Package 'prmisc'

March 7, 2025

Type Package

Title Miscellaneous Printing of Numeric and Statistical Output in R Markdown and Quarto Documents

Version 0.0.3

Author Martin Papenberg [aut, cre] (<https://orcid.org/0000-0002-9900-4268>), Juliane V. Nagel [aut] (<https://orcid.org/0000-0002-5310-8088>)

Description Miscellaneous printing of numeric or statistical results in R Markdown or Quarto documents according to guidelines of the ``Publication Manual'' of the American Psychological Association (2020, ISBN: 978-1-4338-3215-4). These guidelines are usually referred to as APA style (<https://apastyle.apa.org/>) and include specific rules on the formatting of numbers and statistical test results. APA style has to be implemented when submitting scientific reports in a wide range of research fields, especially in the social sciences. The default output of numbers in the R console or R Markdown and Quarto documents does not meet the APA style requirements, and reformatting results manually can be cumbersome and error-prone. This package covers the automatic conversion of R objects to textual representations that meet the APA style requirements, which can be included in R Markdown or Quarto documents. It covers some basic statistical tests (t-test, ANOVA, correlation, chi-squared test, Wilcoxon test) as well as some basic number printing manipulations (formatting p-values, removing leading zeros for numbers that cannot be greater than one, and others). Other packages exist for formatting numbers and tests according to the APA style guidelines, such as 'papaja' (<https://cran.r-project.org/package=papaja>) and 'apa' (<https://cran.r-project.org/package=apa>), but they do not offer all convenience functionality included in 'prmisc'. The vignette has an overview of most of the functions included in the package.

License MIT + file LICENSE

URL https://github.com/m-Py/prmisc

BugReports https://github.com/m-Py/prmisc/issues

Depends R (>= 3.0.0)

Suggests afex, effectsize, knitr, rmarkdown, spgs

VignetteBuilder knitr, rmarkdown

Encoding UTF-8

LazyLoad yes

RoxygenNote 7.3.2

NeedsCompilation no

Maintainer Martin Papenberg <martin.papenberg@hhu.de>

Repository CRAN

Date/Publication 2025-03-07 16:20:02 UTC

Contents

`decimals_only` *Printing a specified number of decimals and ignore leading zeros*

Description

Printing a specified number of decimals and ignore leading zeros

Usage

```
decimals_only(x, decimals = 2, decimals1 = FALSE)
```

Arguments

`x`	the values to be printed
`decimals`	how many decimals are to be printed. Defaults to 2.
`decimals1`	Boolean - should a value of exactly 1 be converted to 1.00. Defaults to FALSE.

Value

Character vector of length `length(x)`. The number(s) in the required format.

Author(s)

Martin Papenberg <martin.papenberg@hhu.de>

Examples

```
decimals_only(c(0.23456, 0.873, 0.3456), decimals = 3)
```

`force_decimals` *Force printing a specified number of decimals for a number*

Description

Force printing a specified number of decimals for a number

Usage

```
force_decimals(x, decimals = 2, round_zero = TRUE)
```

Arguments

`x`	the numeric values to be printed
`decimals`	how many decimals are to be printed. Defaults to 2.
`round_zero`	whether small values should be rounded to zero or whether they should be displayed as e.g. < .01. See details. Defaults to `TRUE`.

Details

By default, `force_decimals()` will round numbers that are small enough down to zero, e.g., 0.0004 will be rounded to 0.00. If `round_zero = FALSE`, `force_decimals(0.0004)` will return `"< 0.01"` instead. See examples.

Value

Character vector of length `length(x)`. The number(s) in the required format.

Author(s)

Martin Papenberg <martin.papenberg@hhu.de>

Examples

```
force_decimals(c(1.23456, 0.873, 2.3456))
force_decimals(c(1.23456, 0.873, 2.3456), 3)

force_decimals(c(0.004, 0.001, 0.0005, 0.02))
force_decimals(c(0.004, 0.001, 0.0005, 0.02), round_zero = FALSE)
force_decimals(c(0.004, 0.001, 0.0005, 0.02), 3, round_zero = FALSE)
```

force_or_cut *Print a number having a specified number of digits or as integer*

Description

Print a number having a specified number of digits or as integer

Usage

```
force_or_cut(x, decimals = 2)
```

Arguments

x	A vector of numbers
decimals	The number of digits that should be printed if x is a decimal number. Defaults to 2.

