

# Package ‘R3port’

May 7, 2026

**Type** Package

**Version** 0.3.1

**Title** Report Functions to Create HTML and PDF Files

**Description** Create and combine HTML and PDF reports from within R.  
Possibility to design tables and listings for reporting and also include R plots.

**Depends** R (>= 3.2.0)

**Imports** plyr, reshape2, whisker, tools, grDevices, methods, stats,  
utils, tinytex

**License** GPL (>= 2)

**Suggests** ggplot2, knitr, rmarkdown, testthat, lifecycle

**RoxygenNote** 7.3.2

**VignetteBuilder** knitr

**URL** <https://github.com/RichardHooijmaijers/R3port>

**Encoding** UTF-8

**NeedsCompilation** no

**Author** Richard Hooijmaijers [aut, cre],  
Richard Hooijmaijers [cph]

**Maintainer** Richard Hooijmaijers <[richardhooijmaijers@gmail.com](mailto:richardhooijmaijers@gmail.com)>

**Repository** CRAN

**Date/Publication** 2025-05-15 07:10:07 UTC

## Contents

freq . . . . .	2
html_combine . . . . .	3
html_doc . . . . .	4
html_list . . . . .	6
html_plot . . . . .	7
html_table . . . . .	9
html_table_design . . . . .	11

ltx_combine . . . . .	12
ltx_doc . . . . .	13
ltx_list . . . . .	15
ltx_plot . . . . .	17
ltx_table . . . . .	19
ltx_table_design . . . . .	21
means . . . . .	23
R3port . . . . .	24
table_prep . . . . .	25

<b>Index</b>	<b>26</b>
--------------	-----------

---

freq	<i>Calculate frequency statistics on data frame</i>
------	-----------------------------------------------------

---

### Description

This function calculates frequency statistics on a data frame. It is intended to use to calculate the frequency and percentages based on a predefined denominator.

### Usage

```
freq(
  dfrm,
  vars,
  id,
  denom = nrow(dfrm),
  dig = 2,
  total = NULL,
  totaldenom = nrow(dfrm),
  spacechar = NULL
)
```

### Arguments

dfrm	data frame to calculate the statistics on
vars	character vector of variable(s) within the data frame to perform the statistics on
id	character vector of variable(s). In case id is set the function will take into account non duplicated values for id in the calculations.
denom	the denominator for the calculation of percentage. See details for valid options
dig	number of digits to use for output percentage (frequency is displayed as integer)
total	vector of variable(s) within the data frame to calculate totals on
totaldenom	the denominator for the calculation of percentage of the totals. See details for valid arguments
spacechar	the character to use for space for combined frequency and percentage in output (e.g. "~" for tex documents)

**Details**

The function calculates frequency statistics of variable(s) within a data frame. Furthermore, the frequencies for a total can also be generated. To calculate the percentages, the denominator should be supplied (and in case totals has a value also totaldenom). The denom and totaldenom can be a constant numerical value, a variable within the data frame or a separate data frame. In case the argument is a data frame, the function will attempt to merge this data frame based on equal variables within dfrm/denom and dfrm/totaldenom. The possibility for a separate denominator is implemented as in many cases the denominator is not the total number of observations but is defined elsewhere (e.g. for number of adverse events, the total number of subjects is used as denominator).

**Value**

The function returns a dataframe with frequencies and percentages

**Examples**

```
data(Indometh)
freq(Indometh,vars="time",id="Subject",
     denom=nrow(Indometh),total="",totaldenom=nrow(Indometh))
```

---

 html\_combine

---

*Combines multiple HTML files to a single tex and compiles document*


---

**Description**

This function combines multiple HTML files. This is done based on the name of the files and should end with raw.tex to make the function pick-up these files.

