Package 'papeR'

May 9, 2026

Title A Toolbox for Writing Pretty Papers and Reports

Version 1.0-6

Date 2025-04-01

Maintainer Benjamin Hofner <benjamin.hofner@pei.de>

Description A toolbox for writing 'knitr', 'Sweave' or other 'LaTeX'- or 'markdown'-based reports and to prettify the output of various estimated models.

Depends car, xtable

Enhances lme4, survival

Imports utils, gmodels, graphics, stats

Suggests nlme, knitr, rmarkdown, testthat (>= 0.10.0), foreign

License GPL-2

URL https://github.com/hofnerb/papeR

Copyright See inst/COPYRIGHTS.

VignetteBuilder knitr

Collate 'helpers.R' 'labels.R' 'summarize.R' 'plot.R' 'prettify.R' 'toLatex.R'

NeedsCompilation no

Author Benjamin Hofner [aut, cre] (ORCID: <https://orcid.org/0000-0003-2810-3186>),
Romain Francois [ctb],
Kurt Hornik [ctb],
Martin Maechler [ctb],
David Dahl [ctb] (see inst/CONTRIBUTIONS for details)

Repository CRAN

Date/Publication 2025-04-01 15:20:02 UTC

Contents

`papeR-package` *A Toolbox for Writing Pretty Papers and Reports*

Description

A toolbox for writing knitr, Sweave or other LaTeX- or markdown-based reports and to prettify the output of various estimated models.

Details

Package:	papeR
Type:	Package
Version:	1.0-6
Date:	2025-04-01
License:	GPL-2

Version 1.0-0 is based on a completely refactored code base. Some functions from previsous versions are deprecated. New functions to create summary tables exist (see `summarize`. The package now also provides a vignette and was extensively tested using `testthat`.

For news and changes see `news(package = "papeR")`.

Author(s)

Benjamin Hofner

Maintainer: Benjamin Hofner <Benjamin.Hofner@fau.de>

`Anova.lme` *Anova Function for lme Models*

Description

This is a wrapper to `anova.lme` from package **nlme** and is coded similar to `Anova` from **car** as it produces marginal tests by default.

Usage

```
## S3 method for class 'lme'
Anova(mod, type = c("marginal", "sequential"), ...)
```

Arguments

`mod`	linear mixed model fitted with package **nlme**.
`type`	type of anova, either marginal (default) or sequential.
`...`	further arguments to be passed to `anova.lme`

See Also

`Anova` (package **car**)

Examples

```
## Example requires package nlme to be installed and loaded
if (require("nlme")) {
    ## Load data set Orthodont
    data(Orthodont, package = "nlme")

    ## Fit a model for distance with random intercept for Subject
    mod <- lme(distance ~ age + Sex, data = Orthodont, random = ~ 1 | Subject)

    Anova(mod)
}
```

`confint.mer` *Confidence intervals for mixed models*

Description

Compute confidence intervals for mixed models from package **lme4** (prior to version 1.0). This function is only needed for backward compatibility.

Usage

```
## S3 method for class 'mer'
confint(object, parm, level = 0.95,
            simulate = c("ifneeded", TRUE, FALSE),
            B = 1000,...)
```

Arguments

object	Model of class mer.
parm	Parameters to be included in the confidence interval. See confint.default for details.
level	the confidence level.
simulate	If ""ifneeded"" is specified (default), simulated confidence intervals are returned if (and only if) no z-value exists in the corresponding summary and asymptotic confidence intervals will be returned otherwise. If TRUE (or ""TRUE"") confidence intervals will be estimated using ci from package **gmodels**, which uses mcmcsamp internally. If FALSE (or "'FALSE""), asymptotic confidence intervals will be returned and an error is given if not possible.
B	number of samples to take in mcmcsamp. Per default 1000 samples are used.
...	Additional arguments. Currently not used.

Value

Matrix with confidence intervals.

Author(s)

Benjamin Hofner, partially based on code from package stats. See source code for documentation.

---

get_option *Extract printing options from* table.cont *and* table.fac *objects*

Description

Helper function to extract printing options from table.cont and table.fac objects as produced by latex.table.cont and latex.table.fac.

Usage

```
get_option(object, name)
```

Arguments

object	table.cont or table.fac object as produced by latex.table.cont and latex.table.fac
name	name of the option, e.g. "table" and "align". See latex.table.cont and latex.table.fac for available options.

Value

Option.

Author(s)

Benjamin Hofner

See Also

latex.table.cont and latex.table.fac

---

labels *Extract labels from and set labels for data frames*

Description

Labels can be stored as an attribute "variable.label" for each variable in a data set using the assignment function. With the extractor function one can assess these labels.

