

# Package: papaja (via r-universe)

August 20, 2024

**Title** Prepare American Psychological Association Journal Articles with R Markdown

**Version** 0.1.2.9000

**Description** Tools to create dynamic, submission-ready manuscripts, which conform to American Psychological Association manuscript guidelines. We provide R Markdown document formats for manuscripts (PDF and Word) and revision letters (PDF). Helper functions facilitate reporting statistical analyses or create publication-ready tables and plots.

**Date** 2023-09-28

**URL** <https://github.com/crsh/papaja>

**BugReports** <https://github.com/crsh/papaja/issues>

**Depends** R (>= 3.6), tinylabels (>= 0.1.0)

**Imports** bookdown (>= 0.9.1), broom (>= 0.7.0), glue (>= 1.4.0), knitr (>= 1.26), methods, rmarkdown (>= 2.4), rmdfiltr (>= 0.1.3), utils, yaml, zip

**Suggests** afex, BayesFactor, beeswarm, boot, car, dplyr (>= 1.1.0), effectsize (>= 0.4.4), emmeans, ggforce, ggplot2, latex2exp, lme4, lmerTest, MASS, MBESS, multcomp, nlme, nnet, R.rsp, skimr, spelling, testthat, VGAM

**SystemRequirements** Rendering the document template requires pandoc (>= 2.0; <https://pandoc.org>) and for PDFs a TeX distribution, such as TinyTeX (>= 0.12; <https://yihui.org/tinytex/>)

**License** MIT + file LICENSE

**Encoding** UTF-8

**RoxygenNote** 7.3.2

**VignetteBuilder** knitr, R.rsp

**Language** en-US

**Roxygen** list(markdown = TRUE)

**Repository** <https://crsh.r-universe.dev>

**RemoteUrl** <https://github.com/crsh/papaja>

**RemoteRef** HEAD

**RemoteSha** c045234d41098721189c04bb7489e6d141c01c75

## Contents

add_equals	3
apa6_pdf	3
apa_barplot	5
apa_beeplot	8
apa_df	12
apa_factorial_plot	13
apa_interval	16
apa_lineplot	18
apa_num	22
apa_p	25
apa_prepare_doc	26
apa_print	26
apa_print.aov	27
apa_print.BFBayesFactor	31
apa_print.emmGrid	35
apa_print.glht	37
apa_print.htest	38
apa_print.list	41
apa_print.lm	43
apa_print.lme	46
apa_print.merMod	47
apa_print.papaja_wsci	49
apa_table	50
apa_violinplot	53
beautify_terms	57
cite_r	58
conf_int	59
fetch_web_refs	59
fetch_zotero_refs	60
generate_author_yaml	61
glue_apa_results	62
hd_int	64
in_paren	65
papaja	65
quote_from_tex	67
remove_comments	67
revision_letter_pdf	68
r_refs	69
sanitize_terms	70
se	71
simple_codebook	71

<i>add_equals</i>	3
<i>sort_terms</i>	73
<i>summary.papaja_wsci</i>	73
<i>theme_apa</i>	74
<i>transmute_df_into_label</i>	75
<i>wsci</i>	76
<i>\$.apa_results_table</i>	77
<b>Index</b>	<b>78</b>

---

<i>add_equals</i>	<i>Add Equals Where Necessary</i>
-------------------	-----------------------------------

---

### Description

This is an internal function that prepends every element of a character vector with an 'equals' sign if the respective element does not contain one of c("=", "<", ">").

### Usage

```
add_equals(x)
```

### Arguments

x                    A character vector.

### Value

Character vector

### Examples

```
add_equals(c("42", "<= 42", "> 42", "= 42"))
```

---

<i>apa6_pdf</i>	<i>APA manuscript (6th edition)</i>
-----------------	-------------------------------------

---

### Description

Template for creating an article according to APA guidelines (6th edition) in PDF or DOCX format.

**Usage**

```

apa6_pdf(
  fig_caption = TRUE,
  number_sections = FALSE,
  toc = FALSE,
  keep_tex = TRUE,
  md_extensions = NULL,
  includes = NULL,
  ...
)

apa6_docx(
  fig_caption = TRUE,
  number_sections = FALSE,
  md_extensions = NULL,
  ...
)

apa6_word(...)

apa6_doc(...)

```

**Arguments**

<code>fig_caption</code>	TRUE to render figures with captions
<code>number_sections</code>	TRUE to number section headings
<code>toc</code>	TRUE to include a table of contents in the output
<code>keep_tex</code>	Keep the intermediate tex file used in the conversion to PDF. Note that this argument does not control whether to keep the auxiliary files (e.g., <code>.aux</code> ) generated by LaTeX when compiling <code>.tex</code> to <code>.pdf</code> . To keep these files, you may set <code>options(tinytex.clean = FALSE)</code> .
<code>md_extensions</code>	Markdown extensions to be added or removed from the default definition of R Markdown. See the <a href="#">rmarkdown_format</a> for additional details.
<code>includes</code>	Named list of additional content to include within the document (typically created using the <a href="#">includes</a> function).
<code>...</code>	Further arguments to pass to <a href="#">pdf_document2</a> or <a href="#">word_document2</a> .

**Details**

When creating PDF documents the YAML option `classoption` is passed to the class options of the LaTeX apa6 document class. In this case, additional options are available. Refer to the apa6 document class [documentation](#) to find out about class options such as paper size or draft watermarks.

Please refer to the [papaja online-manual](#) for additional information on available YAML front matter settings. Note that the available settings for DOCX documents are more limited than for PDF documents.

When creating PDF documents the output device for figures defaults to `c("pdf", "png")`, so that each figure is saved in all four formats at a resolution of 300 dpi.

### Value

R Markdown output format to pass to `rmarkdown::render()`.

### Functions

- `apa6_word()`: Format to create .docx-files. Alias of `apa6_docx`.
- `apa6_doc()`: Format to create .docx-files. Alias of `apa6_docx`.

### See Also

`bookdown::pdf_document2()`, `bookdown::word_document2()`

---

apa\_barplot

*Bar Plots for Factorial Designs that Conform to APA Guidelines*

---

### Description

Create one or more bar plots from a `data.frame` containing data from a factorial design and set APA-friendly defaults.

### Usage

```
apa_barplot(data, ...)  
  
## Default S3 method:  
apa_barplot(  
  data,  
  id,  
  factors = NULL,  
  dv,  
  tendency = mean,  
  dispersion = conf_int,  
  level = 0.95,  
  fun_aggregate = mean,  
  na.rm = TRUE,  
  use = "all.obs",  
  reference = 0,  
  intercept = NULL,  
  args_x_axis = NULL,  
  args_y_axis = NULL,  
  args_title = NULL,  
  args_rect = NULL,  
  args_error_bars = NULL,
```

```

    args_legend = NULL,
    xlab = NULL,
    ylab = NULL,
    main = NULL,
    set_par = TRUE,
    ...
)

## S3 method for class 'afex_aov'
apa_barplot(
  data,
  tendency = mean,
  dispersion = conf_int,
  fun_aggregate = mean,
  ...
)

```

### Arguments

<code>data</code>	A <code>data.frame</code> that contains the data, or an object of class <code>afex_aov</code> .
<code>...</code>	Arguments passed on to <code>apa_factorial_plot</code>
<code>id</code>	Character. Variable name that identifies subjects.
<code>factors</code>	Character. A vector of up to four variable names that is used to stratify the data.
<code>dv</code>	Character. The name of the dependent variable.
<code>tendency</code>	Closure. A function that will be used as measure of central tendency.
<code>dispersion</code>	Closure. A function that will be used to construct error bars (i.e., whiskers). Defaults to <code>conf_int()</code> for 95% between-subjects confidence intervals. See details for more options, especially for within-subjects confidence intervals.
<code>level</code>	Numeric. Defines the width of the interval if confidence intervals are plotted. Defaults to <code>0.95</code> . for 95% confidence intervals. Ignored if <code>dispersion</code> is not a confidence-interval function. See details.
<code>fun_aggregate</code>	Closure. The function that will be used to aggregate observations within subjects and factors before calculating descriptive statistics for each cell of the design. Defaults to <code>mean</code> .
<code>na.rm</code>	Logical. Specifies if missing values are removed. Defaults to <code>TRUE</code> .
<code>use</code>	Character. Specifies a method to exclude cases if there are missing values <i>after</i> aggregating. Possible options are <code>"all.obs"</code> or <code>"complete.obs"</code> .
<code>reference</code>	Numeric. A reference point that determines the y coordinate of the x axis. Useful if there exists a 'nil' value; defaults to <code>0</code> .
<code>intercept</code>	Numeric. Adds a horizontal line at height <code>intercept</code> to the plot. Can be either a single value or a matrix. For the matrix case, multiple lines are drawn, where the dimensions of the matrix determine the number of lines to be drawn.
<code>args_x_axis</code>	An optional list that contains further arguments that may be passed to <code>axis()</code> for customizing the x axis.

args_y_axis	An optional list that contains further arguments that may be passed to <code>axis()</code> for customizing the y axis.
args_title	An optional list that contains further arguments that may be passed to <code>title()</code> .
args_rect	An optional list that contains further arguments that may be passed to <code>rect()</code> .
args_error_bars	An optional list that contains further arguments that may be passed to <code>arrows()</code> .
args_legend	An optional list that contains further arguments that may be passed to <code>legend()</code> .
xlab	Character or expression. Label for x axis.
ylab	Character or expression. Label for y axis.
main	Character or expression. For up to two factors, simply specify the main title. If you stratify the data by more than two factors, either specify a single value that will be added to automatically generated main title, <i>or</i> specify an array of multiple titles, one for each plot area.
set_par	Logical. Determines whether <code>par(mfrow = .)</code> should be set for multi-panel plots.

## Details

The measure of dispersion can be either `conf_int()` for between-subjects confidence intervals, `se()` for standard errors, or any other standard function. For within-subjects confidence intervals, specify `wsci()` or `within_subjects_conf_int()`.

If between- or within-subjects confidence intervals are requested, you can also specify the area of the cumulative distribution function that will be covered. For instance, if you want a 98% confidence interval, specify `level = 0.98`. The default is `level = 0.95` for 95% confidence intervals.

### Customization of plot elements:

`apa_factorial_plot()` and its descendants `apa_barplot()`, `apa_lineplot()`, `apa_beeplot()`, and `apa_violinplot()` are wrapper functions that sequentially call:

- `plot.new()`,
- `plot.window()`,
- `axis()` (once for x axis, once for y axis),
- `title()` for axis labels and titles,
- `rect()` for bars in bar plots,
- `points()` for bee swarms,
- `density()` and `polygon()` for violins,
- `lines()` for lines connecting central tendency points,
- `arrows()` for error bars,
- `points()` for tendency points,
- `legend()` for a legend, and
- `lines()` for intercepts.

These calls can be customized by setting the respective parameters `args_*** = list(...)`.

## Value

A named (nested) list of plot options including raw and derived data. *Note that the structure of the return value is about to change in a forthcoming release of papaja.*

**See Also**

Other plots for factorial designs: [apa\\_beeplot\(\)](#), [apa\\_factorial\\_plot\(\)](#), [apa\\_lineplot\(\)](#), [apa\\_violinplot\(\)](#)

**Examples**

```
apa_barplot(  
  data = npk  
  , id = "block"  
  , dv = "yield"  
  , factors = c("N")  
)  
  
apa_barplot(  
  data = npk  
  , id = "block"  
  , dv = "yield"  
  , factors = c("N", "P")  
)  
  
apa_barplot(  
  data = npk  
  , id = "block"  
  , dv = "yield"  
  , factors = c("N", "P", "K")  
  , ylim = c(0, 80)  
  , level = .34  
  , las = 1  
)
```

---

apa\_beeplot

*Bee-swarm Plots for Factorial Designs that Conform to APA Guidelines*

---

**Description**

Create one or more beeswarm plots from a `data.frame` containing data from a factorial design and set APA-friendly defaults.

**Usage**

```
apa_beeplot(data, ...)
```

```
## Default S3 method:
```

```
apa_beeplot(  
  data,  
  id,  
  factors = NULL,
```



```

    dv,
    tendency = mean,
    dispersion = conf_int,
    level = 0.95,
    fun_aggregate = mean,
    na.rm = TRUE,
    use = "all.obs",
    intercept = NULL,
    args_x_axis = NULL,
    args_y_axis = NULL,
    args_title = NULL,
    args_points = NULL,
    args_swarm = NULL,
    args_error_bars = NULL,
    args_legend = NULL,
    jit = 0.3,
    xlab = NULL,
    ylab = NULL,
    main = NULL,
    set_par = TRUE,
    ...
)

## S3 method for class 'afex_aov'
apa_beeplot(
  data,
  tendency = mean,
  dispersion = conf_int,
  fun_aggregate = mean,
  ...
)

```

### Arguments

<code>data</code>	A data.frame that contains the data, or an object of class <code>afex_aov</code> .
<code>...</code>	Arguments passed on to <a href="#">apa_factorial_plot</a>
<code>id</code>	Character. Variable name that identifies subjects.
<code>factors</code>	Character. A vector of up to four variable names that is used to stratify the data.
<code>dv</code>	Character. The name of the dependent variable.
<code>tendency</code>	Closure. A function that will be used as measure of central tendency.
<code>dispersion</code>	Closure. A function that will be used to construct error bars (i.e., whiskers). Defaults to <a href="#">conf_int()</a> for 95% between-subjects confidence intervals. See details for more options, especially for within-subjects confidence intervals.
<code>level</code>	Numeric. Defines the width of the interval if confidence intervals are plotted. Defaults to 0.95. for 95% confidence intervals. Ignored if <code>dispersion</code> is not a confidence-interval function. See details.

fun_aggregate	Closure. The function that will be used to aggregate observations within subjects and factors before calculating descriptive statistics for each cell of the design. Defaults to mean.
na.rm	Logical. Specifies if missing values are removed. Defaults to TRUE.
use	Character. Specifies a method to exclude cases if there are missing values <i>after</i> aggregating. Possible options are "all.obs" or "complete.obs".
intercept	Numeric. Adds a horizontal line at height intercept to the plot. Can be either a single value or a matrix. For the matrix case, multiple lines are drawn, where the dimensions of the matrix determine the number of lines to be drawn.
args_x_axis	An optional list that contains further arguments that may be passed to <code>axis()</code> for customizing the <i>x</i> axis.
args_y_axis	An optional list that contains further arguments that may be passed to <code>axis()</code> for customizing the <i>y</i> axis.
args_title	An optional list that contains further arguments that may be passed to <code>title()</code> .
args_points	An optional list that contains further arguments that may be passed to <code>points()</code> .
args_swarm	An optional list that contains further arguments to customize the <code>points()</code> of the beeswarm.
args_error_bars	An optional list that contains further arguments that may be passed to <code>arrows()</code> .
args_legend	An optional list that contains further arguments that may be passed to <code>legend()</code> .
jit	Numeric. Determines the amount of horizontal displacement. Defaults to 0.3, defaults to 0.4 if <code>plot = "bars"</code> .
xlab	Character or expression. Label for <i>x</i> axis.
ylab	Character or expression. Label for <i>y</i> axis.
main	Character or expression. For up to two factors, simply specify the main title. If you stratify the data by more than two factors, either specify a single value that will be added to automatically generated main title, <i>or</i> specify an array of multiple titles, one for each plot area.
set_par	Logical. Determines whether <code>par(mfrow = .)</code> should be set for multi-panel plots.

## Details

The measure of dispersion can be either `conf_int()` for between-subjects confidence intervals, `se()` for standard errors, or any other standard function. For within-subjects confidence intervals, specify `wsci()` or `within_subjects_conf_int()`.

If between- or within-subjects confidence intervals are requested, you can also specify the area of the cumulative distribution function that will be covered. For instance, if you want a 98% confidence interval, specify `level = 0.98`. The default is `level = 0.95` for 95% confidence intervals.

### Customization of plot elements:

`apa_factorial_plot()` and its descendants `apa_barplot()`, `apa_lineplot()`, `apa_beeplot()`, and `apa_violinplot()` are wrapper functions that sequentially call:

- `plot.new()`,

- `plot.window()`,
- `axis()` (once for *x* axis, once for *y* axis),
- `title()` for axis labels and titles,
- `rect()` for bars in bar plots,
- `points()` for bee swarms,
- `density()` and `polygon()` for violins,
- `lines()` for lines connecting central tendency points,
- `arrows()` for error bars,
- `points()` for tendency points,
- `legend()` for a legend, and
- `lines()` for intercepts.

These calls can be customized by setting the respective parameters `args_*** = list(...)`.

### Value

A named (nested) list of plot options including raw and derived data. *Note that the structure of the return value is about to change in a forthcoming release of papaja.*

### See Also

Other plots for factorial designs: `apa_barplot()`, `apa_factorial_plot()`, `apa_lineplot()`, `apa_violinplot()`

### Examples

```
apa_beeplot(
  data = npk
  , id = "block"
  , dv = "yield"
  , factors = c("N")
)

apa_beeplot(
  data = npk
  , id = "block"
  , dv = "yield"
  , factors = c("N", "P")
  , args.legend = list(x = "center")
)

apa_beeplot(
  data = npk
  , id = "block"
  , dv = "yield"
  , factors = c("N", "P", "K")
  , ylim = c(0, 80)
  , level = .34
  , las = 1
)
```

---

apa\_df

*Typeset Degrees of Freedom*

---

### Description

This is a function for processing degrees of freedom. It takes care that trailing digits are only printed if non-integer values are given.