Details

If x integer, only the integer is printed, if x is a decimal number, the decimals are printed

Value

Character vector of length `length(x)`. The number(s) in the required format.

Author(s)

Martin Papenberg <martin.papenberg@hhu.de>

Examples

```
force_or_cut(c(1:3, 1.23456, 0.873, 2.3456))
```

format_p *Format a p-value according to APA standards*

Description

Format a p-value according to APA standards

Usage

```
format_p(pvalues, decimals = 3, numbers_only = FALSE, latex = TRUE)
```

Arguments

pvalues	The p-values
decimals	The number of decimals to be printed
numbers_only	Logical, indicates whether the p-values should be printed without the accompanying p. Defaults to FALSE.
latex	Logical, indicates whether the p-values should be printed with or without $ around it. Defaults to TRUE.

Value

Character vector of length length(pvalues). A string representation of the p value(s) to be used in Rmarkdown documents.

Examples

```
# Format a p-value, default is 3 decimals
format_p(0.03123)
format_p(0.000001231)
format_p(0.000001231, decimals = 2)
format_p(0.3123, decimals = 2)
format_p(0.3123, latex = FALSE)
# Format several p-values with one function call
format_p(c(0.3123, 0.001, 0.00001, 0.19))
format_p(c(.999, .9999, 1))
format_p(c(0.3123, 0.001, 0.00001, 0.19, .99999), numbers_only = TRUE)
```

---

print_anova *Print the results of an* afex *ANOVA*

Description

Print the results of an afex ANOVA

Usage

```
print_anova(afex_object, italic_eta = TRUE, decimals = 2, decimals_p = 3)
```

Arguments

afex_object	An object returned by one of afex's ANOVA functions. See details.
italic_eta	Should the effect size symbol eta be printed in italic font. Defaults to TRUE. See details.
decimals	How many decimals should be printed for F values and eta-squared. Defaults to 2.
decimals_p	How many decimals should be printed for p-values. Defaults to 3.

Details

To use this function, you have to install the package afex to compute an ANOVA object, see aov_car. Pass this object as the first argument.

According to APA style, the Greek eta symbol - indicating the effect size in the ANOVA - should be printed in non-italic font. However, the standard \eta symbol is written in italic. To print a non-italic eta, use the argument `italic_eta = FALSE`. However, this option requires that you load the Latex package upgreek in the YAML header of your R markdown document. To this end, use the following option in your YAML header:

```
header-includes:
  -\usepackage{upgreek}
```

This option only works for Latex/PDF output.

Value

A list whose elements are strings describing the effects of the ANOVA (main effects and interactions); to be included in an R markdown document.

Author(s)

Martin Papenberg <martin.papenberg@hhu.de>

References

Singmann, H., Bolker, B., Westfall, J., & Aust, F. (2019). afex: Analysis of Factorial Experiments. https://CRAN.R-project.org/package=afex

Examples

```
library("afex")
# see ?aov_ez
data(md_12.1)
aov_results <- aov_ez("id", "rt", md_12.1, within = c("angle", "noise"))
print_anova(aov_results)

# Print nonitalic eta, which is required according to APA guidelines
print_anova(aov_results, italic_eta = FALSE)

# Example using other (or no) effect size index
pes <- aov_ez("id", "rt", md_12.1, within = c("angle", "noise"),
              anova_table = list(es = "pes"))
print_anova(pes)
print_anova(pes, italic_eta = FALSE)
noes <- aov_ez("id", "rt", md_12.1, within = c("angle", "noise"),
               anova_table = list(es = "none"))
print_anova(noes)

## Access individual elements of the ANOVA print list:
aovpr <- print_anova(aov_results, italic_eta = FALSE)
```

```
# By index:
aovpr[[1]]
# By name (main effect):
aovpr$angle
aovpr[["angle"]]
# By name (interaction effect, here, the $-notation does not work
# due to non-standard `:` in name):
aovpr[["angle:noise"]]
```