Usage

```
## S3 method for class 'data.frame'
labels(object, which = NULL, abbreviate = FALSE, ...)

## assign labels
labels(data, which = NULL) <- value

## check if data.frame is a special labeled data.frame ('ldf')
is.ldf(object)

## convert object to labeled data.frame ('ldf')
convert.labels(object)
as.ldf(object, ...)

## special plotting function for labeled data.frames ('ldf')
## S3 method for class 'ldf'
plot(x, variables = names(x),
     labels = TRUE, by = NULL, with = NULL,
     regression.line = TRUE, line.col = "red", ...)
```

Arguments

object	a data.frame.
data	a data.frame.

which	either a number indicating the label to extract or a character string with the *variable name* for which the label should be extracted. One can also use a vector of numerics or character strings to extract mutiple labels. If which is NULL (default), all labels are returned.
value	a vector containing the labels (in the order of the variables). If which is given, only the corresponding subset is labeled. Note that all other labels contain the variable name as label afterwards.
abbreviate	logical (default: FALSE). If TRUE variable labels are abbreviated such that they remain unique. See abbreviate for details. Further arguments to abbreviate can be specified (see below).
...	further options passed to function abbreviate if argument abbreviate = TRUE. In x[...], ... can be used to specify indices for extraction. See [for details. In plot, ... can be used to specify further graphial parameters.
x	a labeled data.frame with class 'ldf'.
variables	character vector or numeric vector defining (continuous) variables that should be included in the table. Per default, all numeric and factor variables of data are used.
labels	labels for the variables. If labels = TRUE (the default), labels(data, which = variables) is used as labels. If labels = NULL variables is used as label. labels can also be specified as character vector.
by	a character or numeric value specifying a variable in the data set. This variable can be either a grouping factor or is used as numeric y-variable (see with for details). Per default no grouping is applied. See also 'Details' and 'Examples'.
with	a character or numeric value specifying a numeric variable with which to "correlate" all variables specified in variables. For numeric variables a scatterplot is plotted, for factor variables one gets a grouped boxplot. Per default no variable is given here. Instead of with one can also specify a numeric variable in by with the same results. See also 'Details' and 'Examples'.
regression.line	a logical argument specifying if a regression line should be added to scatter plots (which are plotted if both variables and by are numeric values).
line.col	the color of the regression line.

Details

All labels are stored as attributes of the columns of the data frame, i.e., each variable has (up to) one attribute which contains the variable lable.

One can set or extract labels from data.frame objects. If no labels are specified labels(data) returns the column names of the data frame.

Using abbreviate = TRUE, all labels are abbreviated to (at least) 4 characters such that they are unique. Other minimal lengths can specified by setting minlength (see examples below).

Univariate plots can be easily obtained for all numeric and factor variables in a data set data by using plot(data).

Bivariate plots can be obtained by specifying by. In case of a factor variable, grouped boxplots or spineplots are plotted depending on the class of the variable specified in variables. In case

of a numeric variable, grouped boxplots or scatter plots are plotted depending on the class of the variable specified in variables. Note that one cannot specify by and with at the same time (as they are internally identical). Note that missings are excluded plot wise (also for bivariate plots).

Value

labels(data) returns a named vector of variable labels, where the names match the variable names and the values represent the labels.

Note

If a data set is generated by read.spss in package **foreign**, labels are stored in a single attribute of the data set. Assigning new labels, e.g. via labels(data) <- labels(data) removes this attribute and stores all labels as attributes of the variables. Alternatively one can use data <- convert.labels(data).

Author(s)

Benjamin Hofner

See Also

read.spss in package **foreign**

Examples

```
############################################################
### Basic labels manipulations

data <- data.frame(a = 1:10, b = 10:1, c = rep(1:2, 5))
labels(data)  ## only the variable names
is.ldf(data) ## not yet

## now set labels
labels(data) <- c("my_a", "my_b", "my_c")
## one gets a named character vector of labels
labels(data)
## data is now a ldf:
is.ldf(data)

## Altervatively one could use as.ldf(data) or convert.labels(data);
## This would keep the default labels but set the class
## correctly.

## set labels for a and b only
## Note that which represents the variable names!
labels(data, which = c("a", "b")) <- c("x", "y")
labels(data)

## reset labels (to variable names):
labels(data) <- NULL
labels(data)
```

```
## set label for a only and use default for other labels:
labels(data, which = "a") <- "x"
labels(data)

## attach label for new variable:
data2 <- data
data2$z <- as.factor(rep(2:3, each = 5))
labels(data2)  ## no real label for z, only variable name
labels(data2, which = "z") <- "new_label"
labels(data2)


############################################################
### Abbreviate labels

## attach long labels to data
labels(data) <- c("This is a long label", "This is another long label",
                  "This also")
labels(data)
labels(data, abbreviate = TRUE, minlength = 10)


############################################################
### Data manipulations

## reorder dataset:
tmp <- data2[, c(1, 4, 3, 2)]
labels(tmp)
## labels are kept and order is updated

## subsetting to single variables:
labels(tmp[, 2])  ## not working as tmp[, 2] drops to vector
## note that the label still exists but cannot be extracted
## using labels.default()
str(tmp[, 2])

labels(tmp[, 2, drop = FALSE]) ## prevent dropping

## one can also cbind labeled data.frame objects:
labels(cbind(data, tmp[, 2]))
## or better:
labels(cbind(data, tmp[, 2, drop = FALSE]))
## or rbind labeled.data.set objects:
labels(rbind(data, tmp[, -2]))


############################################################
### Plotting data sets

## plot the data auto"magically"; numerics as boxplot, factors as barplots
par(mfrow = c(2,2))
plot(data2)
```

```
## a single plot
plot(data2, variables = "a")
## grouped plot
plot(data2, variables = "a", by = "z")
## make "c" a factor and plot "c" vs. "z"
data2$c <- as.factor(data2$c)
plot(data2, variables = "c", by = "z")
## the same
plot(data2, variables = 3, by = 4)

## plot everithing against "b"
## (grouped boxplots, stacked barplots or scatterplots)
plot(data2, with = "b")
```

`latex.table.cont` *Produce (LaTeX) Summaries for Continuous Variables*

Description

The function produces LaTeX tables with summary statistics for continous variables. It makes use of the booktabs package in LaTeX to obtain tables with a nice layout.