**Usage**

```
html_combine(
  combine = ".",
  out = NULL,
  totheme = TRUE,
  css = paste0(system.file(package = "R3port"), "/style.css"),
  clean = 0,
  ...
)
```

**Arguments**

combine	character string with the location of the raw html files or list with file names within same directory
out	filename for the output HTML file (if NULL it will print to console)
totheme	logical indicating if the created file should also have a toc (take into account for template)

css	character with name of the css style sheet to use, default use package style sheet
clean	integer between 0 and 2 indicating if all individual files should be kept (0), all individual raw html files should be deleted (1) or all individual files should be deleted (2)
...	additional arguments passed through to <code>html_doc()</code> . Most important are template, rendlist, css and show

### Details

Currently the generated output is saved in the same place where the separate tables and plots are located defined in the 'combine' argument. This is done even when a different file path is specified in 'out' (using the basename function). The reason is to not copy files linked within the document and preventing broken links. This behaviour might change in future releases

### Value

The function returns a HTML file (or writes output to console)

### Examples

```
# Take into account the usage of tempfile() with multiple function calls
data(Theoph)
html_list(Theoph[1:11,],out=tempfile(fileext=".html"),show=FALSE)
html_plot(plot(conc~Time,data=Theoph),out=tempfile(fileext=".html"),show=FALSE)
## Not run:
  html_combine(combine=tempdir(),out="rep1.html")

# tocthem can be used to have a clickable toc,
# a bootstrap template for this is provided in the package
html_combine(combine=tempdir(),out="rep1.html",
             template=paste0(system.file(package="R3port"),"/bootstrap.html"),
             tocthem=TRUE)

## End(Not run)
```

---

html\_doc

*Prints a HTML table, listing or plot to a file or console*

---

### Description

This function makes a HTML document using output generated with functions in the R3port package or any other HTML code available as vector. Basically it adds tags to a html template file and let's the user select various options to customize the output.

**Usage**

```
html_doc(
  text,
  out = NULL,
  show = TRUE,
  rtitle = "report",
  template = paste0(system.file(package = "R3port"), "/simple.html"),
  rendlist,
  css = paste0(system.file(package = "R3port"), "/style.css")
)
```

**Arguments**

text	character vector to be placed within HTML document
out	character with filename for the output HTML file (if NULL it will print to console)
show	logical indicating if the resulting pdf file from the compiled tex file should be opened when created
rtitle	string indicating the title of the output document
template	character with file name of the template file to use
rendlist	list with render items to be used for the template file (see <a href="#">whisker::whisker.render</a> )
css	character with name of the css style sheet to use, default use package style sheet

**Details**

This function is used as wrapper within multiple functions in the R3port package but is also convenient in case custom information should be placed within an output document

**Value**

The function returns a HTML file (or writes output to console)

**See Also**

[ltx\\_doc\(\)](#)

**Examples**

```
txt <- "<h1>Example</h1>"
tbl <- "<table><tr><td id='fcol1'>table data</td><td>for custom table</td></tr></table>"
add <- "<p>Including some additional text</p>"
## Not run:
  html_doc(c(txt,tbl,add),out=tempfile(fileext=".html"))

## End(Not run)
```

---

html_list	<i>Creates a HTML listing</i>
-----------	-------------------------------

---

### Description

This function creates a HTML listing which can be written to a file or console

### Usage

```
html_list(
  dfrm,
  vars = names(dfrm),
  fill = "",
  vargroup = NULL,
  porder = TRUE,
  uselabel = TRUE,
  footnote = NULL,
  title = "listing",
  titlepr = NULL,
  group = NULL,
  xrepeat = FALSE,
  tclass = "sample",
  out = NULL,
  rawout = paste0(out, ".rawhtml"),
  ...
)
```

### Arguments

dfrm	data frame to be listed
vars	character vector that defines the variables within the data frame to be placed in the table
fill	<b>[Deprecated]</b> character indicating the character to use in case of missing values this is only applicable for <a href="#">html_table</a>
vargroup	character vector of the same length as vars. Creates a first line in the table to group variables (see details)
porder	logical indicating if the data frame should be ordered on the variables given in vars
uselabel	logical indicating if labels should be used for the variable(s). If set to TRUE, the function will try to use the label attribute for the display of variable(s).
footnote	character string with the footnote to be placed in the footer of the page (HTML coding can be used for example to create line breaks)
title	character string to define the title of the output which will be added to the caption
titlepr	character string to define the prefix of the output title. Can be used to create custom table numbering

group	numeric indicating the last index of the used variables that should be grouped (displayed in table with a certain white space), interpreted as <code>x[1:group]</code>
xrepeat	logical indicating if duplicate x values should be repeated in the table or not
tclass	character string with the table class. Can be used in combination with custom css
out	filename for the output HTML file (if NULL it will print to console)
rawout	character string with the name of the raw HTML file to generate (e.g. only table) In case NULL no raw output will be generated. In order to combine results the filename should end in <code>.rawhtml</code>
...	additional arguments passed through to <code>html_doc()</code> . Most important are template, rendlist, css and show