### Usage

```
apa_df(x, digits = 2L, big.mark = "", elementwise = TRUE)
```

```
print_df(x, digits = 2L, big.mark = "", elementwise = TRUE)
```

### Arguments

x	Numeric. The degrees of freedom to report.
digits	Integer. The desired number of digits after the decimal point to be used if x contains non-integer values.
big.mark	character; if not empty used as mark between every big.interval decimals <i>before</i> (hence big) the decimal point.
elementwise	Logical. Determines whether the number of trailing digits should be determined for each element of x separately (the default), or for the complete vector x.

### Value

An object of the same class as x with all numeric values converted to character.

### See Also

[apa\\_num\(\)](#), [apa\\_p\(\)](#)

### Examples

```
apa_df(c(1, 1.23151))
```

---

apa\_factorial\_plot      *Plots for Factorial Designs that Conform to APA Guidelines*

---

### Description

Create one or more plots by sequentially calling functions from the **graphics** package. `apa_factorial_plot()` is the workhorse function that is called by `apa_barplot()`, `apa_beeplot()`, and `apa_lineplot()`.

### Usage

```
apa_factorial_plot(data, ...)  
  
## Default S3 method:  
apa_factorial_plot(  
  data,  
  id,  
  factors = NULL,  
  dv,  
  tendency = mean,  
  dispersion = conf_int,  
  level = 0.95,  
  fun_aggregate = mean,  
  na.rm = TRUE,  
  use = "all.obs",  
  reference = 0,  
  intercept = NULL,  
  args_x_axis = NULL,  
  args_y_axis = NULL,  
  args_title = NULL,  
  args_rect = NULL,  
  args_points = NULL,  
  args_lines = NULL,  
  args_swarm = NULL,  
  args_violins = NULL,  
  args_density = NULL,  
  args_errorBars = NULL,  
  args_legend = NULL,  
  plot = NULL,  
  jit = 0.3,  
  xlab = NULL,  
  ylab = NULL,  
  main = NULL,  
  set_par = TRUE,  
  ...  
)  
  
## S3 method for class 'afex_aov'
```

```
apa_factorial_plot(
  data,
  tendency = mean,
  dispersion = conf_int,
  fun_aggregate = mean,
  ...
)
```

### Arguments

<code>data</code>	A <code>data.frame</code> that contains the data, or an object of class <code>afex_aov</code> .
<code>...</code>	Arguments passed on to <code>graphics::plot.window</code>
<code>xlim,ylim</code>	numeric vectors of length 2, giving the x and y coordinates ranges.
<code>log</code>	character; indicating which axes should be in log scale.
<code>asp</code>	numeric, giving the <b>aspect</b> ratio y/x, see 'Details'.
<code>id</code>	Character. Variable name that identifies subjects.
<code>factors</code>	Character. A vector of up to four variable names that is used to stratify the data.
<code>dv</code>	Character. The name of the dependent variable.
<code>tendency</code>	Closure. A function that will be used as measure of central tendency.
<code>dispersion</code>	Closure. A function that will be used to construct error bars (i.e., whiskers). Defaults to <code>conf_int()</code> for 95% between-subjects confidence intervals. See details for more options, especially for within-subjects confidence intervals.
<code>level</code>	Numeric. Defines the width of the interval if confidence intervals are plotted. Defaults to 0.95. for 95% confidence intervals. Ignored if <code>dispersion</code> is not a confidence-interval function. See details.
<code>fun_aggregate</code>	Closure. The function that will be used to aggregate observations within subjects and factors before calculating descriptive statistics for each cell of the design. Defaults to mean.
<code>na.rm</code>	Logical. Specifies if missing values are removed. Defaults to TRUE.
<code>use</code>	Character. Specifies a method to exclude cases if there are missing values <i>after</i> aggregating. Possible options are "all.obs" or "complete.obs".
<code>reference</code>	Numeric. A reference point that determines the y coordinate of the x axis. Useful if there exists a 'nil' value; defaults to 0.
<code>intercept</code>	Numeric. Adds a horizontal line at height <code>intercept</code> to the plot. Can be either a single value or a matrix. For the matrix case, multiple lines are drawn, where the dimensions of the matrix determine the number of lines to be drawn.
<code>args_x_axis</code>	An optional list that contains further arguments that may be passed to <code>axis()</code> for customizing the x axis.
<code>args_y_axis</code>	An optional list that contains further arguments that may be passed to <code>axis()</code> for customizing the y axis.
<code>args_title</code>	An optional list that contains further arguments that may be passed to <code>title()</code> .
<code>args_rect</code>	An optional list that contains further arguments that may be passed to <code>rect()</code> .
<code>args_points</code>	An optional list that contains further arguments that may be passed to <code>points()</code> .

args_lines	An optional list that contains further arguments that may be passed to <code>lines()</code> .
args_swarm	An optional list that contains further arguments to customize the <code>points()</code> of the beeswarm.
args_violins	An optional list that contains further arguments to customize the <code>[polygon()]</code> used for violins.
args_density	An optional list that contains further arguments to customize the <code>[density()]</code> plotted as violins.
args_error_bars	An optional list that contains further arguments that may be passed to <code>arrows()</code> .
args_legend	An optional list that contains further arguments that may be passed to <code>legend()</code>
plot	Character. A vector specifying which elements of the plot should be plotted. Available options are <code>c("points", "error_bars", "bars", "swarms", "violins", "lines")</code> .
jit	Numeric. Determines the amount of horizontal displacement. Defaults to 0.3, defaults to 0.4 if <code>plot = "bars"</code> .
xlab	Character or expression. Label for x axis.
ylab	Character or expression. Label for y axis.
main	Character or expression. For up to two factors, simply specify the main title. If you stratify the data by more than two factors, either specify a single value that will be added to automatically generated main title, <i>or</i> specify an array of multiple titles, one for each plot area.
set_par	Logical. Determines whether <code>par(mfrow = .)</code> should be set for multi-panel plots.

## Details

The measure of dispersion can be either `conf_int()` for between-subjects confidence intervals, `se()` for standard errors, or any other standard function. For within-subjects confidence intervals, specify `wsci()` or `within_subjects_conf_int()`.

If between- or within-subjects confidence intervals are requested, you can also specify the area of the cumulative distribution function that will be covered. For instance, if you want a 98% confidence interval, specify `level = 0.98`. The default is `level = 0.95` for 95% confidence intervals.

### Customization of plot elements:

`apa_factorial_plot()` and its descendants `apa_barplot()`, `apa_lineplot()`, `apa_beeplot()`, and `apa_violinplot()` are wrapper functions that sequentially call:

- `plot.new()`,
- `plot.window()`,
- `axis()` (once for x axis, once for y axis),
- `title()` for axis labels and titles,
- `rect()` for bars in bar plots,
- `points()` for bee swarms,
- `density()` and `polygon()` for violins,
- `lines()` for lines connecting central tendency points,

- `arrows()` for error bars,
- `points()` for tendency points,
- `legend()` for a legend, and
- `lines()` for intercepts.

These calls can be customized by setting the respective parameters `args_*** = list(...)`.

### Value

A named (nested) list of plot options including raw and derived data. *Note that the structure of the return value is about to change in a forthcoming release of papaja.*

### See Also

Other plots for factorial designs: `apa_barplot()`, `apa_beeplot()`, `apa_lineplot()`, `apa_violinplot()`

### Examples

```
apa_factorial_plot(
  data = npk
  , id = "block"
  , dv = "yield"
  , factors = c("N", "P", "K")
  , las = 1
  , plot = c("error_bars", "points", "swarms")
  , ylim = c(0, 100)
)
```

---

apa\_interval

*Typeset Interval Estimate*

---

### Description

Creates a character string to report interval estimates, such as frequentist confidence or Bayesian credible intervals.

### Usage

```
apa_interval(
  x,
  ...,
  conf.int = NULL,
  interval_type = NULL,
  enclose_math = FALSE
)

print_interval(
  x,
  ...,
```



```

    conf.int = NULL,
    interval_type = NULL,
    enclose_math = FALSE
  )

## S3 method for class 'numeric'
apa_interval(
  x,
  y = NULL,
  conf.int = NULL,
  interval_type = NULL,
  enclose_math = FALSE,
  ...
)

## S3 method for class 'matrix'
apa_interval(
  x,
  conf.int = NULL,
  interval_type = NULL,
  enclose_math = FALSE,
  ...
)

## S3 method for class 'data.frame'
apa_interval(x, ...)

## S3 method for class 'list'
apa_interval(x, ...)

apa_confint(x, ..., interval_type = "CI")
print_confint(x, ..., interval_type = "CI")

apa_hdint(x, ..., interval_type = "HDI")
print_hdint(x, ..., interval_type = "HDI")

```

### Arguments

x	Numeric. A vector (of length 2, unless y is also specified) with, a two-column matrix, or a data.frame, which can coerced to a matrix.
...	Arguments passed on to <a href="#">apa_num</a>
conf.int	Numeric. Confidence level of the interval. Ignored if level can be inferred from attributes of x, see Details.
interval_type	Character. Abbreviation indicating the type of interval estimate, e.g. CI.

`enclose_math` Logical. Indicates whether the interval should be enclosed in  $(i.e., a math environment).$

`y` Numeric. An optional vector of the same length as `x`.

### Details

If possible the confidence level of the interval is inferred from attributes of `x`. For a vector of length 2, the attribute `conf.level` is consulted; for a `matrix` or `data.frame` the column names are used, if they are of the format "2.5 \

If `x` is a `matrix` or `data.frame` the row names are used as names for the returned list of intervals.

### Value

A single interval is returned as a character vector of length 1; multiple intervals are returned as a named list of character vectors of length 1.

### See Also

[apa\\_num\(\)](#)

### Examples

```
apa_confint(1, 2, conf.int = 0.95)
apa_confint(c(1, 2), conf.int = 0.95)
apa_confint(matrix(c(1, 2), ncol = 2), conf.int = 0.95)
apa_confint(confint(lm(cars)))
apa_confint(confint(lm(cars)), digits = 3)
```

---

apa\_lineplot

*Line Plots for Factorial Designs that Conform to APA Guidelines*

---

### Description

Creates one or more line plots from a `data.frame` containing data from a factorial design and set APA-friendly defaults.

### Usage

```
apa_lineplot(data, ...)

## Default S3 method:
apa_lineplot(
  data,
  id,
  factors = NULL,
  dv,
  tendency = mean,
  dispersion = conf_int,
```

```

    level = 0.95,
    fun_aggregate = mean,
    na.rm = TRUE,
    use = "all.obs",
    intercept = NULL,
    args_x_axis = NULL,
    args_y_axis = NULL,
    args_title = NULL,
    args_points = NULL,
    args_lines = NULL,
    args_error_bars = NULL,
    args_legend = NULL,
    jit = 0.3,
    xlab = NULL,
    ylab = NULL,
    main = NULL,
    set_par = TRUE,
    ...
)

## S3 method for class 'afex_aov'
apa_lineplot(
  data,
  tendency = mean,
  dispersion = conf_int,
  fun_aggregate = mean,
  ...
)

```

### Arguments

data	A data.frame that contains the data, or an object of class afex_aov.
...	Arguments passed on to <a href="#">apa_factorial_plot</a>
id	Character. Variable name that identifies subjects.
factors	Character. A vector of up to four variable names that is used to stratify the data.
dv	Character. The name of the dependent variable.
tendency	Closure. A function that will be used as measure of central tendency.
dispersion	Closure. A function that will be used to construct error bars (i.e., whiskers). Defaults to <a href="#">conf_int()</a> for 95% between-subjects confidence intervals. See details for more options, especially for within-subjects confidence intervals.
level	Numeric. Defines the width of the interval if confidence intervals are plotted. Defaults to 0.95. for 95% confidence intervals. Ignored if dispersion is not a confidence-interval function. See details.
fun_aggregate	Closure. The function that will be used to aggregate observations within subjects and factors before calculating descriptive statistics for each cell of the design. Defaults to mean.

na.rm	Logical. Specifies if missing values are removed. Defaults to TRUE.
use	Character. Specifies a method to exclude cases if there are missing values <i>after</i> aggregating. Possible options are "all.obs" or "complete.obs".
intercept	Numeric. Adds a horizontal line at height intercept to the plot. Can be either a single value or a matrix. For the matrix case, multiple lines are drawn, where the dimensions of the matrix determine the number of lines to be drawn.
args_x_axis	An optional list that contains further arguments that may be passed to <code>axis()</code> for customizing the <i>x</i> axis.
args_y_axis	An optional list that contains further arguments that may be passed to <code>axis()</code> for customizing the <i>y</i> axis.
args_title	An optional list that contains further arguments that may be passed to <code>title()</code> .
args_points	An optional list that contains further arguments that may be passed to <code>points()</code> .
args_lines	An optional list that contains further arguments that may be passed to <code>lines()</code> .
args_error_bars	An optional list that contains further arguments that may be passed to <code>arrows()</code> .
args_legend	An optional list that contains further arguments that may be passed to <code>legend()</code> .
jit	Numeric. Determines the amount of horizontal displacement. Defaults to 0.3, defaults to 0.4 if <code>plot = "bars"</code> .
xlab	Character or expression. Label for <i>x</i> axis.
ylab	Character or expression. Label for <i>y</i> axis.
main	Character or expression. For up to two factors, simply specify the main title. If you stratify the data by more than two factors, either specify a single value that will be added to automatically generated main title, <i>or</i> specify an array of multiple titles, one for each plot area.
set_par	Logical. Determines whether <code>par(mfrow = .)</code> should be set for multi-panel plots.

## Details

The measure of dispersion can be either `conf_int()` for between-subjects confidence intervals, `se()` for standard errors, or any other standard function. For within-subjects confidence intervals, specify `wsci()` or `within_subjects_conf_int()`.

If between- or within-subjects confidence intervals are requested, you can also specify the area of the cumulative distribution function that will be covered. For instance, if you want a 98% confidence interval, specify `level = 0.98`. The default is `level = 0.95` for 95% confidence intervals.

### Customization of plot elements:

`apa_factorial_plot()` and its descendants `apa_barplot()`, `apa_lineplot()`, `apa_beeplot()`, and `apa_violinplot()` are wrapper functions that sequentially call:

- `plot.new()`,
- `plot.window()`,
- `axis()` (once for *x* axis, once for *y* axis),
- `title()` for axis labels and titles,

- `rect()` for bars in bar plots,
- `points()` for bee swarms,
- `density()` and `polygon()` for violins,
- `lines()` for lines connecting central tendency points,
- `arrows()` for error bars,
- `points()` for tendency points,
- `legend()` for a legend, and
- `lines()` for intercepts.

These calls can be customized by setting the respective parameters `args_*** = list(...)`.

### Value

A named (nested) list of plot options including raw and derived data. *Note that the structure of the return value is about to change in a forthcoming release of papaja.*

### See Also

Other plots for factorial designs: [apa\\_barplot\(\)](#), [apa\\_beeplot\(\)](#), [apa\\_factorial\\_plot\(\)](#), [apa\\_violinplot\(\)](#)

### Examples

```
apa_lineplot(  
  data = npk  
  , id = "block"  
  , dv = "yield"  
  , factors = c("N")  
)  
  
apa_lineplot(  
  data = npk  
  , id = "block"  
  , dv = "yield"  
  , factors = c("N", "P")  
  , args_legend = list(x = "center")  
  , jit = 0  
)  
  
apa_lineplot(  
  data = npk  
  , id = "block"  
  , dv = "yield"  
  , factors = c("N", "P", "K")  
  , ylim = c(0, 80)  
  , level = .34  
  , las = 1  
)
```

---

`apa_num`*Typeset Numerical Values for Printing and Reporting*

---

**Description**

Converts numerical values to character strings for printing and reporting.

**Usage**

```
apa_num(x, ...)  
  
printnum(x, ...)  
  
print_num(x, ...)  
  
## Default S3 method:  
apa_num(x, na_string = getOption("papaja.na_string"), ...)  
  
## S3 method for class 'list'  
apa_num(x, ...)  
  
## S3 method for class 'integer'  
apa_num(  
  x,  
  numerals = TRUE,  
  capitalize = FALSE,  
  zero_string = "no",  
  na_string = getOption("papaja.na_string"),  
  ...  
)  
  
## S3 method for class 'numeric'  
apa_num(  
  x,  
  gt1 = TRUE,  
  zero = TRUE,  
  na_string = getOption("papaja.na_string"),  
  use_math = TRUE,  
  add_equals = FALSE,  
  ...  
)  
  
## S3 method for class 'data.frame'  
apa_num(x, margin = 2, ...)  
  
## S3 method for class 'matrix'  
apa_num(x, margin = 2, ...)
```

```
## S3 method for class 'tiny_labelled'
apa_num(x, ...)
```

### Arguments

**x** Can be either a single value, vector, matrix, data.frame.

**...** Arguments passed on to `base::formatC`

**digits** the desired number of digits after the decimal point (format = "f") or *significant* digits (format = "g", "e" or "fg").  
 Default: 2 for integer, 4 for real numbers. If less than 0, the C default of 6 digits is used. If specified as more than 50, 50 will be used with a warning unless format = "f" where it is limited to typically 324. (Not more than 15–21 digits need be accurate, depending on the OS and compiler used. This limit is just a precaution against segfaults in the underlying C runtime.)