Usage

```
latex.table.cont(..., caption = NULL, label = NULL,
    table = c("tabular", "longtable"), align = NULL,
    floating = FALSE, center = TRUE)
```

Arguments

`...`	arguments for `summarize`. See there for details.
`caption`	(optional) character string. Caption of LaTeX table. Note that captions are suported for all tables (see also details below).
`label`	(optional) character string. Label of LaTeX table specified as `\label{"label"}`.
`table`	character string. LaTeX table format, currently either `"tabular"` (default) or `"longtable"`.
`align`	character string. LaTeX alignment of table rows, per default `"llr...r"`, where `"r"` is repeated `ncol - 1` times.
`floating`	logical (default: `FALSE`). Determines whether the table is a floating object (i.e. use a `table` environment or not). Note that a `longtable` cannot be a floating object but captions can be used.
`center`	logical (default: `TRUE`). Determines if table should be centered.

Details

This function is deprecated and only available for backward comaptibility. Use summarize for more flexibility.

The output requires \usepackage{booktabs} in the LaTeX file.

Captions can be added to both, longtables and tabulars. In the latter case, captions are also suported if the table is no floating object. In this case, the LaTeX package capt-of is required.

Value

The output is printed with LaTeX style syntax highlighting to be used e.g. in Sweave chunks with results=tex.

Author(s)

Benjamin Hofner

See Also

latex.table.fac and get_option

Examples

```
## Example requires package nlme to be installed and loaded
if (require("nlme")) {
    ## Use dataset Orthodont
    data(Orthodont, package = "nlme")

    ## Get summary for continuous variables
    latex.table.cont(Orthodont)

    ## Change statistics to display
    latex.table.cont(Orthodont, quantiles = FALSE)
    latex.table.cont(Orthodont, count = FALSE, quantiles = FALSE)
    latex.table.cont(Orthodont, mean_sd = FALSE)

    ## Show column 'Missing' even if no missings are present
    latex.table.cont(Orthodont, show.NAs = TRUE)

    ## Change variables to display
    latex.table.cont(Orthodont, variables = "age")

    ## What happens in the display if we introduce some missing values:
    set.seed(1907)
    Orthodont$age[sample(nrow(Orthodont), 20)] <- NA
    latex.table.cont(Orthodont)
}
```

`latex.table.fac` *Produce (LaTeX) Summaries for Factor Variables*

Description

The function produces LaTeX tables with summary statistics for factor variables. It makes use of the booktabs package in LaTeX to obtain tables with a nice layout.

Usage

```
latex.table.fac(..., caption = NULL, label = NULL,
    table = c("tabular", "longtable"), align = NULL,
    floating = FALSE, center = TRUE)
```

Arguments

`...`	arguments for `summarize`. See there for details.
`caption`	(optional) character string. Caption of LaTeX table. Note that captions are suported for all tables (see also details below).
`label`	(optional) character string. Label of LaTeX table specified as `\label{"label"}`.
`table`	character string. LaTeX table format, currently either `"tabular"` (default) or `"longtable"`.
`align`	character string. LaTeX alignment of table rows, per default `"lllr...r"`, where `"r"` is repeated `ncol - 2` times.
`floating`	logical (default: `FALSE`). Determines whether the table is a floating object (i.e. use a `table` environment or not). Note that a `longtable` cannot be a floating object but captions can be used.
`center`	logical (default: `TRUE`). Determines if table should be centered.

Details

This function is deprecated and only available for backward comaptibility. Use `summarize` for more flexibility.

The output requires `\usepackage{booktabs}` in the LaTeX file.

Captions can be added to both, `longtables` and `tabulars`. In the latter case, captions are also suported if the table is no floating object. In this case, the LaTeX package `capt-of` is required.

Value

The output is printed with LaTeX style syntax highlighting to be used e.g. in Sweave chunks with `results=tex`.

Author(s)

Benjamin Hofner

See Also

latex.table.cont and get_option

Examples

```
## Example requires package nlme to be installed and loaded
if (require("nlme")) {
    ## Use dataset Orthodont
    data(Orthodont, package = "nlme")

    ## Get summary for continuous variables
    latex.table.fac(Orthodont)

    ## Reorder data for table:
    latex.table.fac(Orthodont, variables = c("Sex", "Subject"))

    ## What happens in the display if we introduce some missing values:
    set.seed(1907)
    Orthodont$Sex[sample(nrow(Orthodont), 20)] <- NA
    latex.table.fac(Orthodont)
    latex.table.fac(Orthodont, variables = "Sex")
    ## do not show statistics on missing values
    latex.table.fac(Orthodont, variables = "Sex", show.NAs = FALSE)
}
```

---

prettify *Make Pretty Summary and Anova Tables*

Description

Improve summary tables by replacing variable names with labels and separating variable names and value labels of factor variables. Additionally, confidence intervalls are added to summaries per default and p-values are formated for pretty printing.