### Details

The `vargroup` argument should be provided in the following form:  
`c(rep("", 4), rep("group1", 3), rep("group2", 4))`. The function will place the text within the vector with the given length as first line in the table with a midrule below it. an exception is made for empty strings. The format of the current listing/css was designed to look good in the browser and in a WORD document (when html is opened in WORD). Additional colspans are added in the table header which might lead to additional space in case a user defined css file is used.

### Value

The function returns a HTML file (or writes output to console)

### Examples

```
# an example how vargroup can be used
grp <- c(rep("", 3), rep("grouped variables", 2))
## Not run:
  data(Theoph)
  html_list(Theoph, out=tempfile(fileext=".html"))
  html_list(Theoph, out=tempfile(fileext=".html"), vargroup=grp)

## End(Not run)
```

---

html\_plot

*Prints a R plot to a HTML file or console*

---

### Description

This function makes a HTML document including the defined plots

**Usage**

```
html_plot(
  plot,
  out,
  title = "plot",
  titlepr = NULL,
  footnote = "",
  pwidth = 1000,
  pheight = 600,
  res = NULL,
  fontsize = 12,
  units = "px",
  rawout = paste0(out, ".rawhtml"),
  cleancur = FALSE,
  ...
)
```

**Arguments**

plot	plot object or function call that creates plot to be printed to file
out	filename for the output HTML file
title	character string to define the title of the table which will be added to the caption
titlepr	character string to define the prefix of the table title. Can be used to create custom table numbering
footnote	character string with the footnote to be placed in the footer of the page (HTML coding can be used for example to create line breaks)
pwidth	numeric indicating the width of the plot to be generated in pixels
pheight	numeric indicating the height of the plot to be generated in pixels
res	numeric indicating the resolution of the plot, if set to NULL it will adapt the value according height of the plot
fontsize	character string with the default font or pointsize passed through to png function
units	character string with the units to use for plot width and height passed through to png function
rawout	character string with the name of the raw HTML file to generate (e.g. only plotting code) In case NULL no raw output will be generated. In order to combine results the filename should end in .raw.html
cleancur	logical indicating if the available plots should be deleted before creating new ones
...	additional arguments passed through to <a href="#">html_doc()</a> . Most important are template, rendlist, css and show

**Value**

The function returns a HTML file (or writes output to console)

## Examples

```
# It is convenient to have an object for the plot argument
## Not run:

data(Theoph)
library(ggplot2)
pl <- qplot(Time, conc, data=Theoph, facets=~Subject,geom="line")
html_plot(pl,out=paste0(tempfile(),".html"))

# Base plots work a bit different and can be placed
# in the function directly or wrapped in a function
html_plot(plot(conc~Time,data=Theoph),out=tempfile(fileext=".html"))
pl <- function() {
  plot(conc~Time,data=Theoph)
  title(main="a plot")
}
html_plot(pl(),out=tempfile(fileext=".html"))

## End(Not run)
```

---

html\_table

*Creates a HTML table*

---

## Description

This function creates a HTML table. The function calls a combination of functions within the R3port package to create an overall table and writes it to a file or console

## Usage

```
html_table(
  dfrm,
  x,
  y,
  var,
  fill = "",
  uselabel = TRUE,
  yhead = FALSE,
  footnote = NULL,
  title = "table",
  titlepr = NULL,
  xabove = FALSE,
  group = NULL,
  xrepeat = FALSE,
  tclass = "sample",
  out = NULL,
  rawout = paste0(out, ".rawhtml"),
  ...
)
```