`print_chi2` *Print the results of a chi-square test*

Description

Print the results of a chi-square test

Usage

```
print_chi2(
  x,
  es = TRUE,
  correct = FALSE,
  decimals = 2,
  decimals_p = 3,
  italic_greek = TRUE
)
```

Arguments

`x`	A contingency table (passed as `table` or `matrix`) or an object of type "htest" returned by `chisq.test`. Can also handle objects returned by `chisq.unif.test` from the `spgs` package.
`es`	Boolean. Should the phi coefficient be printed as a measure of effect size. See details.
`correct`	Boolean. Apply a continuity correction? See `chisq.test`. Only has an effect if the chi-square-test is computed by this function, i.e., if `x` is a contingency table. The default value is `FALSE`.
`decimals`	How many decimals should be printed
`decimals_p`	How many decimals should be printed for the p-value (defaults to 3)
`italic_greek`	Should the greek letters (for chi and possily phi) be printed in italic font. Defaults to `TRUE`. According to APA style this should be `FALSE`.

Details

The argument `es` only has an effect if `x` is passed as a 2x2 contingency table. In this case, the phi coefficient is computed as a measure of effect size (see Cohen, 1988, page 223).

According to APA style, the Greek chi symbol (and the phi coefficient) should be printed in non-italic font. However, the standard symbols \chi and \phi are written in italic. To print non-italic symbols, use the argument `italic_greek = FALSE`. However, this option requires that you load the Latex package `upgreek` in the YAML header of your R markdown document. To this end, use the following option in your YAML header:

```
header-includes:
  -\usepackage{upgreek}
```

This option only works for Latex/PDF output.

Value

A string describing the results of the chi-square test to be printed in Rmarkdown documents.

Author(s)

Martin Papenberg <martin.papenberg@hhu.de>

References

Cohen, J. (1988). Statistical power analysis for the behavioral sciences (2nd ed.). Hillsale, NJ: Lawrence Erlbaum.

Examples

```
# Pass a matrix
x <- matrix(c(12, 5, 7, 7), ncol = 2)
print_chi2(x) # does not use continuity correction by default
print_chi2(x, correct = TRUE) # uses continuity correction

# Pass a table
tab <- table(rbinom(150, 1, 0.5), rbinom(150, 1, 0.1))
print_chi2(tab, correct = FALSE)

# Pass a chi-squared test object
print_chi2(chisq.test(tab, correct = FALSE))

# Use non italic chi symbol
print_chi2(tab, italic_greek = FALSE)
```

print_cortest *Printing the results of a significance test for a correlation coefficient*

Description

Printing the results of a significance test for a correlation coefficient

Usage

```
print_cortest(cor_object, decimals = 2, decimals_p = 3)
```

Arguments

cor_object	An object of class "htest" returned by cor.test
decimals	How many decimals should be printed for the test statistic (defaults to 2).
decimals_p	How many decimals should be printed for the p value (defaults to 3).

Value

A string describing the significance test; to be included in an R markdown document.

Author(s)

Martin Papenberg <martin.papenberg@hhu.de>

Examples

```
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_results <- cor.test(x, y)
print_cortest(cor_results)
```

print_mean_sd *Print mean and standard deviation*

Description

Print mean and standard deviation

Usage

```
print_mean_sd(
  x,
  decimals_M = 2,
  decimals_SD = 2,
  parentheses = TRUE,
  short = FALSE,
  na.rm = FALSE
)
```

Arguments

x	a vector of the sample the statistics should be printed for.
decimals_M	how many decimals should be printed for the mean (defaults to 2).
decimals_SD	how many decimals should be printed for the standard deviation (defaults to 2).
parentheses	logical indicating if the statistics should be wrapped in parentheses or not (defaults to TRUE). Will be ignored if short is TRUE.
short	logical indicating if a short version without the letters should be printed. Argument parantheses is ignored in this case. Defaults to FALSE. See details.
na.rm	logical indicating whether missing values should be ignored or not. Defaults to FALSE.

Details

The following formatting options are available:

- When parentheses is TRUE: $(M = XX, SD = XX)$
- When parentheses is FALSE: $M = XX, SD = XX$
- When short is TRUE: $XX(XX)$

Value

A string with information on mean and standard deviation in x.

