In ggbarstats, only p-values from this test will be displayed.

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

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.

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. This argument is relevant only if bf.message = FALSE.
ggplot.component A ggplot component to be added to the plot prepared by `{ggstatsplot}`. This argument is primarily helpful for grouped_ variants of all primary functions. Default is `NULL`. The argument should be entered as a `{ggplot2}` function or a list of `{ggplot2}` functions.

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)`.

ggtheme A `{ggplot2}` theme. Default value is `ggstatsplot::theme_ggstatsplot()`. Any of the `{ggplot2}` themes (e.g., `theme_bw()`), or themes from extension packages are allowed (e.g., `ggthemes::theme_fivethirtyeight()`, `hrbrthemes::theme_ipsum_ps()`, etc.). But note that sometimes these themes will remove some of the details that `{ggstatsplot}` plots typically contains. For example, if relevant, `ggbetweenstats()` shows details about multiple comparison test as a label on the secondary Y-axis. Some themes (e.g. `ggthemes::theme_fivethirtyeight()`) will remove the secondary Y-axis and thus the details as well.

type A character specifying the type of statistical approach:
- "parametric"
- "nonparametric"
- "robust"
- "bayes"

You can specify just the initial letter.

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

conf.level Scalar between 0 and 1 (default: 95% confidence/credible intervals, 0.95). If NULL, no confidence intervals will be computed.

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

counts The variable in data containing counts, or `NULL` if each row represents a single observation.

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::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.

grouping.var A single grouping variable.
grouped_ggscatterstats

plotgrid.args A list of additional arguments passed to patchwork::wrap_plots(), except for guides argument which is already separately specified here.

annotation.args A list of additional arguments passed to patchwork::plot_annotation().

Details

For details, see: https://indrajeetpatil.github.io/ggstatsplot/articles/web_only/ggpiestats.html

See Also

ggbarstats, ggpiestats, grouped_ggbarstats

Examples

set.seed(123)
# grouped one-sample proportion test
grouped_ggpiestats(mtcars, x = cyl, grouping.var = am)

Description

Grouped scatterplots from {ggplot2} combined with marginal distribution plots with statistical details added as a subtitle.

Usage

grouped_ggscatterstats(
  data,
  ..., 
  grouping.var,
  plotgrid.args = list(),
  annotation.args = list()
)

Arguments

data A data frame (or a tibble) from which variables specified are to be taken. Other data types (e.g., matrix, table, array, etc.) will not be accepted. Additionally, grouped data frames from {dplyr} should be ungrouped before they are entered as data.
Arguments passed on to `ggscatterstats`

- `label.var` Variable to use for points labels entered as a symbol (e.g. `var1`).
- `label.expression` An expression evaluating to a logical vector that determines the subset of data points to label (e.g. `y < 4 & z < 20`). While using this argument with `purrr::pmap()`, you will have to provide a quoted expression (e.g. `quote(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.
- `smooth.line.args` A list of additional aesthetic arguments to be passed to `geom_smooth geom` used to display the regression line.
- `point.args` A list of additional aesthetic arguments to be passed to `geom_point geom` used to display the raw data points.
- `marginal` Decides whether marginal distributions will be plotted on axes using `ggside functions`. The default is `TRUE`. The package `ggside` must already be installed by the user.

- `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.

- `xsidehistogram.args`, `ysidehistogram.args` A list of arguments passed to respective `geom`s from the `{ggside}` package to change the marginal distribution histograms plots.

- `x` The column in data containing the explanatory variable to be plotted on the x-axis.

- `y` The column in data containing the response (outcome) variable to be plotted on the y-axis.

- `type` A character specifying the type of statistical approach:
  - "parametric"
  - "nonparametric"
  - "robust"
  - "bayes"

  You can specify just the initial letter.

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

- `conf.level` Scalar between 0 and 1 (default: 95% confidence/credible intervals, 0.95). If NULL, no confidence intervals will be computed.

- `tr` Trim level for the mean when carrying out robust tests. In case of an error, try reducing the value of `tr`, which is by default set to 0.2. Lowering the value might help.

- `bf.prior` A number between 0.5 and 2 (default 0.707), the prior width to use in calculating Bayes factors and posterior estimates. In addition to numeric arguments, several named values are also recognized: "medium", "wide", and "ultrawide", corresponding to r scale values of 1/2, sqrt(2)/2, and 1, respectively. In case of an ANOVA, this value corresponds to scale for fixed effects.
grouped_ggscatterstats

xlab Label for x axis variable. If NULL (default), variable name for x will be used.

ylab Labels for y axis variable. If NULL (default), variable name for y will be used.