Usage

```
## generic function called by all prettify.summary.xxx functions
## S3 method for class 'data.frame'
prettify(object, labels = NULL, sep = ": ", extra.column = FALSE,
         smallest.pval = 0.001, digits = NULL, scientific = FALSE,
         signif.stars = getOption("show.signif.stars"), ...)

## S3 method for class 'summary.lm'
prettify(object, labels = NULL, sep = ": ", extra.column = FALSE,
         confint = TRUE, level = 0.95,
         smallest.pval = 0.001, digits = NULL, scientific = FALSE,
         signif.stars = getOption("show.signif.stars"), ...)
```

```
## S3 method for class 'summary.glm'
prettify(object, labels = NULL, sep = ": ", extra.column = FALSE,
         confint = TRUE, level = 0.95, OR = TRUE,
         smallest.pval = 0.001, digits = NULL, scientific = FALSE,
         signif.stars = getOption("show.signif.stars"), ...)

## S3 method for class 'summary.coxph'
prettify(object, labels = NULL, sep = ": ", extra.column = FALSE,
         confint = TRUE, level = 0.95, HR = TRUE,
         smallest.pval = 0.001, digits = NULL, scientific = FALSE,
         signif.stars = getOption("show.signif.stars"),
         env = parent.frame(), ...)

## S3 method for class 'summary.lme'
prettify(object, labels = NULL, sep = ": ", extra.column = FALSE,
         confint = TRUE, level = 0.95,
         smallest.pval = 0.001, digits = NULL, scientific = FALSE,
         signif.stars = getOption("show.signif.stars"), ...)

## method for mixed models fitted with lme4 (vers. < 1.0)
## S3 method for class 'summary.mer'
prettify(object, labels = NULL, sep = ": ", extra.column = FALSE,
         confint = TRUE, level = 0.95,
         smallest.pval = 0.001, digits = NULL, scientific = FALSE,
         signif.stars = getOption("show.signif.stars"),
         simulate = c("ifneeded", TRUE, FALSE), B = 1000, ...)

## method for mixed models fitted with lme4 (vers. >= 1.0)
## S3 method for class 'summary.merMod'
prettify(object, labels = NULL, sep = ": ", extra.column = FALSE,
         confint = TRUE, level = 0.95,
         smallest.pval = 0.001, digits = NULL, scientific = FALSE,
         signif.stars = getOption("show.signif.stars"),
       method = c("profile", "Wald", "boot"), B = 1000, env = parent.frame(), ...)

## S3 method for class 'anova'
prettify(object, labels,
         smallest.pval = 0.001, digits = NULL, scientific = FALSE,
         signif.stars = getOption("show.signif.stars"), ...)

## helper function for pretty p-values
prettifyPValue(object, smallest.pval = 0.001, digits = NULL,
               scientific = FALSE,
               signif.stars = getOption("show.signif.stars"), ...)
```

Arguments

object	object of class data.frame resulting (most likely) from a call to summary or directly the output from summary, anova or Anova (the latter from package **car**).

labels	specify labels here. For the format see labels.
sep	separator between variable label and value label of a factor variable (default: ": ").
extra.column	logical: provide an extra column for the value labels of factors (default: FALSE).
confint	logical value indicating if confidence intervals sould be added or the confidence intervals themself. Using confint = TRUE is experimental only and special care needs to be taken that the data set used for fitting is neither changed nor deleted. See 'Details' and 'Examples'.
level	confidence level; Per default 0.95% confidence intervals are returned
OR	logical. Should odds ratios be added? Only applicable if a logistic regression model was fitted (i.e., with family = "binomial").
HR	logical. Should hazard ratios be added?
smallest.pval	determines the smallest p-value to be printed exactly. Smaller values are given as "< smallest.pval". This argument is passed to the eps argument of format.pval. See there for details.
digits	number of significant digits. The default, NULL, uses getOption("digits") for formating p-values and leaves all other columns unchanged. If digits are specified, all columns use this number of significant digits (columnwise). See also argument digits in format.
scientific	specifies if numbers should be printed in scientific format. For details and possible values see format.
signif.stars	logical (default = TRUE). Should significance stars be added? Per default system options are used. See getOption("show.signif.stars").
simulate	should the asysmptotic or simulated confidence intervals be used? See confint.mer for details.
B	number of samples to take in mcmcsamp. See confint.mer for details.
method	Determines the method for computing confidence intervals; One of "profile" (default), "Wald", "boot". For details see confint.merMod in package **lme4**.
...	further options. Currently not applicable.
env	specify environment in which the model was fitted. Needed to find the correct data for refitting the model in order to obtain confidence intervals.

Details

Specialized functions that prettify summary tables of various models exist. For the data.frame method, extra.column and sep can only be used if labels are specified as variable names need to be known in order to split variable name and factor level. For summary objects, variable names can be extracted from the objects.

To compute confidence intervalls, the model is refitted internally extrating the call and environment from the model summary. All functions then use confint on the refitted model. For mer models special confint functions are defined in this package (for backward compatibility). For details see there. Note that is it highly important **not** to modify or delete the data in the fitting environment if

one wants to obtain correct confidence intervals. See examples for what might happen. We try ourt best to find changes of the data and to warn the user (but without any warranty).

Alternatively, one can directly specify the confidence intervals using e.g. `confint = confint(model)`, where `model` is the fitted model. This does not rely on refitting of the model and should always work as expected. In this case, arguments `level`, `simulate` and `B` are ignored. Note that in this case it is adviced to also specify the labels by hand!