**Arguments**

dfrm	the data frame for which the table should be generated
x	vector of x variable(s) in the data frame
y	vector of y variable(s) in the data frame (will be cast to generate long format)
var	variable within the data frame with the values to be placed in the table
fill	character vector of one indicating the character to use in case of missing values
uselabel	logical indicating if labels should be used for the x variable(s). If set to TRUE, the function will try to use the label attribute for the display of x variable(s).
yhead	logical indicating if the y variable should also be set as header in the table.
footnote	character string with the footnote to be placed in the footer of the page (HTML coding can be used for example to create line breaks)
title	character string to define the title of the table which will be added to the caption
titlepr	character string to define the prefix of the table title. Can be used to create custom table numbering
xabove	logical indicating if the first unique x variable should be placed in the table row above. Mostly used to save space on a page
group	number indicating which x variables should be grouped (displayed in table with a certain white space) and interpreted as x[1:group]
xrepeat	logical indicating if duplicate x values should be repeated in the table or not
tclass	character string with the table class. Can be used in combination with custom css
out	filename for the output HTML file (if NULL it will print to console)
rawout	character string with the name of the raw HTML file to generate (e.g. only table with) In case NULL no raw output will be generated. In order to combine results the filename should end in .raw.html
...	additional arguments passed through to <code>html_doc()</code> . Most important are template, rendlist, css and show

**Details**

The format of the current table/css was designed to look good in the browser and in a WORD document (when html is opened in WORD). Additional colspans are added in the table header which might lead to additional space in case a user defined css file is used.

**Value**

The function returns a HTML file (or writes output to console)

**Examples**

```
## Not run:
data(Indometh)
Indometh$id <- as.numeric(as.character(Indometh$Subject))
Indometh$strtr <- ifelse(Indometh$id<4,"trtr 1","trtr 2")
```

```

html_table(Indometh,x=c("trt", "time"),y="id",var="conc",
           out=tempfile(fileext=".html"),xabove=TRUE)

# Usage of multiple y values
html_table(Indometh,x="time",y=c("trt", "id"),var="conc",
           out=tempfile(fileext=".html"))

# Some examples for different options
html_table(Indometh,x=c("time", "trt"),y="id",var="conc",
           out=tempfile(fileext=".html"),yhead=TRUE,
           group=1,titlepr="TBL01",title="Dummy table",
           footnote="this table is not very informative")

## End(Not run)

```

---

html\_table\_design      *Designs a table based on a object returned by the table\_prep function*

---

## Description

This function designs the a HTML table based on the data frame list returned by the table\_prep function.

## Usage

```

html_table_design(
  dfl,
  uselabel = TRUE,
  yhead = FALSE,
  footnote = NULL,
  title = "table",
  titlepr = NULL,
  xabove = TRUE,
  group = NULL,
  xrepeat = FALSE,
  tclass = "sample"
)

```

## Arguments

dfl	list generated by the table_prep function which serves as the base of the table to be generated
uselabel	logical indicating if labels should be used for the x variable(s). If set to TRUE, the function will try to use the label attribute for the display of x variable(s).
yhead	logical indicating if the y variable should also be set as header in the table.
footnote	character string with the footnote to be placed in the footer of the page (HTML coding can be used for example to create line breaks)

title	character string to define the title of the table which will be added to the caption
titlepr	character string to define the prefix of the table title. Can be used to create custom table numbering
xabove	logical indicating if the first unique x variable should be placed in the table row above. Mostly used to save space on a page
group	number indicating which x variables should be grouped (displayed in table with a certain white space) and interpreted as x[1:group]
xrepeat	logical indicating if duplicate x values should be repeated in the table or not
tclass	character string with the table class. Can be used in combination with custom css

### Details

This function designs a HTML pivot table based on the results of the table\_prep output. This means that the function Should always be used in conjunction with this function.

### Value

The function returns a vector that defines the entire HTML table. This vector can be adapted manually however it is intended to be used in a print function to add to a HTML document.

### Examples

```
## Not run: html_table_design(lstobject)
```

---

ltx_combine	<i>Combines multiple latex files to a single tex and compiles document</i>
-------------	----------------------------------------------------------------------------

---

### Description

This function combines multiple latex files. This is done based on the name of the files and should end with raw.tex to make the function pick-up these files.