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. This argument is relevant only if bf.message = FALSE.

ggplot.component A ggplot component to be added to the plot prepared by ggstatsplot. This argument is primarily helpful for grouped_ variants of all primary functions. Default is NULL. The argument should be entered as a (ggplot2) function or a list of (ggplot2) functions.

ggtheme A (ggplot2) theme. Default value is ggstatsplot::theme_ggstatsplot(). Any of the (ggplot2) themes (e.g., theme_bw()), or themes from extension packages are allowed (e.g., ggthemes::theme_fivethirtyeight(), hrbrthemes::theme_ipsum_ps(), etc.). But note that sometimes these themes will remove some of the details that ggstatsplot plots typically contains. For example, if relevant, ggbetweenstats() shows details about multiple comparison test as a label on the secondary Y-axis. Some themes (e.g., ggthemes::theme_fivethirtyeight()) will remove the secondary Y-axis and thus the details as well.

grouping.var A single grouping variable.

plotgrid.args A list of additional arguments passed to patchwork::wrap_plots(), except for guides argument which is already separately specified here.

annotation.args A list of additional arguments passed to patchwork::plot_annotation().

Details

For details, see: https://indrajeetpatil.github.io/ggstatsplot/articles/web_only/ggscatterstats.html

See Also
ggscatterstats, ggcorrmat, grouped_ggcorrmat

Examples

# to ensure reproducibility
set.seed(123)
library(dplyr, warn.conflicts = FALSE)
library(ggplot2)

# basic function call
grouped_ggscatterstats(
data = filter(movies_long, genre == "Comedy" | genre == "Drama"),
x = length,
y = rating,
type = "robust",
grouping.var = genre,
ggplot.component = list(geom_rug(sides = "b"))
)

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

# labeling without expression
grouped_ggscatterstats(
data = filter(movies_long, rating == 7, genre %in% c("Drama", "Comedy")),
x = budget,
y = length,
grouping.var = genre,
bf.message = FALSE,
label.var = "title",
annotation.args = list(tag_levels = "a")
)

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,
grouped_ggwithinstats

..., 
  grouping.var,
  plotgrid.args = list(),
  annotation.args = list()
)

Arguments

data A data frame (or a tibble) from which variables specified are to be taken. Other
data types (e.g., matrix, table, array, etc.) will not be accepted. Additionally,
grouped data frames from {dplyr} should be ungrouped before they are entered as data.

... Arguments passed on to ggwithinstats

point.path,centrality.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.

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

boxplot.args A list of additional aesthetic arguments passed on to geom_boxplot.

xlab Label for x axis variable. If NULL (default), variable name for x will be used.

ylab Labels for y axis variable. If NULL (default), variable name for y will be used.

pairwise.comparisons Logical that decides whether pairwise comparisons are to be displayed (default: TRUE). 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 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")
• "non-significant" (abbreviation accepted: "ns")
• "all"

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.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. This argument is relevant only if bf.message = FALSE.

centrality.plotting Logical that decides whether centrality tendency measure is to be displayed as a point with a label (Default: TRUE). Function decides which central tendency measure to show depending on the type argument.

- **mean** for parametric statistics
- **median** for non-parametric statistics
- **trimmed mean** for robust statistics
- **MAP estimator** for Bayesian statistics

If you want default centrality parameter, you can specify this using centrality.type argument.

centrality.type Decides which centrality parameter is to be displayed. The default is to choose the same as type argument. You can specify this to be:

- "parametric" (for mean)
- "nonparametric" (for median)
- robust (for trimmed mean)
- bayes (for MAP estimator)

Just as type argument, abbreviations are also accepted.

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 variants of all primary functions. Default is NULL. The argument should be entered as a (ggplot2) function or a list of (ggplot2) functions.

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).

centrality.point.args,centrality.label.args A list of additional aesthetic arguments to be passed to geom_point and ggrepel::geom_label_repel geometries, which are involved in mean plotting.

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

ggtheme A {ggplot2} theme. Default value is ggstatsplot::theme_ggstatsplot(). Any of the {ggplot2} themes (e.g., theme_bw()), or themes from extension packages are allowed (e.g., ggthemes::theme_fivethirtyeight(), hrbrthemes::theme_ipsum_ps(), etc.). But note that sometimes these
themes will remove some of the details that \texttt{ggstatsplot} plots typically contains. For example, if relevant, \texttt{ggbetweenstats()} shows details about multiple comparison test as a label on the secondary Y-axis. Some themes (e.g. \texttt{ggthemes::theme_fivethirtyeight()}) will remove the secondary Y-axis and thus the details as well.

\textbf{x} The grouping (or independent) variable from data. In case of a repeated measures or within-subjects design, if \texttt{subject.id} argument is not available or not explicitly specified, the function assumes that the data has already been sorted by such an id by the user and creates an internal identifier. So if your data is \textbf{not} sorted, the results \textit{can} be inaccurate when there are more than two levels in \textbf{x} and there are NAs present. The data is expected to be sorted by user in subject-1,subject-2, ..., pattern.

\textbf{y} The response (or outcome or dependent) variable from data.

\textbf{type} A character specifying the type of statistical approach:
- "parametric"
- "nonparametric"
- "robust"
- "bayes"

You can specify just the initial letter.

