

# Package ‘paper’

July 2, 2014

**Title** A toolbox for writing Sweave or other LaTeX-based papers and reports

**Version** 0.4-0

**Date** 2013-09-13

**Author** Benjamin Hofner

**Maintainer** Benjamin Hofner <benjamin.hofner@fau.de>

**Description** A toolbox for writing Sweave or other LaTeX-based reports

**Depends** car, nlme, lme4, survival, utils

**Imports** gmodels

**License** GPL-2

**URL** <http://r-forge.r-project.org/projects/paper/>

**Collate** 'helpers.R' 'labels.R' 'latex\_table.R' 'plot.R' 'prettify.R' 'toLatex.R'

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2013-09-13 11:27:24

## R topics documented:

papeR-package	2
Anova.lme	2
confint.lme	3
get_options	4
labels	5
latex.table.cont	8
latex.table.fac	11
prettify	14
toLatex	19
<b>Index</b>	<b>22</b>

---

paper-package

*A toolbox for writing Sweave or other LaTeX-based papers and reports*

---

### Description

A toolbox for writing Sweave or other LaTeX-based papers and reports

### Details

Package: papeR  
 Type: Package  
 Version: 0.4  
 Date: 2013-08-07  
 License: GPL-2

### 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**

```
## 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.lme

*Confidence intervals for mixed models*


---

**Description**

Compute confidence intervals for mixed models from packages **nlme** and **lme4**

**Usage**

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

**Arguments**

object	Model of class <code>lme</code> or <code>mer</code> .
parm	Parameters to be included in the confidence interval. See <a href="#">confint.default</a> for details.
level	the confidence level.
simulate	If <code>"ifneeded"</code> 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 elsewhere. If <code>TRUE</code> (or <code>"TRUE"</code> ) confidence intervals will be estimated using <a href="#">ci</a> from package <b>gmodels</b> which uses <a href="#">mcmcscamp</a> internally. If <code>FALSE</code> (or <code>"FALSE"</code> ), asymptotic confidence intervals will be returned and an error is given if not possible.
B	number of samples to take in <a href="#">mcmcscamp</a> . 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\_options

*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_options(object, name)
```

**Arguments**

object	table.cont or table.fac object as produced by <a href="#">latex.table.cont</a> and <a href="#">latex.table.fac</a>
name	name of the option, e.g. "table" and "align". See <a href="#">latex.table.cont</a> and <a href="#">latex.table.fac</a> for available options.

**Value**

Option.

**Author(s)**

Benjamin Hofner

**See Also**

[latex.table.cont](#) and [latex.table.fac](#)

## Description

Labels can be stored to an attribute "variable.labels" using the assignment function. With the extractor function one can assess these labels. Usually, these labels are generated by [read.spss](#) in package **foreign**.

## 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
is.labeled.data.frame(object)
## set as.labeled.data.frame
as.labeled.data.frame(object, ...)

## special functions for labeled.data.frame objects that keep the labels
## S3 method for class 'labeled.data.frame'
x[... , drop = TRUE]
## S3 method for class 'labeled.data.frame'
subset(x, ...)
## S3 method for class 'labeled.data.frame'
cbind(..., deparse.level = 1)
## S3 method for class 'labeled.data.frame'
rbind(..., deparse.level = 1)