**width** the total field width; if both **digits** and **width** are unspecified, **width** defaults to 1, otherwise to **digits** + 1. **width** = 0 will use **width** = **digits**, **width** < 0 means left justify the number in this field (equivalent to **flag** = "-"). If necessary, the result will have more characters than **width**. For character data this is interpreted in characters (not bytes nor display width).

**format** equal to "d" (for integers), "f", "e", "E", "g", "G", "fg" (for reals), or "s" (for strings). Default is "d" for integers, "g" for reals.  
 "f" gives numbers in the usual xxx.xxx format; "e" and "E" give n.ddde+nn or n.dddE+nn (scientific format); "g" and "G" put x[i] into scientific format only if it saves space to do so *and* drop trailing zeros and decimal point - unless **flag** contains "#" which keeps trailing zeros for the "g", "G" formats.  
 "fg" (our own hybrid format) uses fixed format as "f", but **digits** as the minimum number of *significant* digits. This can lead to quite long result strings, see examples below. Note that unlike `signif` this prints large numbers with more significant digits than **digits**. Trailing zeros are *dropped* in this format, unless **flag** contains "#".

**flag** for `formatC`, a character string giving a format modifier as in Kernighan and Ritchie (1988, page 243) or the C+99 standard.  
 "0" pads leading zeros;  
 "-" does left adjustment,  
 "+" ensures a sign in all cases, i.e., "+" for positive numbers ,  
 " " if the first character is not a sign, the space character " " will be used instead.  
 "#" specifies "an alternative output form", specifically depending on format.  
 "' ' on some platform–locale combination, activates "thousands' grouping" for decimal conversion,  
 "I" in some versions of 'glibc' allow for integer conversion to use the locale's alternative output digits, if any.

There can be more than one of these flags, in any order. Other characters used to have no effect for character formatting, but signal an error since R 3.4.0.

	mode "double" (or "real"), "integer" or "character". Default: Determined from the storage mode of x.
	big.mark character; if not empty used as mark between every big.interval decimals <i>before</i> (hence big) the decimal point.
	big.interval see big.mark above; defaults to 3.
	small.mark character; if not empty used as mark between every small.interval decimals <i>after</i> (hence small) the decimal point.
	small.interval see small.mark above; defaults to 5.
	decimal.mark the character to be used to indicate the numeric decimal point.
	preserve.width string specifying if the string widths should be preserved where possible in those cases where marks (big.mark or small.mark) are added. "common", the default, corresponds to <code>format</code> -like behavior whereas "individual" is the default in <code>formatC()</code> . Value can be abbreviated.
	zero.print logical, character string or NULL specifying if and how <i>zeros</i> should be formatted specially. Useful for pretty printing 'sparse' objects.
	drop0trailing logical, indicating if trailing zeros, i.e., "0" <i>after</i> the decimal mark, should be removed; also drops "e+00" in exponential formats. This is simply passed to <code>prettyNum()</code> , see the 'Details'.
na_string	Character. String to print if any element of x is NA.
numerals	Logical. Indicates if integers should be converted to words.
capitalize	Logical. Indicates if first letter should be capitalized. Ignored if numerals = TRUE.
zero_string	Character. Word to print if x is a zero integer.
gt1	Logical. Indicates if the statistic can, in principle, have an absolute value greater than 1. If FALSE, leading zeros are omitted.
zero	Logical. Indicates if the statistic can, in principle, be 0. If FALSE, a string of the form <code>&lt; 0.001</code> is returned instead of 0.
use_math	Logical. Indicates whether to use \$ in the output so that Inf or scientific notation is rendered correctly.
add_equals	Logical. Indicates if the output string should be prepended with an =.
margin	Integer. If x is a matrix or data.frame, the function is applied either across rows (margin = 1) or columns (margin = 2). See <code>apply()</code> .

### Details

If x is a vector, all arguments can be vectors according to which each element of the vector is formatted. Parameters are recycled if length of x exceeds the length of the parameter vectors. If x is a matrix or data.frame, the vectors specify the formatting of either rows or columns according to the value of margin.

We recommend to use `apa_num()`, rather than `printnum()` or `print_num()`, which are aliases kept only for backward compatibility.

### Value

An object of the same class as x with all numeric values converted to character.



**See Also**

[apa\\_p\(\)](#), [apa\\_df\(\)](#)

**Examples**

```
apa_num(1/3)
apa_num(1/3, gt1 = FALSE)
apa_num(1/3, digits = 5)

apa_num(0)
apa_num(0, zero = FALSE)
```

---

 apa\_p

*Prepare Numeric Values for Printing as p value*


---

**Description**

Convenience wrapper for [apa\\_num](#) to print  $p$  values.

**Usage**

```
apa_p(x, digits = 3L, na_string = "", add_equals = FALSE)

printp(x, digits = 3L, na_string = "", add_equals = FALSE)

print_p(x, digits = 3L, na_string = "", add_equals = FALSE)
```

**Arguments**

<code>x</code>	Numeric. The $p$ value(s) to report.
<code>digits</code>	Integer. The desired number of digits after the decimal point, passed on to <a href="#">formatC</a> .
<code>na_string</code>	Character. String to print if any element of <code>x</code> is NA.
<code>add_equals</code>	Logical. Indicates if the output string should be prepended with an =.

**Value**

An object of the same class as `x` with all numeric values converted to character.

**See Also**

[apa\\_num\(\)](#), [apa\\_df\(\)](#)

**Examples**

```
apa_p(0.05)
apa_p(0.0005)
apa_p(0.9999999)
apa_p(c(.001, 0), add_equals = TRUE)
```

---

apa_prepare_doc	<i>Prepare APA document (deprecated)</i>
-----------------	--

---

### Description

Prepares objects to be used in the rendering process and creates title page and abstract for MS Word documents. *This function has been defunct. It is no longer needed.*

### Usage

```
apa_prepare_doc()
```

### Details

The function creates and locks a non-exported object `apa_lang` that is used by other **papaja**-functions. `apa_lang` is a list containing localizations for document elements such as abstract and title. The selected language is defined by the `lang`-parameter in the documents yml-header. Currently, English (default) and German ("german") are supported.

If the output document is MS Word (output: `\link{apa6_word}`) the function additionally creates a title page and adds the abstract. You should, therefore, always call `apa_prepare_doc` at the very beginning of the R Markdown document.

### Value

Returns NULL invisibly.

### See Also

[apa6\\_docx\(\)](#)

---

apa_print	<i>Typeset Statistical Results</i>
-----------	------------------------------------

---

### Description

A generic function that takes objects from various statistical methods to create formatted character strings to report the results in accordance with APA manuscript guidelines. The function invokes particular methods, which depend on the `class` of the first argument.

### Usage

```
apa_print(x, ...)
```

**Arguments**

x	A model object.
...	Additional arguments passed to methods.

**Value**

apa\_print()-methods return a named list of class `apa_results` containing the following elements:

estimate	One or more character strings giving point estimates, confidence intervals, and confidence level. A single string is returned in a vector; multiple strings are returned as a named list. If no estimate is available the element is NULL.
statistic	One or more character strings giving the test statistic, parameters (e.g., degrees of freedom), and p-value. A single string is returned in a vector; multiple strings are returned as a named list. If no estimate is available the element is NULL.
full_result	One or more character strings comprised ‘estimate’ and ‘statistic’. A single string is returned in a vector; multiple strings are returned as a named list.
table	A data.frame of class <code>apa_results_table</code> that contains all elements of estimate and statistics. This table can be passed to <code>apa_table()</code> for reporting.

Column names in `apa_results_table` are standardized following the **broom glossary** (e.g., `term`, `estimate`, `conf.int`, `statistic`, `df`, `df.residual`, `p.value`). Additionally, each column is labelled (e.g.,  $\hat{\eta}^2_G$  or  $t$ ) using the **tinylab**s package and these labels are used as column names when an `apa_results_table` is passed to `apa_table()`.

**See Also**

Other `apa_print`: `apa_print.BFBayesFactor()`, `apa_print.aov()`, `apa_print.emmGrid()`, `apa_print.glht()`, `apa_print.htest()`, `apa_print.list()`, `apa_print.lm()`, `apa_print.lme()`, `apa_print.merMod()`

**Examples**

```
# List methods for apa_print()
methods("apa_print")
```

---

apa\_print.aov

*Typeset Statistical Results from Analysis of Variance (or Deviance)*

---

**Description**

These methods take objects from various R functions that calculate analysis of variance (i.e., ANOVA) or analysis of deviance. They create formatted character strings to report the results in accordance with APA manuscript guidelines. For anova-objects from model comparisons see `apa_print.list()`.

**Usage**

```
## S3 method for class 'aov'
apa_print(
  x,
  estimate = getOption("papaja.estimate_anova", "ges"),
  observed = NULL,
  intercept = FALSE,
  mse = getOption("papaja.mse", TRUE),
  in_paren = FALSE,
  ...
)

## S3 method for class 'summary.aov'
apa_print(
  x,
  estimate = getOption("papaja.estimate_anova", "ges"),
  observed = NULL,
  intercept = FALSE,
  mse = getOption("papaja.mse", TRUE),
  in_paren = FALSE,
  ...
)

## S3 method for class 'aovlist'
apa_print(
  x,
  estimate = getOption("papaja.estimate_anova", "ges"),
  observed = NULL,
  intercept = FALSE,
  mse = getOption("papaja.mse", TRUE),
  in_paren = FALSE,
  ...
)

## S3 method for class 'summary.aovlist'
apa_print(
  x,
  estimate = getOption("papaja.estimate_anova", "ges"),
  observed = NULL,
  intercept = FALSE,
  mse = getOption("papaja.mse", TRUE),
  in_paren = FALSE,
  ...
)

## S3 method for class 'Anova.mlm'
apa_print(
  x,
```

```
    estimate = getOption("papaja.estimate_anova", "ges"),
    observed = NULL,
    correction = getOption("papaja.sphericity_correction"),
    intercept = FALSE,
    mse = getOption("papaja.mse", TRUE),
    in_paren = FALSE,
    ...
)

## S3 method for class 'summary.Anova.mlm'
apa_print(
  x,
  estimate = getOption("papaja.estimate_anova", "ges"),
  observed = NULL,
  correction = getOption("papaja.sphericity_correction"),
  intercept = FALSE,
  mse = getOption("papaja.mse", TRUE),
  in_paren = FALSE,
  ...
)

## S3 method for class 'afex_aov'
apa_print(
  x,
  estimate = getOption("papaja.estimate_anova", "ges"),
  observed = NULL,
  correction = getOption("papaja.sphericity_correction"),
  intercept = FALSE,
  mse = getOption("papaja.mse", TRUE),
  in_paren = FALSE,
  ...
)

## S3 method for class 'anova'
apa_print(
  x,
  estimate = getOption("papaja.estimate_anova", "ges"),
  observed = NULL,
  intercept = FALSE,
  mse = getOption("papaja.mse", TRUE),
  in_paren = FALSE,
  ...
)

## S3 method for class 'manova'
apa_print(x, test = "Pillai", in_paren = FALSE, ...)

## S3 method for class 'summary.manova'
```

```
apa_print(x, in_paren = FALSE, ...)
```

### Arguments

<code>x</code>	An object containing the results from an analysis of variance ANOVA
<code>estimate</code>	Character, function, or data frame. Determines which estimate of effect size is to be used. See details.
<code>observed</code>	Character. The names of the factors that are observed, i.e., not manipulated. Necessary only for calculating <i>generalized</i> eta squared; otherwise ignored. If <code>x</code> is of class <code>afex_aov</code> , <code>observed</code> is automatically deduced from <code>x</code> .
<code>intercept</code>	Logical. Indicates if the intercept term should be included in output.
<code>mse</code>	Logical. Indicates if mean squared errors should be included in output. The default is taken from the global option <code>getOption("papaja.mse")</code> . It is <code>FALSE</code> if the <b>effectsize</b> package is installed and <code>TRUE</code> if it is not installed. This can be changed either by supplying a different value in the function call or by changing the global default via <code>options(papaja.mse = ...)</code> .
<code>in_paren</code>	Logical. Whether the formatted string is to be reported in parentheses. If <code>TRUE</code> , parentheses in the formatted string (e.g., those enclosing degrees of freedom) are replaced with brackets.
<code>...</code>	Further arguments that may be passed to <code>apa_num</code> to format estimates (i.e., columns <code>estimate</code> and <code>conf.int</code> ).
<code>correction</code>	Character. For repeated-measures ANOVA, the type of sphericity correction to be used. Possible values are "GG" for the Greenhouse-Geisser method (the default), "HF" for the Huyn-Feldt method, or "none" for no correction.
<code>test</code>	Character. For MANOVA, the multivariate test statistic to be reported, see <a href="#">summary.manova</a> .

### Details

The factor names are sanitized to facilitate their use as list names (see Value section). Parentheses are omitted and other non-word characters are replaced by `_`.

Argument `estimate` determines which measure of effect size is to be used: It is currently possible to provide one of three characters to specify the to-be-calculated effect size: "ges" for generalized  $\eta^2$ , "pes" for partial  $\eta^2$ , and "es" for  $\eta^2$ . Note that  $\eta^2$  is calculated correctly if and only if the design is balanced.

It is also possible to provide a `data.frame` with columns `estimate`, `conf.low`, and `conf.high`, which allows for including custom effect-size measures.

A third option is to provide a function from the **effectsize** package that will be used to calculate effect-size measures from `x`. If **effectsize** is installed (and **papaja** is loaded), this is the new default. This default can be changed via `options(papaja.estimate_anova = ...)`.

### Value

`apa_print()`-methods return a named list of class `apa_results` containing the following elements:

estimate	One or more character strings giving point estimates, confidence intervals, and confidence level. A single string is returned in a vector; multiple strings are returned as a named list. If no estimate is available the element is NULL.
statistic	One or more character strings giving the test statistic, parameters (e.g., degrees of freedom), and p-value. A single string is returned in a vector; multiple strings are returned as a named list. If no estimate is available the element is NULL.
full_result	One or more character strings comprised 'estimate' and 'statistic'. A single string is returned in a vector; multiple strings are returned as a named list.
table	A data.frame of class apa_results_table that contains all elements of estimate and statistics. This table can be passed to apa_table() for reporting.

Column names in apa\_results\_table are standardized following the **broom** glossary (e.g., term, estimate.conf.int, statistic, df, df.residual, p.value). Additionally, each column is labelled (e.g.,  $\hat{\eta}^2_G$  or  $t$ ) using the **tinylab**s package and these labels are used as column names when an apa\_results\_table is passed to apa\_table().

## References

Bakeman, R. (2005). Recommended effect size statistics for repeated measures designs. *Behavior Research Methods*, 37 (3), 379–384. doi: [doi:10.3758/BF03192707](https://doi.org/10.3758/BF03192707)

## See Also

[aov\(\)](#), [car::Anova\(\)](#), [apa\\_print.list\(\)](#)

Other apa\_print: [apa\\_print\(\)](#), [apa\\_print.BFBayesFactor\(\)](#), [apa\\_print.emmGrid\(\)](#), [apa\\_print.glm\(\)](#), [apa\\_print.htest\(\)](#), [apa\\_print.list\(\)](#), [apa\\_print.lm\(\)](#), [apa\\_print.lme\(\)](#), [apa\\_print.merMod\(\)](#)

## Examples

```
## From Venables and Ripley (2002) p. 165.
npk_aov <- aov(yield ~ block + N * P * K, npk)
apa_print(npk_aov)

# Use the effectsize package to calculate partial eta-squared with
# confidence intervals
apa_print(npk_aov, estimate = effectsize::omega_squared)
```

---

apa\_print.BFBayesFactor

*Typeset Bayes Factors*

---

## Description

These methods take result objects from the **BayesFactor** package to create formatted character strings to report the results in accordance with APA manuscript guidelines.