`prettifyPValue` is a helper function used within the prettify functions but can also be used directly on a `data.frame` object. The function tries to (cleverly) "guess" the column of p-values (based on the column names) and formats them nicely. Additionally, significance stars are added if requested.

Value

`data.frame` with prettier variable labels. For summary functions additionally confidence intervalls (if requested), odds ratio (for logistic regression models, if requested), p-values formated for pretty printing and significance stars (if requested) are attached.

Author(s)

Benjamin Hofner

See Also

`summary`, `summary.lm`, `summary.glm`, `summary.lme`, `summary.merMod` (or `summary.mer-class` in **lme4** < 1.0) and `summary.coxph` for summary functions.

`anova` and `Anova` for ANOVA functions.

`confint` and `ci` for confidence intervals. Special functions are implemented for mixed models: `confint.mer`.

Examples

```
## Example requires package nlme to be installed and loaded
if (require("nlme")) {
    ## Load data set Orthodont
    data(Orthodont, package = "nlme")

    ######################################################################
    # Linear model
    ######################################################################

    ## Fit a linear model
    linmod <- lm(distance ~ age + Sex, data = Orthodont)
    ## Extract pretty summary
    prettify(summary(linmod))

    ## Extract anova (sequential tests)
    anova(linmod)
    ## now prettify it
    prettify(anova(linmod))

    ######################################################################
```

```
# Random effects model (nlme)
##############################################################################

### (fit a more suitable model with random effects)
## With package nlme:
require("nlme")
## Fit a model for distance with random intercept for Subject
mod <- lme(distance ~ age + Sex, data = Orthodont, random = ~ 1 | Subject)
summary(mod)
## Extract fixed effects table, add confidence interval and make it pretty
prettify(summary(mod))
## Extract fixed effects table only and make it pretty
prettify(summary(mod), confint = FALSE)

##############################################################################
# Random effects model (lme4)
##############################################################################

set.seed(130913)

## With package lme4:
if (require("lme4") && require("car")) {
    ## Fit a model for distance with random intercept for Subject
    mod4 <- lmer(distance ~ age + Sex + (1|Subject), data = Orthodont)
    summary(mod4)
    ## Extract fixed effects table and make it pretty
    prettify(summary(mod4))

    ## Extract and prettify anova (sequential tests)
    prettify(anova(mod4))

    ## Better: extract Anova (partial instead of sequential tests)
    library("car")
    Anova(mod4)
    ## now prettify it
    prettify(Anova(mod4))
}

##############################################################################
# Cox model
##############################################################################

## survival models
if (require("survival")) {
    ## Load data set ovarian (now part of cancer)
    data(cancer, package = "survival")

    ## fit a Cox model
    mod5 <- coxph(Surv(futime, fustat) ~ age, data=ovarian)
    summary(mod5)
    ## Make pretty summary
    prettify(summary(mod5))
```

```
    ## Make pretty summary
    prettify(Anova(mod5))
}


######################################################################
# ATTENTION when confint = TRUE: Do not modify or delete data
######################################################################

## Fit a linear model (same as above)
linmod <- lm(distance ~ age + Sex, data = Orthodont)
## Extract pretty summary
prettify(summary(linmod))

## Change the data (age in month instead of years)
Orthodont$age <- Orthodont$age * 12
prettify(summary(linmod))  ## confidence intervals for age have changed
                           ## but coefficients stayed the same; a
                           ## warning is issued

## Remove data in fitting environment
rm(Orthodont)
prettify(summary(linmod))  ## confidence intervals are missing as no
                           ## data set was available to refit the model



######################################################################
# Use confint to specify confidence interval without refitting
######################################################################

## make labels without using the data set
labels <- c("distance", "age", "Subject", "Sex")
names(labels) <- labels

## usually easier via: labels(Orthodont)

prettify(summary(linmod), confint = confint(linmod),
         labels = labels)
}
```

---

summarize *Produce Summary Tables for Data Sets*

Description

The function produces summary tables for factors and continuous variables. The obtained tables can be used directly in R, with LaTeX and HTML (by using the `xtable` function) or Markdown (e.g. by using the function `kable`).

Usage

```
summarize(data, type = c("numeric", "factor"),
    variables = names(data), variable.labels = labels, labels = NULL,
    group = NULL, test = !is.null(group), colnames = NULL,
    digits = NULL, digits.pval = 3, smallest.pval = 0.001,
    sep = NULL, sanitize = TRUE, drop = TRUE,
    show.NAs = any(is.na(data[, variables])), ...)
```

Arguments

data	data set to be used.
type	print summary table for either numeric or factor variables.
variables	character vector defining variables that should be included in the table. Per default, all numeric or factor variables of data are used, depending on type.
variable.labels, labels	labels for the variables. If variable.labels = NULL (default) variables is used as label. If variable.labels = TRUE, labels(data, which = variables) is used as labels. Instead of variable.labels one can also use labels.
group	character specifying a grouping factor. Per default no grouping is applied.
test	logical or character string. If a group is given, this argument determines whether a test for group differences is computed. For details see summarize_numeric and summarize_factor.
colnames	a vector of character strings of appropriate length. The vector supplies alternative column names for the resulting table. If NULL default names are used.
digits	number of digits to round to. For defaults see summarize_numeric and summarize_factor.
digits.pval	number of significant digits used for p-values.
smallest.pval	determines the smallest p-value to be printed exactly. Smaller values are given as "< smallest.pval". This argument is passed to the eps argument of format.pval. See there for details.
sep	logical. Determines whether separators (lines) should be added after each variable. For defaults see summarize_numeric and summarize_factor.
sanitize	logical (default: TRUE) or a sanitizing function used to clean the input in order to be useable in LaTeX environments. Per default toLatex.character is used.
drop	logical (default: TRUE). Determines whether variables, which contain only missing values are dropped from the table.
show.NAs	logical. Determines if NAs are displayed. Per default, show.NAs is TRUE if there are any missings in the variables to be displayed (and FALSE if not). For details see summarize_numeric and summarize_factor.
...	additional arguments for summarize_numeric and summarize_factor. See there for details.