## special plotting function for labeled.data.frame objects
## S3 method for class 'labeled.data.frame'
plot(data, variables = names(data),
      labels = TRUE, group = NULL, ...)
```

## Arguments

object	a data.frame or labeled.data.frame. The former is usually a result from <a href="#">read.spss</a> in package <b>foreign</b> , the latter results from adding labels in R or from a call to <code>as.labeled.data.frame</code> .
data	a data.frame or a labeled.data.frame, where labels should be added or altered.

which	either a number indicating the label to extract or a character string with the <i>variable name</i> for which the label should be extracted. One can also use a vector of numerics or character strings to extract multiple 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 <a href="#">abbreviate</a> for details. Further arguments to <a href="#">abbreviate</a> can be specified (see below).
...	further options passed to function <a href="#">abbreviate</a> if argument abbreviate = TRUE. In <code>x[...]</code> , ... can be used to specify indices for extraction. See <a href="#">[</a> for details. In <code>plot, ...</code> can be used to specify further graphical parameters.
x	a <code>labeled.data.frame</code> .
drop	logical (default: TRUE). If TRUE the result is coerced to the lowest possible dimension (i.e. a vector in case of a single column) and labels might be dropped in this case.
deparse.level	see <a href="#">cbind</a> .
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), <code>labels(data, which = variables)</code> is used as labels. If labels = NULL variables is used as label. labels can also be specified as character vector.
group	character or numeric specifying a grouping factor. Per default no grouping is applied.

### Details

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. If labels are set (attached to a `data.frame`) the `data.frame` gets a special class `labeled.data.frame` with specific subset and combination functions.

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

### 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 you import data using `read.spss`, labels are set but the `data.frame` is not coerced to a `labeled.data.frame`. Use `as.labeled.data.frame` in this case or do any manipulation of labels using the assignment function (see examples below).

**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.labeled.data.frame(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 labeled.data.frame:  
is.labeled.data.frame(data)  
  
## Alternatively one could use as.labeled.data.frame(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
## (but only if data.set has class "labeled.data.frame")

## subsetting to single variables:
labels(tmp[, 2]) ## label got lost as tmp drops to vector
labels(tmp[, 2, drop = FALSE]) ## prevent dropping labels

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

#####
### Plotting labeled.data.set objects

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

```

## Description

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



**Usage**

```

latex.table.cont(data, variables = names(data), labels = NULL, group = NULL,
  colnames = NULL, digits = 2, table = c("tabular", "longtable"),
  align = NULL, caption = NULL, label = NULL, floating = FALSE,
  center = TRUE, sep = !is.null(group), sanitize = TRUE,
  count = TRUE, mean_sd = TRUE, quantiles = TRUE,
  incl_outliers = TRUE, drop = TRUE,
  show.NAs = any(is.na(data[, variables])))

## S3 method for class 'table.cont'
print(x,
  colnames = get_options(x, "colnames"),
  table = get_options(x, "table"),
  align = get_options(x, "align"),
  caption = get_options(x, "caption"),
  label = get_options(x, "label"),
  floating = get_options(x, "floating"),
  center = get_options(x, "center"),
  sep = get_options(x, "sep"),
  sanitize = get_options(x, "sanitize"),
  count = get_options(x, "count"),
  mean_sd = get_options(x, "mean_sd"),
  quantiles = get_options(x, "quantiles"),
  ...)

```

**Arguments**

data	data set to be used
variables	character vector defining (continuous) variables that should be included in the table. Per default, all non-factor variables of data are used.
labels	labels for the variables. If labels = NULL (default) variables is used as label. If labels = TRUE, labels(data, which = variables) is used as labels.
group	character specifying a grouping factor. Per default no grouping is applied.
colnames	a vector of character strings of length 9 (variable, N, standard deviation, minimum, first quartile, median, third quartile and maximum) or shorter if some of count, mean_sd or quantiles is FALSE. The vector supplies alternative column names for the resulting LaTeX table. If NULL default names are used.
digits	number of digits to round to.
table	character string. LaTeX table format, currently either "tabular" (default) or "longtable".
align	character string. LaTeX alignment of table rows, per default "lr...r", where "r" is repeated ncol - 1 times.
caption	(optional) character string. Caption of LaTeX table. Note that captions are supported for all tables (see also details below).

label	(optional) character string. Label of LaTeX table specified as <code>\label{"label"}</code> .
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.
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 used to clean the input in order to be useable in LaTeX environments. Per default <code>toLatex.character</code> is used.
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 <code>fivenum</code> to compute the quantiles (if <code>quantiles = TRUE</code> ). If extreme values should be excluded from min/max in the table, <code>boxplot( , plot = FALSE)\$stats</code> is used instead.
drop	logical (default: TRUE). Determines whether continuous 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, <code>show.NAs</code> is TRUE if there are any missings in the variables to be displayed (and FALSE if not).
x	An object of class <code>table.fac</code> . See Value for details.
...	additional arguments for printing. Currently not used.

### Details

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 supported if the table is no floating object. In this case, the LaTeX package `capt-of` is required.

### Value

A special `data.frame` with additional class `table.cont` containing the computed statistics is returned from function `latex.table.cont`. Additional attributes required for the print function are contained (see `attr(x, "latex.table.options")`). These are extracted in the print function using the function `get_options`.