**Usage**

```
## S3 method for class 'BFBayesFactor'
apa_print(
  x,
  stat_name = NULL,
  est_name = NULL,
  subscript = NULL,
  escape_subscript = FALSE,
  scientific_threshold = NULL,
  reciprocal = FALSE,
  log = FALSE,
  mcmc_error = any(x@bayesFactor$error > 0.05),
  iterations = 10000,
  standardized = FALSE,
  central_tendency = median,
  interval = hd_int,
  interval_type = "HDI",
  bf_r1 = NULL,
  bf_lr = NULL,
  ...
)

## S3 method for class 'BFBayesFactorTop'
apa_print(x, reciprocal = FALSE, ...)
```

**Arguments**

<code>x</code>	Output object. See details.
<code>stat_name</code>	Character. If NULL (the default), the name given in <code>x</code> is used for the <i>test statistic</i> , otherwise the supplied name is used. See details.
<code>est_name</code>	Character. If NULL (the default), the name given in <code>x</code> (or a formally correct adaptation) is used for the <i>estimate</i> , otherwise the supplied name is used. See details.
<code>subscript</code>	Character. Index used to specify the model comparison for the Bayes factors, e.g., "+0" yields $BF_{+0}$ . If NULL default to "10".
<code>escape_subscript</code>	Logical. If TRUE special LaTeX characters, such as % or _, in subscript are escaped.
<code>scientific_threshold</code>	Numeric. Named vector of length 2 taking the form <code>c(min = 1/10, max = 1e6)</code> . Bayes factors that exceed these thresholds will be printed in scientific notation.
<code>reciprocal</code>	Logical. If TRUE the reciprocal of all Bayes factors is taken before results are formatted. The advantage over specifying <code>x = t(x)</code> is that the default ( <i>only</i> the default) index specifying the model comparison is automatically reversed, see <code>subscript</code> .
<code>log</code>	Logical. If TRUE the logarithm of the Bayes factor is reported.



mcmc_error	Logical. If TRUE estimation error of the Bayes factor(s) is reported.
iterations	Numeric. Number of iterations of the MCMC sampler to estimate HDIs from the posterior.
standardized	Logical. Whether to return standardized or unstandardized effect size estimates.
central_tendency	Function to calculate central tendency of MCMC samples to obtain a point estimate from the posterior.
interval	Function to calculate an interval estimate of MCMC samples from the posterior. The returned object must be either a named vector or matrix with (column) names giving the interval bounds (e.g., 2.5% and 97.5%) or with an attribute <code>conf.level</code> (e.g., 0.95).
interval_type	Character. Used to specify the type of interval in the formatted text.
bf_r1	Numeric. Vector of the same length as <code>x</code> giving Bayes factors in favor of an order constraint relative to the unconstrained model (see details). Must be on log-scale if <code>log = TRUE</code> .
bf_1r	Numeric. Same as <code>bf_r1</code> (see details).
...	Arguments passed on to <code>apa_num.numeric</code>
gt1	Logical. Indicates if the statistic can, in principle, have an absolute value greater than 1. If FALSE, leading zeros are omitted.
zero	Logical. Indicates if the statistic can, in principle, be 0. If FALSE, a string of the form <code>&lt; 0.001</code> is returned instead of 0.
na_string	Character. String to print if any element of <code>x</code> is NA.
use_math	Logical. Indicates whether to use <code>\$</code> in the output so that Inf or scientific notation is rendered correctly.
add_equals	Logical. Indicates if the output string should be prepended with an <code>=</code> .

## Details

`stat_name` and `est_name` are placed in the output string and are thus passed to pandoc or LaTeX through **knitr**. To the extent it is supported by the final document type, you can pass LaTeX-markup to format the final text (e.g., `M_\Delta` yields  $M_\Delta$ ).

For models with order constraint, the evidence for the order constraint relative to the null model can be obtained by multiplying the Bayes factor  $BF_{r1}$  for the order constraint relative to the unconstrained model (`bf_r1`) with the Bayes factor  $BF_{10}$  for the unconstrained model relative to the null model,

$$\frac{p(y | \mathcal{M}_r)}{p(y | \mathcal{M}_0)} = \frac{p(y | \mathcal{M}_r)}{p(y | \mathcal{M}_1)} \times \frac{p(y | \mathcal{M}_1)}{p(y | \mathcal{M}_0)}$$

$BF_{r1}$  can be calculated from the prior and posterior odds of the order constraint (e.g., Morey & Wagenmakers, 2014). If `bf_r1` (or `bf_1r`) is specified they are multiplied with the corresponding Bayes factor supplied in `x` before the reciprocal is taken and the results are formatted. Note, that it is not possible to determine whether `x` gives  $BF_{10}$  or  $BF_{01}$  and, hence, `bf_r1` and `bf_1r` are treated identically; the different argument names only serve to ensure the expressiveness of the code. It is the user's responsibility to ensure that the supplied Bayes factor is correct!

**Value**

apa\_print()-methods return a named list of class `apa_results` containing the following elements:

<code>estimate</code>	One or more character strings giving point estimates, confidence intervals, and confidence level. A single string is returned in a vector; multiple strings are returned as a named list. If no estimate is available the element is <code>NULL</code> .
<code>statistic</code>	One or more character strings giving the test statistic, parameters (e.g., degrees of freedom), and p-value. A single string is returned in a vector; multiple strings are returned as a named list. If no estimate is available the element is <code>NULL</code> .
<code>full_result</code>	One or more character strings comprised ‘estimate’ and ‘statistic’. A single string is returned in a vector; multiple strings are returned as a named list.
<code>table</code>	A <code>data.frame</code> of class <code>apa_results_table</code> that contains all elements of estimate and statistics. This table can be passed to <code>apa_table()</code> for reporting.

Column names in `apa_results_table` are standardized following the **broom** glossary (e.g., `term`, `estimate`, `conf.int`, `statistic`, `df`, `df.residual`, `p.value`). Additionally, each column is labelled (e.g.,  $\hat{\eta}^2_G$  or  $t$ ) using the **tinylab**s package and these labels are used as column names when an `apa_results_table` is passed to `apa_table()`.

**References**

Morey, R. D., & Wagenmakers, E.-J. (2014). Simple relation between Bayesian order-restricted and point-null hypothesis tests. *Statistics & Probability Letters*, 92, 121–124. doi: [doi:10.1016/j.spl.2014.05.010](https://doi.org/10.1016/j.spl.2014.05.010)

**See Also**

Other `apa_print`: `apa_print()`, `apa_print.aov()`, `apa_print.emmGrid()`, `apa_print.glht()`, `apa_print.htest()`, `apa_print.list()`, `apa_print.lm()`, `apa_print.lme()`, `apa_print.merMod()`

**Examples**

```
# ANOVA

data(sleep, package = "BayesFactor")
bayesian_anova <- BayesFactor::anovaBF(
  extra ~ group + ID
  , data = sleep
  , whichRandom = "ID"
  , progress = FALSE
)

# Paired t-test
ttest_paired <- BayesFactor::ttestBF(
  x = sleep$extra[sleep$group == 1]
  , y = sleep$extra[sleep$group == 2]
  , paired = TRUE
)

# Results for paired t-tests are indistinguishable
```

```

# from one-sample t-tests. We therefore specify the
# appropriate `est_name` manually.
apa_print(
  ttest_paired
  , est_name = "M_D"
  , iterations = 1000
)

apa_print(
  ttest_paired
  , iterations = 1000
  , interval = function(x) quantile(x, probs = c(0.025, 0.975))
  , interval_type = "CrI"
)

```

---

apa\_print.emmGrid

Typeset Statistical Results from Estimated Marginal Means

---

## Description

Takes various **emmeans** objects to create formatted character strings to report the results in accordance with APA manuscript guidelines. **emmeans** supports a wide range of analyses, not all of which are currently (fully) supported. Proceed with caution.

## Usage

```

## S3 method for class 'emmGrid'
apa_print(x, infer = TRUE, conf.int = 0.95, ...)

## S3 method for class 'summary_emm'
apa_print(
  x,
  contrast_names = NULL,
  est_name = "\\hat{\\theta}",
  in_paren = FALSE,
  ...
)

## S3 method for class 'lsmobj'
apa_print(x, ...)

## S3 method for class 'summary.ref.grid'
apa_print(x, ...)

```

## Arguments

x                    Object

<code>infer</code>	A vector of one or two logical values. The first determines whether confidence intervals are displayed, and the second determines whether $t$ tests and $P$ values are displayed. If only one value is provided, it is used for both.
<code>conf.int</code>	Numeric. Confidence level for confidence intervals.
<code>...</code>	Arguments passed on to <code>apa_num</code>
<code>contrast_names</code>	Character. An optional vector of names to label the calculated contrasts.
<code>est_name</code>	Character. If NULL (default) the name of the estimate is inferred from the function call of the model object supplied to <code>emmeans</code> .
<code>in_paren</code>	Logical. Whether the formatted string is to be reported in parentheses. If TRUE, parentheses in the formatted string (e.g., those enclosing degrees of freedom) are replaced with brackets.

### Details

When p-values and confidence intervals are adjusted for multiple testing, the correction method is added as an index to the output (e.g.  $p_{\text{Tukey}(3)}$ ). Values in parenthesis indicate the size of the family of tests or the rank of the set of linear functions (for the Scheffé method).

If possible, each family of tests is additionally marked in the returned table by alphabetic superscripts.

Generally, the `summary_emm` objects returned by `emmeans::summary_emm` omit information that may be needed to add some of the information on the adjustments made to p-values and confidence intervals. It is therefore preferable to pass `emmGrid`-objects if possible. For example, by using `emmeans(object, 1 ~ x1, adjust = "scheffe")`.

### Value

`apa_print()`-methods return a named list of class `apa_results` containing the following elements:

<code>estimate</code>	One or more character strings giving point estimates, confidence intervals, and confidence level. A single string is returned in a vector; multiple strings are returned as a named list. If no estimate is available the element is NULL.
<code>statistic</code>	One or more character strings giving the test statistic, parameters (e.g., degrees of freedom), and p-value. A single string is returned in a vector; multiple strings are returned as a named list. If no estimate is available the element is NULL.
<code>full_result</code>	One or more character strings comprised ‘estimate’ and ‘statistic’. A single string is returned in a vector; multiple strings are returned as a named list.
<code>table</code>	A data.frame of class <code>apa_results_table</code> that contains all elements of estimate and statistics. This table can be passed to <code>apa_table()</code> for reporting.

Column names in `apa_results_table` are standardized following the **broom glossary** (e.g., `term`, `estimate`, `conf.int`, `statistic`, `df`, `df.residual`, `p.value`). Additionally, each column is labelled (e.g.,  $\hat{\eta}^2_G$  or  $t$ ) using the **tinylab** package and these labels are used as column names when an `apa_results_table` is passed to `apa_table()`.

**See Also**

Other `apa_print`: `apa_print()`, `apa_print.BFBayesFactor()`, `apa_print.aov()`, `apa_print.glht()`, `apa_print.htest()`, `apa_print.list()`, `apa_print.lm()`, `apa_print.lme()`, `apa_print.merMod()`

**Examples**

```
# From the emmeans manual:
library(emmeans)
warp.lm <- lm(breaks ~ wool*tension, data = warpbreaks)
warp.emm <- emmeans(warp.lm, ~ tension | wool)
warp.contr <- contrast(warp.emm, "poly")

apa_print(warp.contr)

# In this example, because degrees of freedom are equal across all rows
# of the output, it is possible to move that information to the variable
# labels. This is useful if a compact results table is required:

df_into_label(apa_print(warp.contr))
```

---

`apa_print.glht`
*Typeset Statistical Results from General Linear Hypothesis Tests*


---

**Description**

*These methods are not properly tested and should be considered experimental.*

**Usage**

```
## S3 method for class 'glht'
apa_print(x, test = multcomp::adjusted(), ...)

## S3 method for class 'summary.glht'
apa_print(x, conf.int = 0.95, in_paren = FALSE, ...)
```

**Arguments**

<code>x</code>	Object
<code>test</code>	Function. Computes p-values (adjusted for multiple comparisons).
<code>...</code>	Further arguments to pass to <code>apa_num</code> to format the estimate.
<code>conf.int</code>	Numeric. If NULL (default) the function tries to obtain confidence intervals from <code>x</code> . Other confidence intervals can be supplied as a vector of length 2 (lower and upper boundary, respectively) with attribute <code>conf.level</code> , e.g., when calculating bootstrapped confidence intervals.
<code>in_paren</code>	Logical. Whether the formatted string is to be reported in parentheses. If TRUE, parentheses in the formatted string (e.g., those enclosing degrees of freedom) are replaced with brackets.

**Value**

apa\_print()-methods return a named list of class apa\_results containing the following elements:

estimate	One or more character strings giving point estimates, confidence intervals, and confidence level. A single string is returned in a vector; multiple strings are returned as a named list. If no estimate is available the element is NULL.
statistic	One or more character strings giving the test statistic, parameters (e.g., degrees of freedom), and p-value. A single string is returned in a vector; multiple strings are returned as a named list. If no estimate is available the element is NULL.
full_result	One or more character strings comprised 'estimate' and 'statistic'. A single string is returned in a vector; multiple strings are returned as a named list.
table	A data.frame of class apa_results_table that contains all elements of estimate and statistics. This table can be passed to apa_table() for reporting.

Column names in apa\_results\_table are standardized following the **broom glossary** (e.g., term, estimate.conf.int, statistic, df, df.residual, p.value). Additionally, each column is labelled (e.g.,  $\hat{\eta}^2_G$  or  $t$ ) using the **tinylab** package and these labels are used as column names when an apa\_results\_table is passed to apa\_table().

**See Also**

Other apa\_print: `apa_print()`, `apa_print.BFBayesFactor()`, `apa_print.aov()`, `apa_print.emmGrid()`, `apa_print.htest()`, `apa_print.list()`, `apa_print.lm()`, `apa_print.lme()`, `apa_print.merMod()`

**Examples**

```
# From the multcomp::glht() examples:
library(multcomp)
amod <- aov(breaks ~ tension, data = warpbreaks)
glht_out <- glht(amod, linfct = mcp(tension = "Tukey"))
apa_print(glht_out)

# In this example, because degrees of freedom are equal across all rows
# of the output, it is possible to move that information to the variable
# labels. This is useful if a compact results table is required:

df_into_label(apa_print(glht_out))
```

---

apa\_print.htest

*Typeset Statistical Results from Hypothesis Tests*

---

**Description**

Takes htest objects from various statistical methods (e.g., `t.test()`, `wilcox.test()`, `cor.test()`) to create formatted character strings to report the results in accordance with APA manuscript guidelines.

**Usage**

```
## S3 method for class 'htest'
apa_print(
  x,
  stat_name = NULL,
  est_name = NULL,
  n = NULL,
  conf.int = NULL,
  in_paren = FALSE,
  ...
)
```

**Arguments**

x	An htest object. See details.
stat_name	Character. If NULL (the default), the name given in x (or a formally correct adaptation, such as $\chi^2$ instead of "x-squared") is used for the <i>test statistic</i> , otherwise the supplied name is used. See details.
est_name	Character. If NULL (the default), the name given in x (or a formally correct adaptation, such as $r_S$ instead of "rho") is used for the <i>estimate</i> , otherwise the supplied name is used. See details.
n	Numeric. Sample size; required when reporting $\chi^2$ tests, otherwise this parameter is ignored.
conf.int	Numeric. If NULL (the default), the function tries to obtain confidence intervals from x. Other confidence intervals can be supplied as a vector of length 2 (lower and upper boundary, respectively) with attribute <code>conf.level</code> set, e.g., when calculating bootstrapped confidence intervals.
in_paren	Logical. Whether the formatted string is to be reported in parentheses. If TRUE, parentheses in the formatted string (e.g., those enclosing degrees of freedom) are replaced with brackets.
...	Arguments passed on to <a href="#">apa_num</a>

**Details**

The function should work on a wide range of htest objects. Due to the large number of functions that produce these objects and their idiosyncrasies, the returned strings should be compared to the original object. If you experience inaccuracies you may report these [here](#) (please include a reproducible example in your report).

`stat_name` and `est_name` are placed in the output string and are thus passed to pandoc or LaTeX through **knitr**. Thus, to the extent it is supported by the final document type, you can pass LaTeX-markup to format the final text (e.g., `\\tau` yields  $\tau$ ).

**Value**

`apa_print()`-methods return a named list of class `apa_results` containing the following elements:

estimate	One or more character strings giving point estimates, confidence intervals, and confidence level. A single string is returned in a vector; multiple strings are returned as a named list. If no estimate is available the element is NULL.
statistic	One or more character strings giving the test statistic, parameters (e.g., degrees of freedom), and p-value. A single string is returned in a vector; multiple strings are returned as a named list. If no estimate is available the element is NULL.
full_result	One or more character strings comprised 'estimate' and 'statistic'. A single string is returned in a vector; multiple strings are returned as a named list.
table	A data.frame of class apa_results_table that contains all elements of estimate and statistics. This table can be passed to apa_table() for reporting.

Column names in apa\_results\_table are standardized following the **broom** glossary (e.g., term, estimate.conf.int, statistic, df, df.residual, p.value). Additionally, each column is labelled (e.g.,  $\hat{\eta}^2_G$  or  $t$ ) using the **tinylabels** package and these labels are used as column names when an apa\_results\_table is passed to apa\_table().

### See Also

Other apa\_print: [apa\\_print\(\)](#), [apa\\_print.BFBayesFactor\(\)](#), [apa\\_print.aov\(\)](#), [apa\\_print.emmGrid\(\)](#), [apa\\_print.glht\(\)](#), [apa\\_print.list\(\)](#), [apa\\_print.lm\(\)](#), [apa\\_print.lme\(\)](#), [apa\\_print.merMod\(\)](#)

### Examples

```
# Comparisons of central tendencies
t_stat <- t.test(extra ~ group, data = sleep)
apa_print(t_stat)
apa_print(t_stat, stat_name = "tee")

wilcox_stat <- wilcox.test(extra ~ group, data = sleep, exact = FALSE)
apa_print(wilcox_stat)

# Correlations
## Data from Hollander & Wolfe (1973), p. 187f.
x <- c(44.4, 45.9, 41.9, 53.3, 44.7, 44.1, 50.7, 45.2, 60.1)
y <- c(2.6, 3.1, 2.5, 5.0, 3.6, 4.0, 5.2, 2.8, 3.8)
cor_stat <- cor.test(x, y, method = "spearman")
apa_print(cor_stat)

# Contingency tables
## Data from Fleiss (1981), p. 139.
smokers <- c(83, 90, 129, 70)
patients <- c(86, 93, 136, 82)
prop_stat <- prop.test(smokers, patients)
apa_print(prop_stat, n = sum(patients))
```



**Description**

This method performs comparisons of `lm`-objects and creates formatted character strings and a model comparison table to report the results in accordance with APA manuscript guidelines.