### Usage

```
ltx_combine(combine = ".", out = NULL, presentation = FALSE, clean = 0, ...)
```

### Arguments

combine	character string with the location of the raw tex files or list with file names within same directory
out	filename for the output latex file (if an empty string is provided it will print to console)
presentation	logical indicating if the output is a latex presentation (in this case the results will be placed within frames and without captions and clearpage)

clean	integer between 0 and 2 indicating if all individual files should be kept (0), all individual tex and raw tex files should be deleted (1) or all individual files should be deleted (2)
...	additional arguments passed through to <code>ltx_doc()</code> . Most important are template, rendlist, compile and show

### Details

Currently the generated output is saved in the same place where the separate tables and plots are located defined in the 'combine' argument. This is done even when a different file path is specified in 'out' (using the basename function). The reason is to not copy files linked within the document and preventing broken links. This behaviour might change in future releases

### Value

The function returns a latex file (or writes output to console)

### Examples

```
# Take into account the usage of tempfile() with multiple function calls
## Not run:
data(Theoph)
ltx_list(Theoph[1:11,],out=tempfile(fileext=".tex"),show=FALSE)
ltx_plot(plot(conc~Time,data=Theoph),out=tempfile(fileext=".tex"),show=FALSE)
ltx_combine(combine=tempdir(),out="rep1.tex")

# possibility for presentation layout (beamer template provided in package)
ltx_combine(combine=tempdir(),out="rep1.tex",
            template=paste0(system.file(package="R3port"),"/beamer.tex"),
            presentation=TRUE)

# Or other template with different orientation
ltx_combine(combine=tempdir(),out="rep1.tex",
            template=paste0(system.file(package="R3port"),"/listing.tex"),
            orientation="portrait")

## End(Not run)
```

---

ltx\_doc

---

*Prints latex code for a table, listing, plot, or text to a file or console*


---

### Description

This function makes a latex document using output generated with functions in the R3port package or any other latex code available as vector. Basically it adds a preamble to a tex file and let's the user select various options to customize the output.

**Usage**

```
ltx_doc(
  text,
  out = NULL,
  template = paste0(system.file(package = "R3port"), "/simple.tex"),
  rendlist,
  orientation = "landscape",
  rtitle = "report",
  compile = TRUE,
  show = TRUE
)
```

**Arguments**

text	character vector to be placed within latex document
out	filename for the output latex file (if NULL it will print to console)
template	file name of the template file to use (see examples how templates can be used/adapted)
rendlist	a render list to be used for the template file (see <a href="#">whisker::whisker.render</a> )
orientation	string indicating the page orientation (can be either "landscape" or "portrait")
rtitle	string indicating the title of the output document
compile	logical indicating if the tex file should be compiled (using <code>tools::texi2dvi</code> )
show	logical indicating if the resulting pdf file from the compiled tex file should be opened when created

**Value**

The function returns a latex file (or writes output to console)

**See Also**

[html\\_doc\(\)](#)

**Examples**

```
## Not run:
txt <- "\\section{example}"
tbl <- "\\begin{tabular}{|l|c|r|} 1 & 2 & 3 \\\\ 4 & 5 & 6 \\\\ \\end{tabular}"
add <- "\\\\ Including some additional text"
ltx_doc(c(txt, tbl, add), out=paste0(tempfile(), ".tex"), show=FALSE)

# You can use xtable (and any other packages that output tex)
library(xtable)
data(Theoph)
xtbl <- print(xtable(Theoph), tabular.environment="longtable", floating=FALSE, print.results=FALSE)
ltx_doc(xtbl, out=tempfile(fileext = ".tex"))

## End(Not run)
```

---

ltx_list	<i>Creates a latex listing</i>
----------	--------------------------------

---

## Description

This function creates a latex listing which can be written to a file or console

## Usage

```
ltx_list(
  dfrm,
  vars = names(dfrm),
  fill = "",
  vargroup = NULL,
  porder = TRUE,
  uselabel = TRUE,
  footnote = "",
  tablenote = "",
  mancol = NULL,
  size = "\\footnotesize",
  title = "listing",
  titlepr = NULL,
  group = NULL,
  xrepeat = FALSE,
  hyper = TRUE,
  out = NULL,
  rawout = paste0(out, ".rawtex"),
  convchar = TRUE,
  tabenv = "longtable",
  label = NULL,
  flt = "h",
  ...
)
```