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\\_options](#)

**Examples**

```
## 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)

## We can also have a look at the returned object:
df.cont <- latex.table.cont(Orthodont)
## Structure of object
str(df.cont)
## Make object a data.frame and have a look at the raw data
class(df.cont) <- "data.frame"
df.cont

print.table.cont(df.cont)
```

---

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(data, variables = names(data), labels = NULL, group = NULL,
```

```

colnames = NULL, digits = 2, table = c("tabular", "longtable"),
align = NULL, caption = NULL, label = NULL, floating = FALSE,
center = TRUE, sep = TRUE, sanitize = TRUE, drop = TRUE,
show.NAs = any(is.na(data[, variables])), na.lab = "<Missing>")

## S3 method for class 'table.fac'
print(x,
      colnames = get_options(x, "colnames"),
      table = get_options(x, "table"),
      align = get_options(x, "align"),
      caption = get_options(x, "caption"),
      label = get_options(x, "label"),
      floating = get_options(x, "floating"),
      center = get_options(x, "center"),
      sep = get_options(x, "sep"),
      sanitize = get_options(x, "sanitize"),
      ...)

```

## Arguments

data	data set to be used
variables	character vector defining (factor) variables that should be included in the table. Per default, all factor variables of data are used.
labels	labels for the variables. If labels = NULL (default) variables is used as label. If labels = TRUE, labels(data, which = variables) is used as labels.
group	character specifying a grouping factor. Per default no grouping is applied.
colnames	a vector of character strings of length 4 (variable, label, N, and fraction in %). The vector supplies alternative column names for the resulting LaTeX table. If NULL default names are used.
digits	number of digits to round to (only used for fractions).
table	character string. LaTeX table format, currently either "tabular" (default) or "longtable".
align	character string. LaTeX alignment of table rows, per default "l1r...r", where "r" is repeated ncol - 2 times.
caption	(optional) character string. Caption of LaTeX table. Note that captions are supported for all tables (see also details below).
label	(optional) character string. Label of LaTeX table specified as \label{"label"}.
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.
sep	logical (default: TRUE). Determines whether separators (lines) should be added after each factor variable.

sanitize	logical (default: TRUE) or a sanitizing function used to clean the input in order to be useable in LaTeX environments. Per default <code>toLatex.character</code> is used.
drop	logical (default: TRUE). Determines wether factor 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).
na.lab	label for missing values (default: "<Missing>").
x	An object of class <code>table.fac</code> . See Value for details.
...	additional arguments for printing. Currently not used.

### Details

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 supported if the table is no floating object. In this case, the LaTeX package `capt-of` is required.

### Value

A special `data.frame` with additional class `table.fac` containing the computed statistics is returned from function `latex.table.cont`. Additional attributes required for the print function are contained (see `attr(x, "latex.table.options")`). These are extracted in the print function using the function `get_options`.

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\\_options](#)

### Examples

```
## 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 intervals are added to summaries per default and p-values are formatted 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 <code>data.frame</code> resulting (most likely) from a call to <code>summary</code> or directly the output from <code>summary</code> , <code>anova</code> or <code>Anova</code> (the latter from package <code>car</code> ).
labels	specify labels here. For the format see <code>labels</code> .
sep	separator between variable label and value label of a factor variable (default: <code>": "</code> ).
extra.column	logical: provide an extra column for the value labels of factors (default: <code>FALSE</code> ).
confint	logical value indicating if confidence intervals should be added or the confidence intervals themselves. Using <code>confint = TRUE</code> 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 <code>family = "binomial"</code> ).
HR	logical. Should hazard ratios be added?
smallest.pval	determines the smallest p-value to be printed exactly. Smaller values are given as <code>"&lt; smallest.pval"</code> . This argument is passed to the <code>eps</code> argument of <code>format.pval</code> . See there for details.