**Usage**

```
## S3 method for class 'list'
apa_print(
  x,
  anova_fun = stats::anova,
  conf.int = 0.9,
  boot_samples = 10000,
  progress_bar = interactive(),
  observed = TRUE,
  in_paren = FALSE,
  ...
)
```

**Arguments**

<code>x</code>	List. A list containing to-be-compared <code>lm</code> objects. If the list is completely named, element names are used as model names in the output object.
<code>anova_fun</code>	Function. Function to compare model-objects contained in <code>x</code> .
<code>conf.int</code>	Numeric. Confidence level for the bootstrap confidence interval for $\Delta R^2$ (range [0, 1]); ignored if <code>boot_samples = 0</code> .
<code>boot_samples</code>	Numeric. Number of bootstrap samples to estimate confidence intervals for $\Delta R^2$ .
<code>progress_bar</code>	Logical. Determines whether a progress bar is printed while bootstrapping.
<code>observed</code>	Logical. Indicates whether predictor variables were observed. See details.
<code>in_paren</code>	Logical. Whether the formatted string is to be reported in parentheses. If TRUE, parentheses in the formatted string (e.g., those enclosing degrees of freedom) are replaced with brackets.
<code>...</code>	Additional arguments passed to the function specified as <code>anova_fun</code> .

**Details**

As demonstrated by Algina, Keselman & Penfield (2007), asymptotic confidence intervals for  $\Delta R^2$  are often unreliable. Confidence intervals for model comparisons of `lm` objects are, therefore, estimated using their modified percentile bootstrap method. Note that the accuracy of the confidence intervals depends on the number of predictors  $p$ , their distribution, and the sample size  $n$ :

"When the predictor distribution is multivariate normal, one can obtain accurate CIs for  $\rho^2$  with  $n \geq 50$  when  $p = 3$ . For  $p = 6$  and for  $p = 9$ ,  $n \geq 100$  is advisable. When the predictor distribution is nonnormal in form, sample size requirements vary with type of nonnormality." (p. 939, Algina, Keselman & Penfield, 2010)

If **MBESS** is available, confidence intervals for  $R^2$  are computed using `MBESS::ci.R2()` to obtain a confidence region that corresponds to the confidence level `conf.int`, the default being a 90% CI (see Steiger, 2004). If `observed = FALSE`, it is assumed that predictors are fixed variables, i.e., "the values of the [predictors] were selected a priori as part of the research design" (p. 15, Kelly, 2007); put differently, it is assumed that predictors are not random. The confidence intervals for the regression coefficients in the model comparison table correspond to the  $\alpha$ -level chosen for  $R^2$  and  $\Delta R^2$  (e.g., 90% CI or  $\alpha = 0.10$  for  $R^2$  and  $\Delta R^2$  yields a 95% CI for regression coefficients, Steiger, 2004).

## Value

`apa_print()`-methods return a named list of class `apa_results` containing the following elements:

<code>estimate</code>	One or more character strings giving point estimates, confidence intervals, and confidence level. A single string is returned in a vector; multiple strings are returned as a named list. If no estimate is available the element is <code>NULL</code> .
<code>statistic</code>	One or more character strings giving the test statistic, parameters (e.g., degrees of freedom), and p-value. A single string is returned in a vector; multiple strings are returned as a named list. If no estimate is available the element is <code>NULL</code> .
<code>full_result</code>	One or more character strings comprised 'estimate' and 'statistic'. A single string is returned in a vector; multiple strings are returned as a named list.
<code>table</code>	A data.frame of class <code>apa_results_table</code> that contains all elements of <code>estimate</code> and <code>statistics</code> . This table can be passed to <code>apa_table()</code> for reporting.

Column names in `apa_results_table` are standardized following the **broom glossary** (e.g., `term`, `estimate`, `conf.int`, `statistic`, `df`, `df.residual`, `p.value`). Additionally, each column is labelled (e.g.,  $\hat{\eta}^2_G$  or  $t$ ) using the **tinylabels** package and these labels are used as column names when an `apa_results_table` is passed to `apa_table()`.

## References

- Algina, J., Keselman, H. J., & Penfield, R. D. (2007). Confidence intervals for an effect size measure in multiple linear regression. *Educational and Psychological Measurement*, 67(2), 207–218. doi: [doi:10.1177/0013164406292030](https://doi.org/10.1177/0013164406292030)
- Algina, J., Keselman, H. J., & Penfield, R. D. (2010). Confidence intervals for squared semipartial correlation coefficients: The effect of nonnormality. *Educational and Psychological Measurement*, 70(6), 926–940. doi: [doi:10.1177/0013164410379335](https://doi.org/10.1177/0013164410379335)
- Steiger (2004). Beyond the F test: Effect size confidence intervals and tests of close fit in the analysis of variance and contrast analysis. *Psychological Methods*, 9(2), 164–182. doi: [doi:10.1037/1082989X.9.2.164](https://doi.org/10.1037/1082989X.9.2.164)
- Kelley, K. (2007). Confidence intervals for standardized effect sizes: Theory, application, and implementation. *Journal of Statistical Software*, 20(8), 1–24. doi: [doi:10.18637/jss.v020.i08](https://doi.org/10.18637/jss.v020.i08)

**See Also**

`stats::anova()`

Other `apa_print`: `apa_print()`, `apa_print.BFBayesFactor()`, `apa_print.aov()`, `apa_print.emmGrid()`, `apa_print.glht()`, `apa_print.htest()`, `apa_print.lm()`, `apa_print.lme()`, `apa_print.merMod()`

**Examples**

```
mod1 <- lm(Sepal.Length ~ Sepal.Width, data = iris)
mod2 <- update(mod1, formula = . ~ . + Petal.Length)
mod3 <- update(mod2, formula = . ~ . + Petal.Width)

# No bootstrapped Delta R^2 CI
apa_print(list(Baseline = mod1, Length = mod2, Both = mod3), boot_samples = 0)
```

---

`apa_print.lm`

*Typeset Statistical Results from GLM*

---

**Description**

These methods take (generalized) linear model objects to create formatted character strings to report the results in accordance with APA manuscript guidelines.

**Usage**

```
## S3 method for class 'lm'
apa_print(
  x,
  est_name = NULL,
  standardized = FALSE,
  conf.int = 0.95,
  observed = TRUE,
  in_paren = FALSE,
  ...
)

## S3 method for class 'glm'
apa_print(
  x,
  est_name = NULL,
  standardized = FALSE,
  conf.int = 0.95,
  observed = TRUE,
  in_paren = FALSE,
  ...
)

## S3 method for class 'summary.glm'
```

```
apa_print(x, ...)

## S3 method for class 'summary.lm'
apa_print(x, ...)
```

### Arguments

x	glm or lm object.
est_name	Character. If NULL (the default) the name given in x (or a formally correct adaptation, such as "b*" instead of "b" for standardized regression coefficients) is used. Otherwise the supplied name is used. See details.
standardized	Logical. Indicates if coefficients were standardized (e.g., using <code>scale()</code> ), and leading zeros should be omitted if appropriate. See details.
conf.int	Numeric. Either a single value (range [0, 1]) giving the confidence level or a two-column matrix with confidence region bounds as column names (e.g. "2.5 %" and "97.5 %") and coefficient names as row names (in the same order as they appear in <code>summary(x)\$coefficients</code> ). See details.
observed	Logical. Indicates whether predictor variables were observed. See details.
in_paren	Logical. Whether the formatted string is to be reported in parentheses. If TRUE, parentheses in the formatted string (e.g., those enclosing degrees of freedom) are replaced with brackets.
...	Arguments passed on to <code>apa_num</code>

### Details

The coefficient names are sanitized to facilitate their use as list names. Parentheses are omitted and other non-word characters are replaced by `_` (see `sanitize_terms()`).

`est_name` is placed in the output string and is then passed to pandoc or LaTeX through `knitr`. Thus, to the extent it is supported by the final document type, you can pass LaTeX-markup to format the final text (e.g., "`\\beta`" yields  $\beta$ ).

If `standardized = TRUE`, `scale()` is removed from coefficient names (see examples). This option is currently ignored for `glm`-objects.

If `conf.int` is a single value, confidence intervals are calculated using `stats::confint()`.

If `x` is an `lm` object and the **MBESS** package is available, confidence intervals for  $R^2$  are computed using `MBESS::ci.R2()` to obtain a confidence region that corresponds to the  $\alpha$ -level chosen for the confidence intervals of regression coefficients (e.g., 95% CI or  $\alpha = 0.05$  for regression coefficients yields a 90% CI for  $R^2$ , see Steiger, 2004). If `observed = FALSE`, it is assumed that predictors are fixed variables, i.e., "the values of the [predictors] were selected a priori as part of the research design" (p. 15, Kelly, 2007); put differently, it is assumed that predictors are not random.

### Value

`apa_print()`-methods return a named list of class `apa_results` containing the following elements:

estimate	One or more character strings giving point estimates, confidence intervals, and confidence level. A single string is returned in a vector; multiple strings are returned as a named list. If no estimate is available the element is NULL.
statistic	One or more character strings giving the test statistic, parameters (e.g., degrees of freedom), and p-value. A single string is returned in a vector; multiple strings are returned as a named list. If no estimate is available the element is NULL.
full_result	One or more character strings comprised 'estimate' and 'statistic'. A single string is returned in a vector; multiple strings are returned as a named list.
table	A data.frame of class apa_results_table that contains all elements of estimate and statistics. This table can be passed to apa_table() for reporting.

Column names in apa\_results\_table are standardized following the **broom** glossary (e.g., term, estimate conf.int, statistic, df, df.residual, p.value). Additionally, each column is labelled (e.g.,  $\hat{\eta}^2_G$  or  $t$ ) using the **tinylabels** package and these labels are used as column names when an apa\_results\_table is passed to apa\_table().

## References

Steiger (2004). Beyond the F Test: Effect Size Confidence Intervals and Tests of Close Fit in the Analysis of Variance and Contrast Analysis. *Psychological Methods*, 9(2), 164-182. doi: [doi:10.1037/1082989X.9.2.164](https://doi.org/10.1037/1082989X.9.2.164)

Kelley, K. (2007). Confidence intervals for standardized effect sizes: Theory, application, and implementation. *Journal of Statistical Software*, 20(8), 1-24. doi: [doi:10.18637/jss.v020.i08](https://doi.org/10.18637/jss.v020.i08)

## See Also

[stats::confint\(\)](#), [MBESS::ci.pvaf\(\)](#)

Other apa\_print: [apa\\_print\(\)](#), [apa\\_print.BFBayesFactor\(\)](#), [apa\\_print.aov\(\)](#), [apa\\_print.emmGrid\(\)](#), [apa\\_print.glht\(\)](#), [apa\\_print.htest\(\)](#), [apa\\_print.list\(\)](#), [apa\\_print.lme\(\)](#), [apa\\_print.merMod\(\)](#)

## Examples

```
# Data from Dobson (1990), p. 9.
ctl <- c(4.17, 5.58, 5.18, 6.11, 4.50, 4.61, 5.17, 4.53, 5.33, 5.14)
trt <- c(4.81, 4.17, 4.41, 3.59, 5.87, 3.83, 6.03, 4.89, 4.32, 4.69)
group <- gl(2, 10, 20, labels = c("Ctl", "Trt"))
weight <- c(ctl, trt)
lm_fit <- lm(weight ~ group)

apa_print(lm_fit)

trt <- rep(trt, 2) # More data is always better
ctl <- rep(ctl, 2)
lm_fit2 <- lm(scale(trt) ~ scale(ctl))

apa_print(lm_fit2, standardized = TRUE)

# It is possible to simplify the regression table with transmute_df_into_label():
transmute_df_into_label(apa_print(lm_fit2, standardized = TRUE))
```

```
# Dobson (1990) Page 93: Randomized Controlled Trial :
counts <- c(18,17,15,20,10,20,25,13,12)
outcome <- gl(3,1,9)
treatment <- gl(3,3)
d.AD <- data.frame(treatment, outcome, counts)
glm.D93 <- glm(counts ~ outcome + treatment, family = poisson())

apa_print(glm.D93)
```

---

 apa\_print.lme

*Typeset Statistical Results from Nonlinear Hierarchical Models*


---

## Description

These methods take mixed-effects models fitted with the **nlme** package and create formatted character strings report the results in accordance with APA manuscript guidelines.

## Usage

```
## S3 method for class 'lme'
apa_print(x, conf.int = 0.95, in_paren = FALSE, est_name = NULL, ...)

## S3 method for class 'anova.lme'
apa_print(x, in_paren = FALSE, ...)
```

## Arguments

x	A (non-)linear mixed-effects model fitted with <code>nlme::lme()</code> or <code>nlme::nlme()</code>
conf.int	Numeric specifying the required confidence level <i>or</i> a named list of additional arguments that are passed to <code>nlme::intervals.lme()</code> .
in_paren	Logical. Whether the formatted string is to be reported in parentheses. If TRUE, parentheses in the formatted string (e.g., those enclosing degrees of freedom) are replaced with brackets.
est_name	An optional character. The label to be used for fixed-effects coefficients.
...	Further arguments that may be passed to <code>apa_num</code> to format estimates (i.e., columns estimate and conf.int).

## Value

`apa_print()`-methods return a named list of class `apa_results` containing the following elements:

estimate	One or more character strings giving point estimates, confidence intervals, and confidence level. A single string is returned in a vector; multiple strings are returned as a named list. If no estimate is available the element is NULL.
----------	--

statistic	One or more character strings giving the test statistic, parameters (e.g., degrees of freedom), and p-value. A single string is returned in a vector; multiple strings are returned as a named list. If no estimate is available the element is NULL.
full_result	One or more character strings comprised 'estimate' and 'statistic'. A single string is returned in a vector; multiple strings are returned as a named list.
table	A data.frame of class apa_results_table that contains all elements of estimate and statistics. This table can be passed to apa_table() for reporting.

Column names in apa\_results\_table are standardized following the **broom** glossary (e.g., term, estimate, conf.int, statistic, df, df.residual, p.value). Additionally, each column is labelled (e.g.,  $\hat{\eta}^2_G$  or  $t$ ) using the **tinylabels** package and these labels are used as column names when an apa\_results\_table is passed to apa\_table().

### See Also

Other apa\_print: `apa_print()`, `apa_print.BFBayesFactor()`, `apa_print.aov()`, `apa_print.emmGrid()`, `apa_print.glht()`, `apa_print.htest()`, `apa_print.list()`, `apa_print.lm()`, `apa_print.merMod()`

### Examples

```
library(nlme)
fm1 <- lme(distance ~ age, data = Orthodont, method = "ML") # random is ~ age
apa_print(fm1, conf.int = .9)
# ANOVA-like tables
single_anova <- anova(fm1)
apa_print(single_anova)
```

---

apa\_print.merMod

*Typeset Statistical Results from Hierarchical GLM*

---

### Description

These methods take objects from various R functions that calculate hierarchical (generalized) linear models to create formatted character strings to report the results in accordance with APA manuscript guidelines.

### Usage

```
## S3 method for class 'merMod'
apa_print(
  x,
  effects = "fixed",
  conf.int = 0.95,
  in_paren = FALSE,
  est_name = NULL,
  ...
)
```

```
)
## S3 method for class 'mixed'
apa_print(x, ...)
```

### Arguments

x	A fitted hierarchical (generalized) linear model, either from <code>lme4::lmer()</code> , <code>lmerTest::lmer()</code> , <code>afex::mixed()</code> , or <code>lme4::glmer()</code> .
effects	Character. Determines which information is returned. Currently, only fixed-effects terms ("fixed") are supported.
conf.int	Numeric specifying the required confidence level <i>or</i> a named list specifying additional arguments that are passed to <code>lme4::confint.merMod()</code> , see details.
in_paren	Logical. Whether the formatted string is to be reported in parentheses. If TRUE, parentheses in the formatted string (e.g., those enclosing degrees of freedom) are replaced with brackets.
est_name	An optional character. The label to be used for fixed-effects coefficients.
...	Further arguments that may be passed to <code>apa_num</code> to format estimates (i.e., columns estimate and conf.int).

### Details

Confidence intervals are calculated by calling `lme4::confint.merMod()`. By default, *Wald* confidence intervals are calculated, but this may change in the future.

### Value

`apa_print()`-methods return a named list of class `apa_results` containing the following elements:

estimate	One or more character strings giving point estimates, confidence intervals, and confidence level. A single string is returned in a vector; multiple strings are returned as a named list. If no estimate is available the element is NULL.
statistic	One or more character strings giving the test statistic, parameters (e.g., degrees of freedom), and p-value. A single string is returned in a vector; multiple strings are returned as a named list. If no estimate is available the element is NULL.
full_result	One or more character strings comprised 'estimate' and 'statistic'. A single string is returned in a vector; multiple strings are returned as a named list.
table	A <code>data.frame</code> of class <code>apa_results_table</code> that contains all elements of estimate and statistics. This table can be passed to <code>apa_table()</code> for reporting.

Column names in `apa_results_table` are standardized following the **broom** glossary (e.g., term, estimate conf.int, statistic, df, df.residual, p.value). Additionally, each column is labelled (e.g.,  $\hat{\eta}^2_G$  or  $t$ ) using the **tinylabels** package and these labels are used as column names when an `apa_results_table` is passed to `apa_table()`.