Value

A special data.frame with additional class summary containing the computed statistics is returned from function summarize. Addtional attributes required for the xtable.summary or print.xtable.summary function are contained as attributes. These are extracted using the function get_option.

Author(s)

Benjamin Hofner

See Also

For details see summarize_numeric and summarize_factor.

Conversion to LaTeX tables can be done using xtable.summary and print.xtable.summary.

get_option

Examples

```
if (require("nlme")) {
    ## Use dataset Orthodont
    data(Orthodont, package = "nlme")

    ## Get summary for continuous variables
    (tab1 <- summarize(Orthodont, type = "numeric"))

    ## Change statistics to display
    summarize(Orthodont, quantiles = FALSE, type = "numeric")
    summarize(Orthodont, quantiles = FALSE, count = FALSE, type = "numeric")
    summarize(Orthodont, mean_sd = FALSE, type = "numeric")

    ## Get summary for categorical variables
    (tab2 <- summarize(Orthodont, type = "fac"))

    ## use fraction instead of percentage
    summarize(Orthodont, percent = FALSE, type = "fac")

    ## Using the tables with Markdown
    if (require("knitr")) {
        kable(tab1)
        kable(tab2)
    }

    ## Using the tables with LaTeX
    if (require("xtable")) {
        xtable(tab1)
        ## grouped table
        xtable(summarize(Orthodont, group = "Sex"))
        xtable(tab2)
    }
}
```

---

summarize_factor *Produce Summary Tables for Data Sets*

Description

The function produces summary tables for factor variables. The obtained tables can be used directly in R, with LaTeX and HTML (by using the xtable function) or Markdown (e.g. by using the function kable).

Usage

```
summarize_factor(data,
    variables = names(data), variable.labels = labels, labels = NULL,
    group = NULL, test = !is.null(group), colnames = NULL,
    digits = 3, digits.pval = 3, smallest.pval = 0.001,
    sep = TRUE, sanitize = TRUE, drop = TRUE,
    show.NAs = any(is.na(data[, variables])),
    ## additional specific arguments
    percent = TRUE, cumulative = FALSE,
    na.lab = "<Missing>", ...)
```

Arguments

data	data set to be used.
variables	variables that should be included in the table. For details see summarize.
variable.labels, labels	labels for the variables. For details see summarize.
group	character specifying a grouping factor. For details see summarize.
test	logical or charachter specifying test for group differences. For details see summarize.
colnames	a vector of character strings of appropriate length. For details see summarize.
digits	number of digits to round to (only used for fractions). Per default all values are rounded to three digits.
digits.pval	number of significant digits used for p-values.
smallest.pval	determines the smallest p-value to be printed exactly. For details see summarize.
sep	logical (default: TRUE). Determines whether separators (lines) should be added after each variable.
sanitize	logical (default: TRUE) or a sanitizing function. For details see summarize.
drop	logical (default: TRUE). Determines whether variables, which contain only missing values are dropped from the table.
show.NAs	logical. Determines if NAs are displayed as a separate category for each factor variable with missings. If TRUE, an additional statistic which includes the missings is displayed (see Examples). Per default, show.NAs is TRUE if there are any missings in the variables to be displayed (and FALSE if not).
percent	logical. Should the fractions be given as percent values? Otherwise, fractions are given as decimal numbers.
cumulative	logical. Should cumulative fractions be displayed?
na.lab	label for missing values (default: "<Missing>").
...	additional arguments. Currently not used.

Value

A special `data.frame` with additional class `summary` containing the computed statistics is returned from function `summarize`. Addtional attributes required for the `xtable.summary` or `print.xtable.summary` function are contained as attributes. These are extracted using the function `get_option`.

Author(s)

Benjamin Hofner

See Also

For details see `link{summarize}` and `link{summarize_factor}`.

Conversion to LaTeX tables can be done using `xtable.summary` and `print.xtable.summary`.

`get_option`

Examples

```
## Example requires package nlme to be installed and loaded
if (require("nlme")) {
    ## Use dataset Orthodont
    data(Orthodont, package = "nlme")

    ## Get summary for continuous variables
    summarize(Orthodont, type = "factor")

    ## Reorder data for table:
    summarize(Orthodont, variables = c("Sex", "Subject"), type = "factor")

    ## What happens in the display if we introduce some missing values:
    set.seed(1907)
    Orthodont$Sex[sample(nrow(Orthodont), 20)] <- NA
    summarize(Orthodont, type = "factor")
    summarize(Orthodont, variables = "Sex", type = "factor")
    ## do not show statistics on missing values
    summarize(Orthodont, variables = "Sex", show.NAs = FALSE, type = "factor")
}
```

`summarize_numeric` *Produce Summary Tables for Data Sets*

Description

The function produces summary tables for continuous variables. The obtained tables can be used directly in R, with LaTeX and HTML (by using the `xtable` function) or Markdown (e.g. by using the function `kable`).