Author(s)

Juliane Nagel <juliane.nagel@zi-mannheim.de>

Examples

```
print_mean_sd(rnorm(100, 0, 1))

print_mean_sd(1:20, decimals_M = 0, decimals_SD = 3)

print_mean_sd(c(2, 10, 12.5, 3), parentheses = FALSE)
```

print_ttest *Print the results of a t-test*

Description

Print the results of a t-test

Usage

```
print_ttest(
  t_object,
  d_object = NULL,
  decimals = 2,
  decimals_p = 3,
  confidence = FALSE
)
```

Arguments

t_object	An object of class "htest" returned by t.test.
d_object	An effect size table returned by cohens_d from package effectsize. Optional argument.
decimals	How many decimals should be printed for the t-value (defaults to 2).
decimals_p	How many decimals should be printed for the p-value (defaults to 3).
confidence	Logical. Whether a confidence interval for the effectsize should be printed or not. Can only be TRUE if d_object is provided.

Details

To use this function, you need to install the R package effectsize to compute Cohen's d; pass this object as the second argument.

Value

A string describing the t-test; to be included in an R markdown document.

Author(s)

Martin Papenberg <martin.papenberg@hhu.de>

References

Cohen, J. (1988). Statistical power analysis for the behavioral sciences (2nd ed.). Hillsale, NJ: Lawrence Erlbaum.

Examples

```
ttest <- t.test(1:10, y = c(7:20), var.equal = TRUE)
library("effectsize") # for Cohen's d
cohend <- cohens_d(1:10, c(7:20))
print_ttest(ttest, cohend) # include this call in Rmd inline code

# An example for paired data:
data(sleep) # ?sleep
tt <- t.test(sleep$extra[sleep$group == 1],
             sleep$extra[sleep$group == 2], paired = TRUE)
cd <- cohens_d(sleep$extra[sleep$group == 1],
               sleep$extra[sleep$group == 2], paired = TRUE)
print_ttest(tt, cd)
# Print the confidence interval
print_ttest(tt, cd, confidence = TRUE)
# The information about the CI is taken from the effectsize object:
cd <- cohens_d(sleep$extra[sleep$group == 1],
               sleep$extra[sleep$group == 2], paired = TRUE, ci = .8)
print_ttest(tt, cd, confidence = TRUE)
# effect size object can be left out:
print_ttest(tt)
```

---

print_wilcoxon_rs *Print the results of a Wilcoxon rank sum test (Mann-Whitney-U test)*

Description

Print the results of a Wilcoxon rank sum test (Mann-Whitney-U test)

Usage

```
print_wilcoxon_rs(
  wc_object,
  decimals_p = 3,
  consistent = NULL,
  group1 = NULL,
  group2 = NULL,
  groupvar = NULL,
  effsize = NULL,
  neg = FALSE,
  N = NULL,
  decimals_eff = NULL
)
```

Arguments

wc_object	an object returned by wilcox.test

decimals_p	how many decimals should be printed for the p-value (defaults to 3)
consistent	an optional parameter determining for which group the test statistic U should be reported. Can be set to 'min' or 'max'. See details.
group1	a vector containing the cases of the first group
group2	a vector containing the cases of the second group
groupvar	a vector containing a grouping variable
effsize	a character indicating which effect size should be reported, if any. Possible values are: 'r', 'rsqu' and 'd'. By default, no effect size will be reported. See details.
neg	a logical indicating whether the effect size should be reported with a negative sign. Defaults to FALSE. See details.
N	an integer passing information about the total N of the sample. Needed when effect sizes should be calculated, but N cannot be inferred because neither of group1, group2 or groupvar have been provided. Should groups or a grouping variable have been provided, N must not be used.
decimals_eff	an integer specifying how many digits the effect size should be printed with, if an effect size is desired. Defaults to 2. r and r squared will be printed without a leading zero, d will be printed with a leading zero because it is possible for d to take values larger than one.

Details

In order to calculate a Wilcoxon rank sum test, the argument paired in wilcox.test needs to be FALSE. Otherwise, a Wilcoxon signed rank test will be computed instead and the statistics printed by print_wilcoxon_rs will be misleading.