### See Also

Other `apa_print`: `apa_print()`, `apa_print.BFBayesFactor()`, `apa_print.aov()`, `apa_print.emmGrid()`, `apa_print.glht()`, `apa_print.htest()`, `apa_print.list()`, `apa_print.lm()`, `apa_print.lme()`



## Examples

```
# Fit a linear mixed model using the lme4 package
# or the lmerTest package (if dfs and p values are desired)
library(lmerTest)
fm1 <- lmer(Reaction ~ Days + (Days | Subject), sleepstudy)
# Format statistics for fixed-effects terms (the default)
apa_print(fm1)
```

---

apa\_print.papaja\_wsci *Typeset Within-Subjects Confidence Intervals*

---

## Description

This method takes an output object from `wsci` and creates a table and character strings to report means and within-subjects confidence intervals in a table or in text.

## Usage

```
## S3 method for class 'papaja_wsci'
apa_print(x, ...)
```

## Arguments

`x` An object of class `papaja_wsci`.  
`...` Arguments passed on to `apa_num`

## Value

`apa_print()`-methods return a named list of class `apa_results` containing the following elements:

<code>estimate</code>	One or more character strings giving point estimates, confidence intervals, and confidence level. A single string is returned in a vector; multiple strings are returned as a named list. If no estimate is available the element is <code>NULL</code> .
<code>statistic</code>	One or more character strings giving the test statistic, parameters (e.g., degrees of freedom), and p-value. A single string is returned in a vector; multiple strings are returned as a named list. If no estimate is available the element is <code>NULL</code> .
<code>full_result</code>	One or more character strings comprised 'estimate' and 'statistic'. A single string is returned in a vector; multiple strings are returned as a named list.
<code>table</code>	A <code>data.frame</code> of class <code>apa_results_table</code> that contains all elements of estimate and statistics. This table can be passed to <code>apa_table()</code> for reporting.

Column names in `apa_results_table` are standardized following the **broom** glossary (e.g., `term`, `estimate`, `conf.int`, `statistic`, `df`, `df.residual`, `p.value`). Additionally, each column is labelled (e.g.,  $\hat{\eta}^2_G$  or  $t$ ) using the **tinylabels** package and these labels are used as column names when an `apa_results_table` is passed to `apa_table()`.

---

`apa_table`*Prepare Table for Printing and Reporting*

---

**Description**

Formats matrices and data frames to report them as tables in R Markdown documents according to APA guidelines.

**Usage**

```
apa_table(x, ...)  
  
## Default S3 method:  
apa_table(x, ...)  
  
## S3 method for class 'apa_results_table'  
apa_table(x, escape = FALSE, ...)  
  
## S3 method for class 'apa_results'  
apa_table(x, ...)  
  
## S3 method for class 'matrix'  
apa_table(  
  x,  
  caption = NULL,  
  note = NULL,  
  stub_indents = NULL,  
  added_stub_head = NULL,  
  col_spanners = NULL,  
  midrules = NULL,  
  placement = "tbp",  
  landscape = FALSE,  
  font_size = NULL,  
  escape = TRUE,  
  span_text_columns = TRUE,  
  ...,  
  format.args = NULL  
)  
  
## S3 method for class 'list'  
apa_table(  
  x,  
  caption = NULL,  
  note = NULL,  
  stub_indents = NULL,  
  added_stub_head = NULL,  
  col_spanners = NULL,
```

```

midrules = NULL,
placement = "tbp",
landscape = FALSE,
font_size = NULL,
escape = TRUE,
merge_method = "indent",
span_text_columns = TRUE,
...,
format.args = NULL
)

## S3 method for class 'data.frame'
apa_table(
  x,
  caption = NULL,
  note = NULL,
  stub_indents = NULL,
  added_stub_head = NULL,
  col_spanners = NULL,
  midrules = NULL,
  placement = "tbp",
  landscape = FALSE,
  font_size = NULL,
  escape = TRUE,
  span_text_columns = TRUE,
  ...,
  format.args = NULL
)

```

## Arguments

**x** Object to print, either a matrix, data.frame, or list. See details.

**...** Arguments passed on to `knitr::kable`

**format** A character string. Possible values are latex, html, pipe (Pandoc's pipe tables), simple (Pandoc's simple tables), rst, jira, and org (Emacs Org-mode). The value of this argument will be automatically determined if the function is called within a **knitr** document. The format value can also be set in the global option `knitr.table.format`. If format is a function, it must return a character string.

**digits** Maximum number of digits for numeric columns, passed to `round()`. This can also be a vector of length `ncol(x)`, to set the number of digits for individual columns.

**row.names** Logical: whether to include row names. By default, row names are included if `rownames(x)` is neither NULL nor identical to `1:nrow(x)`.

**col.names** A character vector of column names to be used in the table.

**align** Column alignment: a character vector consisting of 'l' (left), 'c' (center) and/or 'r' (right). By default or if `align = NULL`, numeric columns are right-aligned, and other columns are left-aligned. If `length(align) ==`

	1L, the string will be expanded to a vector of individual letters, e.g. 'clc' becomes c('c', 'l', 'c'), unless the output format is LaTeX.
	label The table reference label. By default, the label is obtained from <code>knitr::opts_current\$get('label')</code> (i.e., the current chunk label). To disable the label, use <code>label = NA</code> .
escape	Logical. If TRUE special LaTeX characters, such as % or _, in column names, row names, caption, note and table contents are escaped.
caption	Character. Caption to be printed above the table.
note	Character. Note to be printed below the table.
stub_indents	List. A named list of vectors that contain indices of rows to indent. The name of each list element containing the vector is used as title for indented sections.
added_stub_head	Character. Used as stub head (name of first column) if <code>row.names = TRUE</code> is passed to <code>kable</code> ; ignored if row names are omitted from the table.
col_spanners	List. A named list of vectors of length 2 that contain the indices of the first and last column to span. The name of each list element is used as grouping column name. Currently ignored in Word documents.
midrules	Numeric. Vector of line numbers in table (not counting column headings) that should be followed by a horizontal rule; currently ignored in Word documents.
placement	Character. Indicates whether table should be placed, for example, at the current location (h), at the top (t), bottom (b), or on a separate page (p). Arguments can be combined to indicate order of preference (htb); currently ignored when <code>longtable = TRUE</code> , <code>landscape = TRUE</code> , and in Word documents.
landscape	Logical. If TRUE the table is printed in landscape mode; currently ignored in Word documents.
font_size	Character. Font size to use for table contents (can be tiny, scriptsize, footnotesize, small, normalsize (default), large, Large, LARGE, huge, Huge). Ignored in Word documents.
span_text_columns	Logical. If TRUE tables span across text columns in two-column PDF documents (e.g. when setting <code>classoption: jou</code> ). Otherwise ignored.
format.args	List. A named list of arguments to be passed to <code>apa_num</code> to format numeric values.
merge_method	Character. Determines how to merge tables if <code>x</code> is a list of matrices or data frames with a common structure. Can be either <code>indent</code> or <code>table_spanner</code> . See details.

## Details

When using `apa_table`, the type of the output (LaTeX or Word) is determined automatically by the rendered document type. In interactive R session the output defaults to LaTeX.

If `x` is a list, all list elements are merged by columns into a single table and the names of list elements are added according to the setting of `merge_method`.

By default, the width of the table is set to accommodate its contents. In some cases, this may cause the table to exceed the page width. To address this, tables can be rotated 90 degrees by setting

landscape = TRUE or, by explicitly using "paragraph columns" with fixed column widths, such that the contents is automatically broken into multiple lines. For example, set align = "lm{5cm}l" to limit the second column to a width of 5 cm. Similarly, to space columns equally use align = paste0("m{", 1/(ncol(x) + 1), "\\linewidth}")

Note that placement options are not supported in appendices of apa6 documents and will be printed to the document. To omit the printed options set placement = NULL.

### Value

A character vector of the table source code of class `knitr_asis`, see `knitr::asis_output()`.

### See Also

`knitr::kable()`, `apa_num()`

### Examples

```
my_table <- t(apply(cars, 2, function(x) # Create data
  round(c(Mean = mean(x), SD = sd(x), Min = min(x), Max = max(x)), 2)
))
```

```
apa_table(
  my_table
  , align = c("l", rep("r", 3))
  , caption = "A summary table of the cars dataset."
)
```

```
apa_table(
  cbind(my_table, my_table)
  , align = c("l", rep("r", 8))
  , caption = "A summary table of the cars dataset."
  , note = "This table was created using apa\\_table()"
  , added_stub_head = "Variables"
  , col_spanners = list(`Cars 1` = c(2, 5), `Cars 2` = c(6, 9))
)
```

```
apa_table(
  list(`Cars 1` = my_table, `Cars 2` = my_table)
  , caption = "A summary table of the cars dataset."
  , added_stub_head = "Variables"
)
```

### Description

Creates one or more violin plots from a data.frame containing data from a factorial design and sets APA-friendly defaults.

**Usage**

```
apa_violinplot(data, ...)

## Default S3 method:
apa_violinplot(
  data,
  id,
  factors = NULL,
  dv,
  tendency = mean,
  dispersion = conf_int,
  level = 0.95,
  fun_aggregate = mean,
  na.rm = TRUE,
  use = "all.obs",
  intercept = NULL,
  args_x_axis = NULL,
  args_y_axis = NULL,
  args_title = NULL,
  args_points = NULL,
  args_lines = NULL,
  args_error_bars = NULL,
  args_legend = NULL,
  jit = 0.3,
  xlab = NULL,
  ylab = NULL,
  main = NULL,
  ...
)

## S3 method for class 'afex_aov'
apa_violinplot(
  data,
  tendency = mean,
  dispersion = conf_int,
  fun_aggregate = mean,
  ...
)
```

**Arguments**

<code>data</code>	A data.frame that contains the data, or an object of class <code>afex_aov</code> .
<code>...</code>	Arguments passed on to <a href="#">apa_factorial_plot</a>
<code>id</code>	Character. Variable name that identifies subjects.
<code>factors</code>	Character. A vector of up to four variable names that is used to stratify the data.
<code>dv</code>	Character. The name of the dependent variable.

tendency	Closure. A function that will be used as measure of central tendency.
dispersion	Closure. A function that will be used to construct error bars (i.e., whiskers). Defaults to <code>conf_int()</code> for 95% between-subjects confidence intervals. See details for more options, especially for within-subjects confidence intervals.
level	Numeric. Defines the width of the interval if confidence intervals are plotted. Defaults to 0.95. for 95% confidence intervals. Ignored if <code>dispersion</code> is not a confidence-interval function. See details.
fun_aggregate	Closure. The function that will be used to aggregate observations within subjects and factors before calculating descriptive statistics for each cell of the design. Defaults to <code>mean</code> .
na.rm	Logical. Specifies if missing values are removed. Defaults to <code>TRUE</code> .
use	Character. Specifies a method to exclude cases if there are missing values <i>after</i> aggregating. Possible options are "all.obs" or "complete.obs".
intercept	Numeric. Adds a horizontal line at height <code>intercept</code> to the plot. Can be either a single value or a matrix. For the matrix case, multiple lines are drawn, where the dimensions of the matrix determine the number of lines to be drawn.
args_x_axis	An optional list that contains further arguments that may be passed to <code>axis()</code> for customizing the <i>x</i> axis.
args_y_axis	An optional list that contains further arguments that may be passed to <code>axis()</code> for customizing the <i>y</i> axis.
args_title	An optional list that contains further arguments that may be passed to <code>title()</code> .
args_points	An optional list that contains further arguments that may be passed to <code>points()</code> .
args_lines	An optional list that contains further arguments that may be passed to <code>lines()</code> .
args_error_bars	An optional list that contains further arguments that may be passed to <code>arrows()</code> .
args_legend	An optional list that contains further arguments that may be passed to <code>legend()</code> .
jit	Numeric. Determines the amount of horizontal displacement. Defaults to 0.3, defaults to 0.4 if <code>plot = "bars"</code> .
xlab	Character or expression. Label for <i>x</i> axis.
ylab	Character or expression. Label for <i>y</i> axis.
main	Character or expression. For up to two factors, simply specify the main title. If you stratify the data by more than two factors, either specify a single value that will be added to automatically generated main title, <i>or</i> specify an array of multiple titles, one for each plot area.

## Details

The measure of dispersion can be either `conf_int()` for between-subjects confidence intervals, `se()` for standard errors, or any other standard function. For within-subjects confidence intervals, specify `wsci()` or `within_subjects_conf_int()`.

If between- or within-subjects confidence intervals are requested, you can also specify the area of the cumulative distribution function that will be covered. For instance, if you want a 98% confidence interval, specify `level = 0.98`. The default is `level = 0.95` for 95% confidence intervals.

**Customization of plot elements:**

`apa_factorial_plot()` and its descendants `apa_barplot()`, `apa_lineplot()`, `apa_beeplot()`, and `apa_violinplot()` are wrapper functions that sequentially call:

- `plot.new()`,
- `plot.window()`,
- `axis()` (once for  $x$  axis, once for  $y$  axis),
- `title()` for axis labels and titles,
- `rect()` for bars in bar plots,
- `points()` for bee swarms,
- `density()` and `polygon()` for violins,
- `lines()` for lines connecting central tendency points,
- `arrows()` for error bars,
- `points()` for tendency points,
- `legend()` for a legend, and
- `lines()` for intercepts.

These calls can be customized by setting the respective parameters `args_*** = list(...)`.

**Value**

A named (nested) list of plot options including raw and derived data. *Note that the structure of the return value is about to change in a forthcoming release of papaja.*

**See Also**

Other plots for factorial designs: `apa_barplot()`, `apa_beeplot()`, `apa_factorial_plot()`, `apa_lineplot()`

**Examples**

```
apa_violinplot(
  data = npk
  , id = "block"
  , dv = "yield"
  , factors = c("N")
)

apa_violinplot(
  data = npk
  , id = "block"
  , dv = "yield"
  , factors = c("N", "P")
  , args_legend = list(x = "center")
  , jit = 0.1
)
```



---

beautify_terms	<i>Prettify Term Names</i>
----------------	----------------------------

---

### Description

Remove parentheses, replace colons with  $\times$ . Useful to prettify term names in `apa_print()` tables.

### Usage

```
beautify_terms(x, ...)  
  
## S3 method for class 'character'  
beautify_terms(x, standardized = FALSE, retain_period = FALSE, ...)  
  
## S3 method for class 'numeric'  
beautify_terms(x, standardized = FALSE, ...)  
  
## S3 method for class 'factor'  
beautify_terms(x, standardized = FALSE, ...)  
  
## S3 method for class 'data.frame'  
beautify_terms(x, ...)
```

### Arguments

x	Character. Vector of term names to be prettified.
...	Additional arguments passed to <code>apa_num()</code> , for numeric values in x, ignored otherwise.
standardized	Logical. If TRUE, the name of the function <code>scale()</code> will be removed from term names.
retain_period	Logical. If TRUE, any periods in term names will be retained, otherwise they will be replaced by a space.

### Value

A character vector or data.frame (if x is a data.frame) containing term names modified for nicer printing.

### Examples

```
beautify_terms("a:b")  
beautify_terms("scale(x)", standardized = TRUE)  
beautify_terms("snake_case")
```

## Description

Creates character strings to cite R and R packages.

## Usage

```
cite_r(  
  file = NULL,  
  prefix = "R-"  
  footnote = FALSE,  
  pkgs = NULL,  
  omit = TRUE,  
  ...  
)
```

## Arguments

<code>file</code>	Character. Path and name of the <code>.bib</code> -file holding the references. If <code>NULL</code> , only R is cited.
<code>prefix</code>	Character. Prefix used for all R-package reference handles.
<code>footnote</code>	Logical. Indicates if packages should be cited in a footnote. Ignored if no package information is available.
<code>pkgs</code>	Character. Vector of package names to cite or omit depending on <code>omit</code> .
<code>omit</code>	Logical. If <code>TRUE</code> , <code>pkgs</code> constitutes a list of packages <i>not</i> to cite (a blacklist). If <code>FALSE</code> , <code>pkgs</code> constitutes a list of packages to cite (a whitelist).
<code>...</code>	Additional arguments, which are currently ignored.

## Details

If `footnote = FALSE`, a character string citing R and R packages including version numbers is returned. Otherwise a named list with the elements `r` and `pkgs` is returned. The former element holds a character string citing R and a reference to a footnote; the latter element contains a character string that creates the footnote. For correct rendering, the footnote string needs to be a separate paragraph in the R Markdown document.

## Value

If `footnote = FALSE` a character string is returned, otherwise a named list with the elements `r` and `pkgs`.

## See Also

[r\\_refs\(\)](#), [knitr::write\\_bib\(\)](#)

**Examples**

```
cite_r()
```

---

conf_int	<i>Between-Subjects Confidence Intervals</i>
----------	--

---

**Description**

Calculates the deviation that is needed to construct confidence intervals for a vector of observations.

**Usage**

```
conf_int(x, level = 0.95, na.rm = TRUE)
```

```
conf.int(x, level = 0.95, na.rm = TRUE)
```

```
ci(x, level = 0.95, na.rm = TRUE)
```

**Arguments**

x	Numeric. A vector of observations from your dependent variable.
level	Numeric. Defines the width of the interval if confidence intervals are plotted. Defaults to 0.95 for 95% confidence intervals.
na.rm	Logical. Specifies if missing values should be removed.

**Value**

Returns a single numeric value, the deviation of the symmetric confidence bounds from the mean based on the t distribution.

---

fetch_web_refs	<i>Fetch a .bib-reference file from the web (defunct)</i>
----------------	---

---

**Description**

Downloads and saves a .bib-reference file from the web, so it can be used to cite references in a Markdown-document using pandoc or LaTeX. *This function has been defunct. Please use download from the **downloader** instead.*

**Usage**

```
fetch_web_refs(x, bib_name)
```

**Arguments**

x	Character. URL of the .bib-file to fetch.
bib_name	Character. The path and name of the file to be created.