Usage

```
summarize_numeric(data,
    variables = names(data), variable.labels = labels, labels = NULL,
    group = NULL, test = !is.null(group), colnames = NULL,
    digits = 2, digits.pval = 3, smallest.pval = 0.001,
    sep = !is.null(group), sanitize = TRUE,
    drop = TRUE, show.NAs = any(is.na(data[, variables])),
    ## additional specific arguments
    count = TRUE, mean_sd = TRUE, quantiles = TRUE,
    incl_outliers = TRUE, ...)
```

Arguments

data	data set to be used.
variables	variables that should be included in the table. For details see summarize.
variable.labels, labels	labels for the variables. For details see summarize.
group	character specifying a grouping factor. For details see summarize.
test	logical or charachter specifying test for group differences. For details see summarize.
colnames	a vector of character strings of appropriate length. For details see summarize.
digits	number of digits to round to. Per default all values are rounded to two digits.
digits.pval	number of significant digits used for p-values.
smallest.pval	determines the smallest p-value to be printed exactly. For details see summarize.
sep	logical (default: TRUE if grouping specified, FALSE otherwise). Determines whether separators (lines) should be added after each variable.
sanitize	logical (default: TRUE) or a sanitizing function. For details see summarize.
drop	logical (default: TRUE). Determines whether variables, which contain only missing values are dropped from the table.
show.NAs	logical. Determines if the number of missings (NAs) is displayed as a separate column. Per default, show.NAs is TRUE if there are any missings in the variables to be displayed (and FALSE if not).
count	(logical) indicator if number of complete cases ("n") should be included in the table (default: TRUE).
mean_sd	(logical) indicator if mean and standard deviation should be included in the table (default: TRUE).
quantiles	(logical) indicator if quantiles (including min and max) should be included in the table (default: TRUE).
incl_outliers	Per default we use fivenum to compute the quantiles (if quantiles = TRUE). If extreme values should be excluded from min/max in the table, boxplot(, plot = FALSE)$stats is used instead.
...	additional arguments. Currently not used.

Value

A special `data.frame` with additional class `summary` containing the computed statistics is returned from function `summarize`. Addtional attributes required for the `xtable.summary` or `print.xtable.summary` function are contained as attributes. These are extracted using the function `get_option`.

Author(s)

Benjamin Hofner

See Also

For details see `link{summarize}` and `link{summarize_factor}`.

Conversion to LaTeX tables can be done using `xtable.summary` and `print.xtable.summary`.

`get_option`

Examples

```
if (require("nlme")) {
    ## Use dataset Orthodont
    data(Orthodont, package = "nlme")

    ## Get summary for continuous variables
    summarize(Orthodont, type = "numeric")

    ## Change statistics to display
    summarize(Orthodont, quantiles = FALSE, type = "numeric")
    summarize(Orthodont, quantiles = FALSE, count = FALSE, type = "numeric")
    summarize(Orthodont, mean_sd = FALSE, type = "numeric")

    ## for more examples see ?summarize
}
```

---

toLatex *Cleaning R Code for printing in LaTeX environments*

Description

The function produces code that LaTeX is able to typeset.

Usage

```
## S3 method for class 'character'
toLatex(object, ...)

## S3 method for class 'sessionInfo'
toLatex(object, pkgs = NULL, locale = FALSE,
        base.pkgs = FALSE, other.pkgs = TRUE,
        namespace.pkgs = FALSE, citations = TRUE,
```

```
citecommand = "\\citep", file = NULL,
append = FALSE, ...)
```

Arguments

object	either an object of class character which should be cleaned for printing in LaTeX environments or an object of class sessionInfo.
pkgs	character vector (optional). Specify specific packages here to show information on these (instead of all attached packages). See package in sessionInfo.
locale	logical (default = FALSE). Show information on locale.
base.pkgs	logical (default = FALSE). Show information on base packages.
other.pkgs	logical (default = TRUE). Show information on currently attached packages. If pkgs is specified, information on these packages is given instead of all attached packages.
namespace.pkgs	logical (default = FALSE). Show information on packages whose namespaces are currently loaded but not attached.
citations	logical (default = TRUE). Should citations for all packages be added? BibTeX is used for storing the citations.
citecommand	Specify LaTeX-command for citation here. Curly brackets are added internally. Note that \ needs to be escaped, i.e., one needs to write \\ instead.
file	Specify path to BibTeX file where citations should be saved. If file = NULL is specified, the BibTeX entries are attached to the output as attribute "BibTeX". See examples for details.
append	logical (default = FALSE). Should citations be added to an existing BibTeX file (if existing) or should old BibTeX files be overwritten?
...	additional arguments. Currently not used.

Value

A character string with special markup is returned: The output is printed with LaTeX style syntax highlighting to be used e.g. in Sweave chunks with results=tex.

Author(s)

Benjamin Hofner, based on code from package xtable, bibtex and package utils. See source code for documentation.

See Also

toLatex. For details on toLatex.sessionInfo see also sessionInfo.