<code>digits</code>	number of significant digits. The default, <code>NULL</code> , uses <code>getOption("digits")</code> for formatting p-values and leaves all other columns unchanged. If <code>digits</code> are specified, all columns use this number of significant digits (columnwise). See also argument <code>digits</code> in <code>format</code> .
<code>scientific</code>	specifies if numbers should be printed in scientific format. For details and possible values see <code>format</code> .
<code>signif.stars</code>	logical (default = <code>TRUE</code> ). Should significance stars be added? Per default system options are used. See <code>getOption("show.signif.stars")</code> .
<code>simulate</code>	should the asymptotic or simulated confidence intervals be used? See <code>confint.mer</code> for details.
<code>B</code>	number of samples to take in <code>mcmcSamp</code> . See <code>confint.mer</code> for details.
<code>method</code>	Determines the method for computing confidence intervals; One of "profile" (default), "Wald", "boot". For details see <code>confint.merMod</code> in package <b>lme4</b> .
<code>...</code>	further options. Currently not applicable.
<code>env</code>	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 intervals, the model is refitted internally extracting the call and environment from the model summary. All functions then use `confint` on the refitted model. For `lme` and `mer` models special `confint` functions are defined in this package. For details see there. Note that it is 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 our 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 advised 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 intervals (if requested), odds ratio (for logistic regression models, if requested), p-values formatted 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.lme](#) and [confint.mer](#).

**Examples**

```
## 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:
require("lme4")
## 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")
try(Anova(mod4)) ## doesn't currently work with lme4 >= vers. 1.0.0
## now prettify it
try(prettify(Anova(mod4))) ## doesn't currently work with lme4 >= vers. 1.0.0

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

library("survival")
## Load data set ovarian
data(ovarian, 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)
```

---

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 = "Rpackages.bib",
         append = FALSE, ...)
```

## Arguments

object	either an object of class <code>character</code> which should be cleaned for printing in LaTeX environments or an object of class <code>sessionInfo</code> .
pkgs	character vector (optional). Specify specific packages here to show information on these (instead of all attached packages). See package in <code>sessionInfo</code> .
locale	logical (default = <code>FALSE</code> ). Show information on locale.
base.pkgs	logical (default = <code>FALSE</code> ). Show information on base packages.
other.pkgs	logical (default = <code>TRUE</code> ). Show information on currently attached packages. If <code>pkgs</code> is specified, information on these packages is given instead of all attached packages.
namespace.pkgs	logical (default = <code>FALSE</code> ). 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(), file = NULL)
info

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

#####
## usual usage scenario

## Do not run 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)
```

# Index

- \*Topic **IO**
  - get\_options, 4
  - latex.table.cont, 8
  - latex.table.fac, 11
  - toLatex, 19
- \*Topic **methods**
  - Anova.lme, 2
  - labels, 5
  - prettify, 14
- \*Topic **models**
  - Anova.lme, 2
  - prettify, 14
- \*Topic **package**
  - papeR-package, 2
- \*Topic **print**
  - get\_options, 4
  - latex.table.cont, 8
  - latex.table.fac, 11
  - toLatex, 19
- \*Topic **univar**
  - latex.table.cont, 8
  - latex.table.fac, 11
  - toLatex, 19
- [, 6
- [.labeled.data.frame (labels), 5
  
- abbreviate, 6
- Anova, 2, 3, 15, 17
- anova, 15, 17
- Anova.lme, 2
- anova.lme, 2
- as.labeled.data.frame (labels), 5
- as.labelled.data.frame (labels), 5
  
- cbind, 6
- cbind.labeled.data.frame (labels), 5
- ci, 3, 17
- confint, 16, 17
- confint.default, 3
- confint.lme, 3, 17
  
- confint.mer, 16, 17
- confint.mer (confint.lme), 3
  
- data.frame, 6
  
- fivenum, 10
- format, 16
- format.pval, 15
  
- get\_options, 4, 10, 11, 13
  
- is.labeled.data.frame (labels), 5
- is.labelled.data.frame (labels), 5
  
- labels, 5, 15
- labels.data.frame (labels), 5
- labels.default (labels), 5
- labels<- (labels), 5
- latex.table.cont, 4, 8, 13
- latex.table.fac, 4, 11, 11
  
- mcmcSamp, 3, 16
  
- papeR (papeR-package), 2
- papeR-package, 2
- plot.labeled.data.frame (labels), 5
- prettify, 14
- prettifyPValue (prettify), 14
- print.table.cont (latex.table.cont), 8
- print.table.fac (latex.table.fac), 11
  
- rbind.labeled.data.frame (labels), 5
- read.spss, 5–7
  
- sessionInfo, 19, 20
- subset.labeled.data.frame (labels), 5
- summary, 15–17
- summary.coxph, 17
- summary.glm, 17
- summary.lm, 17
- summary.lme, 17

toLatex, [19](#), [20](#)

toLatex.character, [10](#), [13](#)