**Details**

If the function is called in an RMarkdown-document the file name specified as bib\_name can be used in the YAML header as bibliography.

**Value**

Returns NULL invisibly.

**See Also**

[cite\\_r\(\)](#), [r\\_refs\(\)](#), [knitr::write\\_bib\(\)](#)

---

fetch_zotero_refs	<i>Save a collection from a Zotero-Account to a BibTeX-file (defunct)</i>
-------------------	---

---

**Description**

Downloads and saves a Zotero reference library (or a subset) and saves it as BibTeX file. *This function has been defunct. Use ReadZotero() from the **RefManagerR** package instead.*

**Usage**

```
fetch_zotero_refs(
  x,
  bib_name,
  API_key = NULL,
  collection = NULL,
  lib_type = "user"
)
```

**Arguments**

x	Character. Zotero user or group ID, see details.
bib_name	Character. Name of the BibTeX-file references are saved to.
API_key	Character. Zotero API key, see details.
collection	Character. Optional ID of a collection in the Zotero library, see details.
lib_type	Character. Specifies if the supplied ID is associated with a Zotero user or group.

## Details

This function retrieves references through the Zotero web API. `x` takes a Zotero user or group ID that can be retrieved from the Zotero.org user or group Feeds/API settings. An authentication key (API\_key) is required to access nonpublic Zotero libraries. Authentication keys can also be generated in the Zotero.org user or group Feeds/API settings.

If the requested reference collection is larger than 100 records, multiple API calls are initiated because the number of retrieved records is limited to 100 per API call. Frequent API calls will result in a temporary access block. Thus, there is an (currently unknown) upper limit to the length of reference collections that can be retrieved through this function. It is generally recommended to comment out calls to this function in R Markdown documents during periods of frequent knitting to limit the number of API calls and limit the number of references to those needed for the current document by setting up collections in your Zotero library.

Collection keys (collection), i.e. identifiers of reference library subsets, can be retrieved by accessing them via a web browser. They keys are contained in the URL:

`https://www.zotero.org/<USERNAME>/items/collectionKey/<COLLECTIONKEY>`

Zotero web API calls can be slow, especially for large reference collections. If available, this function will use the **downloader**-package, which speeds up reference downloads considerably.

## Value

Returns bib\_name invisibly.

## Author(s)

Christoph Stahl, Frederik Aust

## See Also

[cite\\_r\(\)](#), [r\\_refs\(\)](#)

---

generate\_author\_yaml    *generate\_author\_yaml*

---

## Description

This function helps organize YAML author and affiliation fields such that authorship order can be changed without having to also update the order of affiliations.

## Usage

```
generate_author_yaml(  
  researchers,  
  affiliations,  
  corres_name,  
  corres_address,  
  corres_email  
)
```

**Arguments**

researchers	a list of named character vectors. Author information is stored in the name of the vector. Abbreviated affiliations are stored in the vector.
affiliations	a list of named character strings. Abbreviated affiliations are stored in the names, the full affiliation is stored in the string.
corres_name	A character string. The name of the corresponding author, must match the author details in the researcher argument identically.
corres_address	A character string. The address of the corresponding author
corres_email	A character string. The email of the corresponding author.

**Examples**

```
library(papaja)

generate_author_yaml (
  researchers = list(
    "Emma J. Citizen" = c("example_hospital", "example_college"),
    "John H. Smith" = "example_college",
    "Kate C. Jones" = "example_hospital"
  ),
  affiliations = list(
    "example_hospital" = "Southern Example Hospital, NSW, Australia",
    "example_college" = "New Example College, VIC, Australia"
  ),
  corres_name = "Emma J. Citizen",
  corres_address = "123 Example Street, Epping, NSW 2121",
  corres_email = "jane@example.com"
)
```

---

glue\_apa\_results

*Create a New apa\_results Object*


---

**Description**

Typeset the contents of an object according to the specified expression strings and create a new or extend an existing `apa_results` object.

**Usage**

```
glue_apa_results(x = NULL, term_names = NULL, ...)

add_glue_to_apa_results(
  ...,
  est_glue,
  stat_glue,
  container,
  sublist = NULL,
```

```

    term_names = NULL,
    in_paren = FALSE,
    est_first = TRUE,
    simplify = TRUE
  )

```

## Arguments

<code>x</code>	An environment, list or data frame used to look up values for substitution.
<code>term_names</code>	Character. Used as names for the <code>estimate</code> -, <code>statistics</code> -, and <code>full_result</code> sub-lists, if multiple estimates or statistics are glued. Defaults to <code>attr(x, "sanitized_term_names")</code> .
<code>...</code>	[expressions] Unnamed arguments are taken to be expression string(s) to format. Multiple inputs are concatenated together before formatting. Named arguments are taken to be temporary variables available for substitution.  For <code>glue_data()</code> , elements in <code>...</code> override the values in <code>x</code> .
<code>est_glue</code>	Character. (Named vector of) expressions string(s) to format. Each string creates a new (named) element in the <code>estimate</code> sub-list.
<code>stat_glue</code>	Character. (Named vector of) expressions string(s) to format. Each string creates a new (named) element in the <code>statistic</code> sub-list.
<code>container</code>	List of class <code>apa_results</code> to add the glued results to.
<code>sublist</code>	Character. Name of (new) sub-list in <code>estimate</code> <code>statistics</code> , and <code>full_result</code> to append glued results to (e.g., <code>modelfit</code> ).
<code>in_paren</code>	Logical. Whether the formatted string is to be reported in parentheses. If <code>TRUE</code> , parentheses in the formatted string (e.g., those enclosing degrees of freedom) are replaced with brackets.
<code>est_first</code>	Logical. Determines in which order <code>estimate</code> and <code>statistic</code> are glued together to <code>full_result</code> .
<code>simplify</code>	Logical. Determines whether the <code>estimate</code> , <code>statistic</code> , and <code>full_result</code> sub-lists should be simplified if only one term is available from the model object.

## Value

Returns a list of class `apa_results`, see [apa\\_print\(\)](#).

## Examples

```

# Tidy and typeset output
iris_lm <- lm(Sepal.Length ~ Petal.Length + Petal.Width, iris)
tidy_iris_lm <- broom::tidy(iris_lm, conf.int = TRUE)
tidy_iris_lm$p.value <- apa_p(tidy_iris_lm$p.value)

glance_iris_lm <- broom::glance(iris_lm)
glance_iris_lm$p.value <- apa_p(glance_iris_lm$p.value, add_equals = TRUE)
glance_iris_lm$df <- apa_num(as.integer(glance_iris_lm$df))
glance_iris_lm$df.residual <- apa_num(as.integer(glance_iris_lm$df.residual))

```

```

# Create `apa_results`-list
lm_results <- glue_apa_results(
  x = tidy_iris_lm
  , df = glance_iris_lm$df.residual
  , est_glue = "$b = <<estimate>>, 95% CI $[<<conf.low>>,~<<conf.high>>]$"
  , stat_glue = "$t(<<df>>) = <<statistic>>$, $p <<p.value>> $"
  , term_names = make.names(names(coef(iris_lm)))
)

# Add modelfit information
add_glue_to_apa_results(
  .x = glance_iris_lm
  , container = lm_results
  , sublist = "modelfit"
  , est_glue = c(
    r2 = "$R^2 = <<r.squared>> $"
    , aic = ""
  )
  , stat_glue = c(
    r2 = "$F(<<df>>, <<df.residual>>) = <<statistic>>$, $p <<add_equals(p.value)>> $"
    , aic = "$\\mathrm{AIC} = <<AIC>> $"
  )
)

```

---

hd\_int

*Highest-Density Intervals*


---

## Description

Calculates the highest-density interval of a vector of values.

## Usage

```
hd_int(x, level = 0.95)
```

## Arguments

x	Numeric. A vector of observations.
level	Numeric. Defines the width of the interval. Defaults to 95% highest-density intervals.



---

in\_paren *Replace Parentheses with Brackets*

---

### Description

Takes a single character or a list of characters and replaces parentheses with brackets. Can be used to prepare a string of statistics (e.g. containing degrees of freedom) for reporting within parentheses.

### Usage

```
in_paren(x)
```

### Arguments

x                      Character. Single character or list of characters.

### Value

An object of the same type as x, where all parentheses have been replaced by brackets.

### See Also

[apa\\_print\(\)](#)

### Examples

```
t_stat <- t.test(extra ~ group, data = sleep)
t_test_res <- apa_print(t_stat)
in_paren(t_test_res$stat)
in_paren(t_test_res[1:3])
```

---

papaja *Prepare APA Journal Articles with R Markdown*

---

### Description

papaja is an **award-winning** R package that facilitates creating computationally reproducible, submission-ready manuscripts which conform to the American Psychological Association (APA) manuscript guidelines (6th Edition).

### Details

papaja provides

- an **R Markdown** template that can be used with (or without) **RStudio** to create PDF documents (using the **apa6** LaTeX class) or Word documents (using a .docx-reference file).
- Functions to *typeset* the results from *statistical analyses* (e.g., [apa\\_print\(\)](#)),
- functions to create *tables* ([apa\\_table\(\)](#)), and
- functions to create *figures* in accordance with APA guidelines (e.g., [apa\\_factorial\\_plot\(\)](#)).

## System requirements

To use papaja you need either an up-to-date version of **RStudio** or **pandoc**. If you want to create PDF- in addition to DOCX-documents you additionally need a **TeX** distribution. We recommend **TinyTeX**, which can be installed from within R via the **tinytex** package.

Please refer to the **papaja manual** for detailed installation instructions.

## Getting help

For a comprehensive introduction to papaja, see the current draft of the **manual**. If you have a specific question that is not answered in the manual, feel free to ask a question on Stack Overflow **using the papaja tag**. If you believe you have found a bug or would like to request a new feature, **open an issue** on Github and provide a **minimal complete verifiable example**.

## Authors

Frederik Aust (frederik.aust at uni-koeln.de). Marius Barth (marius.barth at uni-koeln.de).

## Maintainer

Frederik Aust (frederik.aust at uni-koeln.de).

## Author(s)

**Maintainer:** Frederik Aust <frederik.aust@uni-koeln.de> (**ORCID**)

Authors:

- Marius Barth <marius.barth@uni-koeln.de> (**ORCID**)

Other contributors:

- Birk Diedenhofen <mail@birkdiedenhofen.de> [contributor]
- Christoph Stahl <christoph.stahl@uni-koeln.de> [contributor]
- Joseph V. Casillas <joseph.casillas@rutgers.edu> [contributor]
- Rudolf Siegel <rudolf.siegel@uni-saarland.de> [contributor]

## See Also

Useful links:

- <https://github.com/crsh/papaja>
- Report bugs at <https://github.com/crsh/papaja/issues>

---

quote_from_tex	<i>Quote from TeX document</i>
----------------	--------------------------------

---

### Description

Includes a labelled quote from a LaTeX document 'asis'.

### Usage

```
quote_from_tex(
  x,
  file = paste0(rmarkdown::metadata[["manuscript-tex"]], ".tex")
)
```

### Arguments

x	Character. One or more quote labels.
file	Character. Path to LaTeX file from which to quote.

### Details

Searches the LaTeX document specified in file for labelled quotes, i.e. paragraphs that are enclosed in % <@~{#quote-label} and % ~@> tags in LaTeX comments on separate lines. The labelled quote is then inserted and rendered asis. To use labelled quote-tags in a `apa6_pdf()`-document, set the YAML front matter options `quote_labels: true`.

### Value

A character vector of LaTeX document text of class `knit_asis`, see `knitr::asis_output()`.

---

remove_comments	<i>Remove Comments</i>
-----------------	------------------------

---

### Description

Removes markdown comments from an R Markdown file.

### Usage

```
remove_comments(x, file)
```

### Arguments

x	Character. Path to an R Markdown file.
file	Character. Name of the new R Markdown file without comments.

**Value**

No return value, called to write text to file.

---

revision\_letter\_pdf    *Revision Letter*

---

**Description**

Template for creating a journal revision letters.

**Usage**

```
revision_letter_pdf(...)
```

**Arguments**

...                    Arguments passed on to [bookdown::pdf\\_document2](#)

**Details**

This document format is adapted from by the [revision letter template](#) by Martin Schrön.

It is possible to reference sections, figures, or tables in the revised manuscript, either by their number or by page. To do so, specify a path to the revised manuscript (omitting the file extension) in the YAML front matter (i.e., `manuscript-tex: file_name`) and ensure that you retain the aux file when rendering the revised manuscript. To do so, set the following option in a code chunk of the revised manuscript: `options(tinytex.clean = FALSE)`. To reference section, figure, or table numbers it is possible to use LaTeX (i.e., `\ref{label}`) or **bookdown** cross-referencing syntax (i.e., `\@ref(label)`). To reference the corresponding page numbers you must use the LaTeX syntax (i.e., `\pageref{label}`).

To quote entire paragraphs directly from the revised manuscript see [quote\\_from\\_tex\(\)](#).

**Value**

R Markdown output format to pass to [rmarkdown::render\(\)](#).

**See Also**

[bookdown::pdf\\_document2\(\)](#)

## Description

Creates a .bib-reference file for the installed R version and R-packages, so they can be cited in an R Markdown-document.

## Usage

```
r_refs(  
  file,  
  append = TRUE,  
  prefix = "R-"  
  type_pref = c("Article", "Book"),  
  tweak = TRUE  
)  
  
create_bib(  
  x,  
  file,  
  append = TRUE,  
  prefix = "R-"  
  type_pref = c("Article", "Book"),  
  tweak = TRUE  
)
```

## Arguments

file	Character. Path and name of the file to be created or updated.
append	Logical. Indicates if existing bibliography should be complemented or overwritten. See details.
prefix	Character. Prefix for all R-package reference handles.
type_pref	Character. A vector of BibTeX entry types in the order by which to prioritize packages CITATION entries. See details.
tweak	Logical. Indicates whether to fix some known problems in citations (based on <a href="#">write_bib</a> ).
x	Character. Names of packages to include in bibliography.

## Details

r\_refs is a wrapper for [create\\_bib](#) to create a bibliography for R and all attached or cached packages.

By default, if a file exists at the specified location, r\_refs reads the file and appends missing citation information to the end of the file ([create\\_bib](#) always overwrites existing files). It is recommended to use a bibliography-file dedicated to R-references.

**Beware that chunks loading packages should generally *not* be cached.** `r_refs` will make all packages loaded in cached chunks citable, but it won't know when you remove a package from a cached chunk. This can result in unused package references in your bibliography-file that will be cited when using `cite_r`.

If a package provides citation information in a CITATION file, a reference is selected based on the preferred order of reference types specified in `type_pref`. By default, available articles are cited rather than books. If no reference of the specified types is available, the first reference is used. If multiple references of the preferred type are given all of them are cited. Finally, if no CITATION file exists a reference is generated from the DESCRIPTION file by `citation`.

### Value

Invisibly returns the bibliography written to file.

### See Also

`cite_r()`, `knitr::write_bib()`, `utils::citation()`, `utils::toLatex()`

---

sanitize_terms	<i>Sanitize Term Names</i>
----------------	----------------------------

---

### Description

Remove characters from term names that will be difficult to address using the `$`-operator. *This function is not exported.*

### Usage

```
sanitize_terms(x, standardized = FALSE)

## S3 method for class 'character'
sanitize_terms(x, standardized = FALSE)

## S3 method for class 'factor'
sanitize_terms(x, standardized = FALSE)

## S3 method for class 'data.frame'
sanitize_terms(x, standardized = FALSE)

## S3 method for class 'list'
sanitize_terms(x, standardized = FALSE)
```

### Arguments

<code>x</code>	Character. Vector of term names to be sanitized.
<code>standardized</code>	Logical. If TRUE, the name of the function [ <code>scale()</code> ] will be removed from term names.

**Value**

An object of the same class as `x` containing sanitized term names as characters.

**Examples**

```
sanitize_terms(c("(Intercept)", "Factor A", "Factor B", "Factor A:Factor B", "scale(FactorA)"))
```

---

se	<i>Standard Error of the Mean</i>
----	-----------------------------------

---

**Description**

Calculates the standard error of the mean.

**Usage**

```
se(x, na.rm = TRUE)
```

**Arguments**

x	Numeric. A vector of observations.
na.rm	Logical. Specifies if missing values should be removed.

**Value**

The standard error of the mean as numeric vector of length 1.

---

simple_codebook	<i>Simple Codebook</i>
-----------------	------------------------

---

**Description**

Generate a simple codebook in CSV-format from a (labelled) `data.frame`.