\textbf{k} Number of digits after decimal point (should be an integer) (Default: k = 2L).

\textbf{conf.level} Scalar between 0 and 1 (default: 95\% confidence/credible intervals, 0.95). If NULL, no confidence intervals will be computed.

\textbf{effsize.type} Type of effect size needed for \textit{parametric} tests. The argument can be "eta" (partial eta-squared) or "omega" (partial omega-squared).

\textbf{bf.prior} A number between 0.5 and 2 (default 0.707), the prior width to use in calculating Bayes factors and posterior estimates. In addition to numeric arguments, several named values are also recognized: "medium", "wide", and "ultrawide", corresponding to \(r\) scale values of 1/2, \(\sqrt{2}/2\), and 1, respectively. In case of an ANOVA, this value corresponds to \(r\) scale for fixed effects.

\textbf{tr} Trim level for the mean when carrying out robust tests. In case of an error, try reducing the value of \textbf{tr}, which is by default set to 0.2. Lowering the value might help.

\textbf{nboot} Number of bootstrap samples for computing confidence interval for the effect size (Default: 100L).

\textbf{grouping.var} A single grouping variable.

\textbf{plotgrid.args} A list of additional arguments passed to \texttt{patchwork::wrap_plots()}, except for \texttt{guides} argument which is already separately specified here.

\textbf{annotation.args} A list of additional arguments passed to \texttt{patchwork::plot_annotation()}.

\textbf{See Also}

\texttt{ggwithinstats}, \texttt{ggbetweenstats}, \texttt{grouped_ggbetweenstats}
Examples

```r
# for reproducibility
set.seed(123)
library(dplyr, warn.conflicts = FALSE)
library(ggplot2)

# the most basic function call
grouped_ggwithinstats(
  data = filter(bugs_long, condition %in% c("HDHF", "HDLF")),
  x = condition,
  y = desire,
  grouping.var = gender,
  type = "np",
  # additional modifications for **each** plot using `{ggplot2}` functions
ggplot.component = scale_y_continuous(breaks = seq(0, 10, 1), limits = c(0, 10))
)
```

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*.
- attribute. What attribute is being measured ("Sepal" or "Petal").
- 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.
movies_long

Examples

```r
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/](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.

Source

[https://CRAN.R-project.org/package=ggplot2movies](https://CRAN.R-project.org/package=ggplot2movies)

Examples

```r
dim(movies_long)
head(movies_long)
dplyr::glimpse(movies_long)
```
theme_ggstatsplot  Default theme used in \texttt{ggstatsplot}

Description

Common theme used across all plots generated in \texttt{ggstatsplot} and \textit{assumed} by the author to be aesthetically pleasing to the user/reader. The theme is a wrapper around \texttt{theme_bw()}. All \texttt{ggstatsplot} functions have a \texttt{ggtheme} parameter that let you choose a different theme.

Usage

\begin{verbatim}
theme_ggstatsplot()
\end{verbatim}

Value

A \texttt{ggplot} object with the \texttt{theme_ggstatsplot} theme overlaid.

Examples

\begin{verbatim}
library(ggplot2)

ggplot(mtcars, aes(wt, mpg)) +
  geom_point() +
  theme_ggstatsplot()
\end{verbatim}

Titanic_full  Titanic dataset.

Description

Titanic dataset.

Usage

\texttt{Titanic\_full}

Format

A data frame with 2201 rows and 5 variables

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

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

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

* datasets
  - bugs_long, 3
  - bugs_wide, 4
  - iris_long, 76
  - movies_long, 77
  - Titanic_full, 78

bugs_long, 3
bugs_wide, 4

combine_plots, 5

extract_caption (extract_stats), 6
extract_stats, 6
extract_subtitle (extract_stats), 6

ggbarstats, 7, 39, 52, 54, 69
ggbetweenstats, 13, 50, 55, 57, 75
ggcoefstats, 20
ggcorrmat, 24, 43, 58, 60, 71
ggdotplotstats, 27, 35, 61, 63, 66
gghistostats, 31, 33, 63, 64, 66
ggpiestats, 12, 35, 54, 67, 69
ggscatterstats, 27, 35, 60, 70, 71
ggwithinstats, 20, 43, 57, 73, 75
grouped_ggbarstats, 12, 39, 51, 69
grouped_ggbetweenstats, 20, 50, 54, 75
grouped_ggcorrmat, 27, 43, 58, 71
grouped_ggdotplotstats, 31, 35, 61, 66
grouped_gghistostats, 31, 33, 63, 63
grouped_ggpiestats, 12, 39, 54, 66
grouped_ggscatterstats, 27, 43, 60, 69
grouped_ggwithinstats, 20, 50, 57, 72

iris_long, 76

movies_long, 77

theme(legend.position=...), 5
theme_ggstatsplot, 78
Titanic_full, 78