Note that the test statistic W calculated in wilcox.test that is printed as test statistic U in print_wilcoxon_rs will correspond to the U of the first of the two groups compared in wilcox.test. In the default method of wilcox.test, this is the vector assigned to x. In the formula method, this is the first group as identified by the grouping variable. Some software, like SPSS, consistently reports the smaller or larger U. If you wish to mimic this, you can specify the desired behaviour by providing the consistent argument. Setting consistent to 'min' will print the smaller of the two U, setting it to 'max' will print the larger U. In order to do so, you need to provide the n for both groups.

You can either do that by passing the data of both groups to the arguments group1 and group2, respectively. From those, print_wilcoxon_rs will calculate the group sizes. Any vector with the length of the corresponding group size will produce the desired result. This is the recommended option if your wc_object has been created using the default version of wilcox.test, i.e. if groups were passed as x and y. The order in which you pass group1 and group2 to print_wilcoxon_rs does not have to correspond to the order in which x and y were passed to wilcox.test.

Alternatively, you can pass a grouping variable to the argument groupvar. From this, print_wilcoxon_rs will calculate group sizes. This is the recommended option if your wc_object has been created using the formula version of wilcox.test, i.e. if a grouping variable was passed after the ~. To prevent mistakes, you can either use group1 and group2 **or** groupvar.

By default, when consistent is not provided, print_wilcoxon_rs will print U using W as taken from wilcox.test.

There are three options available for calculating an effect size via the argument effsize:

- the point biserial correlation r (`effsize = 'r'`), which is calculated by dividing Z-score by the square root of N. According to Cohen (1988) a small, medium and large effect correspond to r = .1, .3 and .5, respectively. Currently, `print_wilcoxon_rs` infers the Z-score from the p-value.
- r squared (identical to eta squared; `effsize = 'rsqu'`), which is the ratio of variability associated with an effect compared to the ratio of overall variance
- d (`effsize = 'd'`), which is calculated from r as follows:

```
2*r/(sqrt(1-r^2))
```

According to Cohen (1988), a small, medium and large effect correspond to r = .2, .5 and .8, respectively.

Information on the direction of the effect (as indicated in the sign of the Z-score) or in which order the groups have been compared is not included in the output of `wilcox.test`. Hence, only absolute effect sizes can be calculated. It is advised to report absolute effect sizes unless there is a meaningful for the two groups tested (e.g. Fritz et al, 2012). In some cases, it might be desired to specify the direction of an effect by including the sign of the effect. To that end, it is possible to print the negative effect size with `neg = TRUE`, but caution is advised: You should always check if the sign of the effect size you report is the correct one, especially in the case of one-sided testing.

Value

A string describing the results of the Wilcoxon test; to be included in an R markdown document.

Author(s)

Juliane Nagel <juliane.nagel@zi-mannheim.de>

References

Cohen, J. (1988). Statistical power analysis for the behavioral sciences (2nd ed.). Hillsdale, NJ: Erlbaum.

Fritz, C. O., Morris, P. E., & Richler, J. J. (2012). Effect size estimates: Current use, calculations, and interpretation. Journal of Experimental Psychology: General, 141(1), 2-18. http://dx.doi.org/10.1037/a0024338

Examples

```
data(iris)
dat <- subset(iris, Species %in% c("setosa", "versicolor"))
wc_iris <- wilcox.test(dat$Sepal.Length ~ dat$Species, correct = FALSE)
# include this call in Rmd inline code
print_wilcoxon_rs(wc_iris)

wc_iris2 <- wilcox.test(dat$Sepal.Width ~ dat$Species, correct = FALSE)

print_wilcoxon_rs(wc_iris2, consistent = "min",
                  group1 = dat$Sepal.Width[dat$Species == "setosa"],
                  group2 = dat$Sepal.Width[dat$Species == "versicolor"])

print_wilcoxon_rs(wc_iris2, consistent = "max",
```

```
                    groupvar = dat$Species)

print_wilcoxon_rs(wc_iris2, consistent = "max",
                    groupvar = dat$Species, effsize = "rsqu")
```

Index