**Usage**

```
simple_codebook(x, ...)
```

**Arguments**

- `x` `data.frame`. Data to be documented.
- `...` Arguments passed on to `utils::write.table`
- `file` either a character string naming a file or a `connection` open for writing. "" indicates output to the console.
- `append` logical. Only relevant if `file` is a character string. If TRUE, the output is appended to the file. If FALSE, any existing file of the name is destroyed.
- `quote` a logical value (TRUE or FALSE) or a numeric vector. If TRUE, any character or factor columns will be surrounded by double quotes. If a numeric vector, its elements are taken as the indices of columns to quote. In both cases, row and column names are quoted if they are written. If FALSE, nothing is quoted.
- `sep` the field separator string. Values within each row of `x` are separated by this string.
- `eol` the character(s) to print at the end of each line (row). For example, `eol = "\r\n"` will produce Windows' line endings on a Unix-alike OS, and `eol = "\r"` will produce files as expected by Excel:mac 2004.
- `na` the string to use for missing values in the data.
- `dec` the string to use for decimal points in numeric or complex columns: must be a single character.
- `row.names` either a logical value indicating whether the row names of `x` are to be written along with `x`, or a character vector of row names to be written.
- `col.names` either a logical value indicating whether the column names of `x` are to be written along with `x`, or a character vector of column names to be written. See the section on 'CSV files' for the meaning of `col.names = NA`.
- `qmethod` a character string specifying how to deal with embedded double quote characters when quoting strings. Must be one of "escape" (default for `write.table`), in which case the quote character is escaped in C style by a backslash, or "double" (default for `write.csv` and `write.csv2`), in which case it is doubled. You can specify just the initial letter.
- `fileEncoding` character string: if non-empty declares the encoding to be used on a file (not a connection) so the character data can be re-encoded as they are written. See [file](#).

**Details**

If the `skimr` package is installed, an in-line histogram is added for all numeric variables. If columns are labelled, the labels are included in the codebook.

**Value**

Returns NULL invisibly.

**See Also**

[utils::write.csv\(\)](#)



**Examples**

```
variable_labels(cars) <- c(speed = "Speed [ft/s]", dist = "Distance traveled [m]")
simple_codebook(cars, file = file.path(tempdir(), "cars_codebook.csv"))
```

---

 sort\_terms

*Sort ANOVA or Regression Table by Predictors/Effects*


---

**Description**

Sort rows in ANOVA or regression tables produced by [apa\\_print\(\)](#) by complexity (i.e., main effects, two-way interactions, three-way interactions, etc.).

**Usage**

```
sort_terms(x, colname)
```

**Arguments**

`x` A data frame. For example, the table element produced by [apa\\_print\(\)](#).  
`colname` Character. Column name of the data.frame containing the terms to sort.

**Value**

Returns the same data.frame with reordered rows.

**Examples**

```
## From Venables and Ripley (2002) p. 165.
npk_aov <- aov(yield ~ block + N * P * K, npk)
npk_aov_results <- apa_print(npk_aov)
sort_terms(npk_aov_results$table, "term")
```

---

 summary.papaja\_wsci

*Summarize Within-Subjects Confidence Intervals*


---

**Description**

Calculate upper and lower limits of within-subjects confidence intervals calculated with [wsci\(\)](#) and return them along their respective means.

**Usage**

```
## S3 method for class 'papaja_wsci'
summary(object, ...)
```

**Arguments**

object            An object of class `papaja_wsci`, generated with function `wsci()`.  
 ...              Further arguments that may be passed, currently ignored.

**Value**

A data frame containing means as well as lower and upper confidence bounds for each cell of the design.

---

theme_apa	<i>APA-style ggplot2 Theme</i>
-----------	--------------------------------

---

**Description**

**ggplot2** theme with a white panel background, no grid lines, large axis and legend titles, and increased text padding for better readability.

**Usage**

```
theme_apa(base_size = 12, base_family = "", box = FALSE)
```

**Arguments**

base\_size        Numeric. Base font size; other font sizes and margins are adjusted relative to this.  
 base\_family     Character. Base font family.  
 box             Logical. Indicates whether to draw a black panel border.

**Value**

Object of class `theme` and `gg`, see `ggplot2::theme()`.

**See Also**

`ggplot2::theme_bw()`, `ggplot2::theme()`

**Examples**

```
# Copied from ?ggtheme
mtcars2 <- within(mtcars, {
  vs <- factor(vs, labels = c("V-shaped", "Straight"))
  am <- factor(am, labels = c("Automatic", "Manual"))
  cyl <- factor(cyl)
  gear <- factor(gear)
})

library("ggplot2")
```

```

p1 <- ggplot(mtcars2) +
  geom_point(aes(x = wt, y = mpg, colour = gear)) +
  labs(
    title = "Fuel economy declines as weight increases",
    subtitle = "(1973-1974)",
    x = "Weight (1000 lbs)",
    y = "Fuel economy (mpg)",
    colour = "Gears"
  )

p1
p1 + theme_apa()

```

---

transmute\_df\_into\_label

*Transmute Degrees-of-Freedom Columns into Variable Labels*


---

### Description

Takes the output from `apa_print()` methods and modifies the results table by transmuting information about degrees of freedom into the variable labels of test-statistic columns.

### Usage

```
transmute_df_into_label(x, check_df = TRUE, ...)
```

```
df_into_label(x, check_df = TRUE, ...)
```

```
## S3 method for class 'apa_results'
transmute_df_into_label(x, check_df = TRUE, ...)
```

```
## S3 method for class 'apa_results_table'
transmute_df_into_label(x, check_df = TRUE, ...)
```

### Arguments

<code>x</code>	Either the complete output object created by <code>apa_print()</code> methods, or the table component of such objects.
<code>check_df</code>	Logical. If TRUE (the default), checks if degrees-of-freedom-columns contain non-integer values.
<code>...</code>	further arguments passed from an to other methods

### Value

An object of the same class as `x`, where a redundant column with degrees of freedom has been incorporated into the column label of the column statistic.

**Examples**

```
apa_out <- apa_print(aov(yield ~ N * P, npk))

# Standard output with separate columns for degrees of freedom:
apa_out$table

# Modified output where degrees of freedom are incorporated into the variable
# label of column 'statistic':
transmute_df_into_label(apa_out)$table
```

wsci

*Within-Subjects Confidence Intervals***Description**

Calculate Cousineau-Morey within-subjects confidence intervals.

**Usage**

```
wsci(data, id, factors, dv, level = 0.95, method = "Morey")
within_subjects_conf_int(data, id, factors, dv, level = 0.95, method = "Morey")
```

**Arguments**

data	A data.frame that contains the data.
id	Character. Variable name that identifies subjects.
factors	Character. A vector of variable names that is used to stratify the data.
dv	Character. The name of the dependent variable.
level	Numeric. Defines the width of the interval. Defaults to 0.95 for 95% confidence intervals.
method	Character. The method that is used to calculate CIs. Currently, "Morey" and "Cousineau" are supported. Defaults to "Morey".

**Value**

A data.frame with additional class `papaja_wsci`. The `summary()` method for this class returns a data.frame with means along lower and upper limit for each cell of the design.

**References**

Morey, R. D. (2008). Confidence Intervals from Normalized Data: A correction to Cousineau (2005). *Tutorials in Quantitative Methods for Psychology*, 4(2), 61–64.

Cousineau, D. (2005). Confidence intervals in within-subjects designs: A simpler solution to Loftus and Masson's method. *Tutorials in Quantitative Methods for Psychology*, 1(1), 42–45.

**Examples**

```
wsci(
  data = npk
  , id = "block"
  , dv = "yield"
  , factors = c("N", "P")
)
```

---

```
$.apa_results_table  Extract Parts of an APA Results Table
```

---

**Description**

*These methods are only defined for backward compatibility with older versions of **papaja**. In the past, the column names of `apa_results_tables` were less standardized than they are today. In order to maintain backwards compatibility, it is still possible to extract columns with the old column names, because we here provide *aliased* indexing. Note that aliased indexing will be defunct in a future release of **papaja**.*

**Usage**

```
## S3 method for class 'apa_results_table'
x$name

## S3 method for class 'apa_results_table'
x[[i, exact = TRUE]]

## S3 method for class 'apa_results_table'
x[i, j, ..., drop = TRUE]
```

**Arguments**

<code>x</code>	Object from which to extract element(s).
<code>name</code>	a literal character string or a <a href="#">name</a> (possibly <a href="#">backtick</a> quoted). For extraction, this is normally (see under ‘Environments’) partially matched to the <a href="#">names</a> of the object.
<code>i, j, ...</code>	Indices specifying elements to extract. See <a href="#">base::Extract()</a> for details.
<code>exact</code>	controls possible partial matching of <code>[[</code> when extracting by a character vector (for most objects, but see under ‘Environments’). The default is no partial matching. Value <code>NA</code> allows partial matching but issues a warning when it occurs. Value <code>FALSE</code> allows partial matching without any warning.
<code>drop</code>	relevant for matrices and arrays. If <code>TRUE</code> the result is coerced to the lowest possible dimension (see the examples). This only works for extracting elements, not for the replacement. See <a href="#">drop</a> for further details.

**Value**

A (vector of) character values as extracted from an object of class `apa_results_table`.

# Index

- \* **apa\_print**
  - apa\_print, 26
  - apa\_print.aov, 27
  - apa\_print.BFBayesFactor, 31
  - apa\_print.emmGrid, 35
  - apa\_print.glht, 37
  - apa\_print.htest, 38
  - apa\_print.list, 41
  - apa\_print.lm, 43
  - apa\_print.lme, 46
  - apa\_print.merMod, 47
- \* **plots for factorial designs**
  - apa\_barplot, 5
  - apa\_beeplot, 8
  - apa\_factorial\_plot, 13
  - apa\_lineplot, 18
  - apa\_violinplot, 53
- [.apa\_results\_table
  - (\$.apa\_results\_table), 77
- [[.apa\_results\_table
  - (\$.apa\_results\_table), 77
- \$.apa\_results\_table, 77
- add\_equals, 3
- add\_glue\_to\_apa\_results
  - (glue\_apa\_results), 62
- afex::mixed(), 48
- aov(), 31
- apa6\_doc(apa6\_pdf), 3
- apa6\_docx(apa6\_pdf), 3
- apa6\_docx(), 26
- apa6\_pdf, 3
- apa6\_pdf(), 67
- apa6\_word(apa6\_pdf), 3
- apa\_barplot, 5, 11, 16, 21, 56
- apa\_barplot(), 7, 10, 13, 15, 20, 56
- apa\_beeplot, 8, 8, 16, 21, 56
- apa\_beeplot(), 7, 10, 13, 15, 20, 56
- apa\_confint(apa\_interval), 16
- apa\_df, 12
- apa\_df(), 25
- apa\_factorial\_plot, 6, 8, 9, 11, 13, 19, 21, 54, 56
- apa\_factorial\_plot(), 7, 10, 15, 20, 56, 65
- apa\_hdint(apa\_interval), 16
- apa\_interval, 16
- apa\_lineplot, 8, 11, 16, 18, 56
- apa\_lineplot(), 7, 10, 13, 15, 20, 56
- apa\_num, 17, 22, 25, 30, 36, 37, 39, 44, 46, 48, 49, 52
- apa\_num(), 12, 18, 25, 53, 57
- apa\_num.numeric, 33
- apa\_p, 25
- apa\_p(), 12, 25
- apa\_prepare\_doc, 26
- apa\_print, 26, 31, 34, 37, 38, 40, 43, 45, 47, 48
- apa\_print(), 57, 63, 65, 73, 75
- apa\_print.afex\_aov(apa\_print.aov), 27
- apa\_print.anova(apa\_print.aov), 27
- apa\_print.anova.lme(apa\_print.lme), 46
- apa\_print.Anova.mlm(apa\_print.aov), 27
- apa\_print.aov, 27, 27, 34, 37, 38, 40, 43, 45, 47, 48
- apa\_print.aovlist(apa\_print.aov), 27
- apa\_print.BFBayesFactor, 27, 31, 31, 37, 38, 40, 43, 45, 47, 48
- apa\_print.BFBayesFactorTop
  - (apa\_print.BFBayesFactor), 31
- apa\_print.emmGrid, 27, 31, 34, 35, 38, 40, 43, 45, 47, 48
- apa\_print.glht, 27, 31, 34, 37, 37, 40, 43, 45, 47, 48
- apa\_print.glm(apa\_print.lm), 43
- apa\_print.htest, 27, 31, 34, 37, 38, 38, 43, 45, 47, 48
- apa\_print.list, 27, 31, 34, 37, 38, 40, 41, 45, 47, 48
- apa\_print.list(), 27, 31

- apa\_print.lm, 27, 31, 34, 37, 38, 40, 43, 43, 47, 48
- apa\_print.lme, 27, 31, 34, 37, 38, 40, 43, 45, 46, 48
- apa\_print.lsmobj (apa\_print.emmGrid), 35
- apa\_print.manova (apa\_print.aov), 27
- apa\_print.merMod, 27, 31, 34, 37, 38, 40, 43, 45, 47, 47
- apa\_print.mixed (apa\_print.merMod), 47
- apa\_print.papaja\_wsci, 49
- apa\_print.summary.Anova.mlm (apa\_print.aov), 27
- apa\_print.summary.aov (apa\_print.aov), 27
- apa\_print.summary.aovlist (apa\_print.aov), 27
- apa\_print.summary.glht (apa\_print.glht), 37
- apa\_print.summary.glm (apa\_print.lm), 43
- apa\_print.summary.lm (apa\_print.lm), 43
- apa\_print.summary.manova (apa\_print.aov), 27
- apa\_print.summary.ref.grid (apa\_print.emmGrid), 35
- apa\_print.summary\_emm (apa\_print.emmGrid), 35
- apa\_table, 50
- apa\_table(), 65
- apa\_violinplot, 8, 11, 16, 21, 53
- apa\_violinplot(), 7, 10, 15, 20, 56
- apply(), 24
- arrows(), 7, 10, 11, 15, 16, 20, 21, 55, 56
- axis(), 6, 7, 10, 11, 14, 15, 20, 55, 56
  
- backtick, 77
- base::Extract(), 77
- base::formatC, 23
- beautify\_terms, 57
- bookdown::pdf\_document2, 68
- bookdown::pdf\_document2(), 5, 68
- bookdown::word\_document2(), 5
  
- car::Anova(), 31
- ci (conf\_int), 59
- citation, 70
- cite\_r, 58, 70
- cite\_r(), 60, 61, 70
- class, 26
- conf.int (conf\_int), 59
  
- conf\_int, 59
- conf\_int(), 6, 7, 9, 10, 14, 15, 19, 20, 55
- connection, 72
- cor.test(), 38
- create\_bib, 69
- create\_bib (r\_refs), 69
  
- density(), 7, 11, 15, 21, 56
- df\_into\_label (transmute\_df\_into\_label), 75
- drop, 77
  
- fetch\_web\_refs, 59
- fetch\_zotero\_refs, 60
- file, 72
- format, 24
- formatC, 25
  
- generate\_author\_yaml, 61
- ggplot2::theme(), 74
- ggplot2::theme\_bw(), 74
- glue\_apa\_results, 62
- graphics::plot.window, 14
  
- hd\_int, 64
  
- in\_paren, 65
- includes, 4
  
- kable, 52
- knitr::asis\_output(), 53, 67
- knitr::kable, 51
- knitr::kable(), 53
- knitr::write\_bib(), 58, 60, 70
  
- legend(), 7, 10, 11, 15, 16, 20, 21, 55, 56
- lines(), 7, 11, 15, 16, 20, 21, 55, 56
- lm, 41
- lme4::confint.merMod(), 48
- lme4::glmer(), 48
- lme4::lmer(), 48
- lmerTest::lmer(), 48
  
- MBESS::ci.pvaf(), 45
- MBESS::ci.R2(), 42, 44
  
- name, 77
- names, 77
- nlme::intervals.lme(), 46
- nlme::lme(), 46

`nlme::nlme()`, 46  
`opts_current`, 52  
  
`papaja`, 65  
`papaja-package (papaja)`, 65  
`pdf_document2`, 4  
`plot.new()`, 7, 10, 15, 20, 56  
`plot.window()`, 7, 11, 15, 20, 56  
`points()`, 7, 10, 11, 14–16, 20, 21, 55, 56  
`polygon()`, 7, 11, 15, 21, 56  
`print_confint (apa_interval)`, 16  
`print_df (apa_df)`, 12  
`print_hdint (apa_interval)`, 16  
`print_interval (apa_interval)`, 16  
`print_num (apa_num)`, 22  
`print_p (apa_p)`, 25  
`printnum (apa_num)`, 22  
`printp (apa_p)`, 25  
  
`quote_from_tex`, 67  
`quote_from_tex()`, 68  
  
`r_refs`, 69  
`r_refs()`, 58, 60, 61  
`rect()`, 7, 11, 14, 15, 21, 56  
`remove_comments`, 67  
`revision_letter_pdf`, 68  
`rmarkdown::render()`, 5, 68  
`rmarkdown_format`, 4  
  
`sanitize_terms`, 70  
`sanitize_terms()`, 44  
`scale()`, 57, 70  
`se`, 71  
`se()`, 7, 10, 15, 20, 55  
`signif`, 23  
`simple_codebook`, 71  
`sort_terms`, 73  
`stats::anova()`, 43  
`stats::confint()`, 44, 45  
`summary.manova`, 30  
`summary.papaja_wsci`, 73  
  
`t.test()`, 38  
`theme_apa`, 74  
`title()`, 7, 10, 11, 14, 15, 20, 55, 56  
`transmute_df_into_label`, 75  
  
`utils::citation()`, 70  
`utils::toLatex()`, 70  
`utils::write.csv()`, 72  
`utils::write.table`, 72  
  
`wilcox.test()`, 38  
`within_subjects_conf_int (wsci)`, 76  
`within_subjects_conf_int()`, 7, 10, 15, 20, 55  
`word_document2`, 4  
`write_bib`, 69  
`wsci`, 49, 76  
`wsci()`, 7, 10, 15, 20, 55, 73, 74