Examples

```
txt <- "Price: <= 500$ & additional goodies"
toLatex(txt)

############################################################
```

```
## session info for automatic inclusion in reports

info <- toLatex(sessionInfo())
info

## extract first part (the Latex part)
toLatex(info)
## extract second part (the BibTeX part)
toBibtex(info)


############################################################
## usual usage scenario

## Do not run the following code automatically as it needs
## write access to the current working directory.
## This code (without removing the file) could for example
## be included in a LaTeX chunk of your Sweave or knitr
## report.

## Not run: getwd()     ## location where write access is needed
toLatex(sessionInfo(), file = "packages.bib")
file.remove("packages.bib")

## End(Not run)
```

`xtable.summary` *Create And Print Tables With Markup*

Description

The function produces objects which can be printed to LaTeX and HTML code.

Usage

```
## S3 method for class 'summary'
xtable(x, caption = NULL, label = NULL, align = NULL,
       digits = NULL, display = NULL, ...)

## S3 method for class 'xtable.summary'
print(x, rules = NULL, header = NULL,
      caption.placement = getOption("xtable.caption.placement", "top"),
      hline.after = getOption("xtable.hline.after", NULL),
      include.rownames = FALSE,
      add.to.row = getOption("xtable.add.to.row", NULL),
      booktabs = getOption("xtable.booktabs", TRUE),
      sanitize.text.function = get_option(x, "sanitize"),
      math.style.negative = getOption("xtable.math.style.negative", TRUE),
```

```
math.style.exponents = getOption("xtable.math.style.exponents", TRUE),
tabular.environment = getOption("xtable.tabular.environment", "tabular"),
floating = getOption("xtable.floating", FALSE),
latex.environments = getOption("xtable.latex.environments", c("center")),
...)
```

Arguments

x	object of class "summary", which is produced by the function summarize or an object of class "xtable.summary" produced by xtable.
caption	character vector specifying the table's caption; see xtable for details.
label	character string specifying the LaTeX label or HTML anchor; see xtable for details.
align	character string specifying the alignment of table columns; see xtable for details.
digits	numeric vector specifying the number of digits to display in each column; see xtable for details.
display	character string specifying the column types; see xtable for details.
rules	character string specifying the rules to be used. Per default the rules are defined by summarize and subsequently extracted from x via get_option(x, "rules").
header	character string specifying the table header to be used. Per default the header is defined by summarize and subsequently extracted from x via get_option(x, "header").
caption.placement	can be either "bottom" or "top" (default). Note that the standard default of print.xtable is "bottom".
hline.after	vector indicating the rows after which a horizontal line is printed. Here, the default is to not draw hlines (i.e. hline.after = NULL) and horizontal lines are added via add.to.row (see there for details). Note that the standard default of print.xtable is c(-1,0,nrow(x)).
add.to.row	list of row numbers (pos) and text (command) to be added to the specified rows. Per default, top and bottom rules are added to the table and a rule specified in rules is added below the heading. If sep = TRUE in summarize additional separators (as specified in rules) are added after each variable.
include.rownames	logical. Always set to FALSE.
booktabs	logical. If TRUE (default), the toprule, midrule and bottomrule tags from the LaTeX package "booktabs" are used rather than hline for the horizontal line tags. Note that the standard default of print.xtable is FALSE.
sanitize.text.function	All non-numeric enteries (except row and column names) are sanitised in an attempt to remove characters which have special meaning for the output format. Per default the function toLatex is used to sanitize the text. For more options see print.xtable.

`math.style.negative`
: logical. If `TRUE` (default) the negative sign is wrapped in dollar signs for LaTeX tables. Note that the standard default of `print.xtable` is `FALSE`.

`math.style.exponents`
: logical. If `TRUE` (default) scientific numers are set as exponents. See `print.xtable` for details. Note that the standard default of `print.xtable` is `FALSE`.

`tabular.environment`
: character string. Per default `"tabular"` is used. For long tables that span over more than one page, one can use `"longtable"`. For more options see `print.xtable`.

`floating`
: logical. Determine if the table is printed in a floating environment. Note that the standard default of `print.xtable` is `TRUE`. See there for details.

`latex.environments`
: character string. Per default `"center"` is used. In contrast to the default behavior of `print.xtable`, tables are also centered if no floating environment is used. For details and more options see `print.xtable`.

`...`
: additional arguments passed to `xtable` or `print.xtable`. See there for details.

Details

We use the standard `xtable` function but add a special class that allows different defaults in the `print.xtable` function.

In general, all options of `print.xtable` can be used as well as global options set via `options()`. E.g. `options(xtable.booktabs = FALSE` will set the argument `booktabs` per default to `FALSE` for all calls to `print.xtable`.

Value

After printing, a table with LaTeX markup is returned.

Author(s)

Benjamin Hofner

See Also

For details see `xtable` and `print.xtable`.

`summarize`, `get_option`

Examples

```
if (require("nlme")) {
    ## Use dataset Orthodont
    data(Orthodont, package = "nlme")

    ## Get summary for continuous variables
    (tab1 <- summarize(Orthodont, type = "numeric"))

    ## Get summary for categorical variables
```

```
    (tab2 <- summarize(Orthodont, type = "fac"))

    ## Using the tables with LaTeX
    if (require("xtable")) {
        xtable(tab1)
        ## grouped table
        xtable(summarize(Orthodont, group = "Sex"))
        xtable(tab2)
    }
}
```

Index