## Arguments

dfrm	the data frame to be prepared
vars	character vector that defines the variables within the data frame to be placed in the listing
fill	<b>[Deprecated]</b> character vector of one indicating the character to use in case of missing values this is only applicable for <a href="#">ltx_table</a>
vargroup	a vector of the same length as vars. Creates a first line in the table to group variables (see details)
porder	logical indicating if the data frame should be ordered on the variables given in vars

uselabel	logical indicating if labels should be used for the x variable(s). If set to TRUE, the function will try to use the label attribute for the display of x variable(s).
footnote	character string with the footnote to be placed in the footer of the page (LaTeX coding can be used for example to create line breaks)
tablenote	character string with the table note to be placed directly below the table (LaTeX coding can be used for example to create line breaks)
mancol	character string to define manual column alignment. in case argument is NULL, a sensible default will be set.
size	character string to define the font size of the table
title	character string to define the title of the table which will be added to the caption
titlepr	character string to define the prefix of the table title. Can be used to create custom table numbering
group	number indicating which x variables should be grouped (displayed in table with a certain white space) and interpreted as $x[1:group]$
xrepeat	logical indicating if duplicate x values should be repeated in the table or not
hyper	logical indicating if a hypertarget should be set used for bookmarks
out	filename for the output latex file (if NULL it will print to console)
rawout	character string with the name of the raw latex file to generate (e.g. only listing with no preamble and document ending) In case NULL no raw output will be generated. In order to combine results the filename should end in .rawtex
convchar	logical indicating if special characters should be masked
tabenv	character with the table environment to use. Currently "longtable" and "tabular" are supported
label	character with the label to add after the caption for referencing the table in text
flt	character with the type of floating environment to use (only applicable for tabular environment)
...	additional arguments passed through to <code>ltx_doc()</code> . Most important are template, rendlist, compile and show

### Details

The vargroup argument should be provided in the following form:  
`c(rep("", 4), rep("group1", 3), rep("group2", 4))`. The function will place the text within the vector with the given length as first line in the table with a midrule below it. an exception is made for empty strings.

### Value

The function returns a latex file (or writes output to console)

**Examples**

```
## Not run:
data(Theoph)
grp <- c(rep("",3),rep("grouped variables",2))
ltx_list(Theoph,out=tempfile(fileext=".tex"),vargroup=grp,
         template=paste0(system.file(package="R3port"),"/listing.tex"))

## End(Not run)
```

---

`ltx_plot`*Prints a R plot to a latex file or console*

---

**Description**

This function makes a latex document including the plots defined

**Usage**

```
ltx_plot(
  plot,
  out,
  title = "plot",
  titlepr = NULL,
  footnote = "",
  plotnote = "",
  lwidth = NULL,
  pwidth = 10,
  pheight = 5.5,
  res = NULL,
  hyper = TRUE,
  outfmt = "pdf",
  fontsize = 12,
  units = "px",
  rawout = paste0(out, ".rawtex"),
  linebreak = TRUE,
  label = NULL,
  captpl = "top",
  rotate = FALSE,
  cleancur = FALSE,
  titlesub = NULL,
  shorttitle = NULL,
  ...
)
```

**Arguments**

plot	plot object or function call that creates plot to be printed to file
out	filename for the output latex file
title	character string to define the title of the plot which will be added to the caption
titlepr	character string to define the prefix of the title. Can be used to create custom numbering
footnote	character string with the footnote to be placed in the footer of the page (LaTeX coding can be used for example to create line breaks)
plotnote	character string with the plot note to be placed directly below the plot (LaTeX coding can be used for example to create line breaks)
lwidth	character string indicating the width of the plot within latex (e.g. "\linewidth")
pwidth	numeric indicating the width of the plot to be generated in inches or pixels (for respectively the extensions pdf and png)
pheight	numeric indicating the height of the plot to be generated in inches or pixels (for respectively the extensions pdf and png)
res	numeric indicating the resolution of the plot (in case png is used), if set to NULL it will adapt the value according height of the plot
hyper	logical indicating if a hypertarget should be set used for bookmarks
outfmt	character string indicating the format of the output file (currently "pdf" and "png" are accepted)
fontsize	character string with the default font or pointsize passed through to png or pdf function
units	character string with the units to use for plot width and height passed through to png function
rawout	character string with the name of the raw latex file to generate (e.g. only plot code with no preamble and document ending) In case NULL no raw output will be generated. In order to combine results the filename should end in .rawtex
linebreak	logical indicating if a linebreak (clearpage) should be given after a plot
label	character with the label to add after the caption for referencing the table in text
captpl	character with the caption placement, can be either "top" or "bottom"
rotate	logical indicating if the resulting figure should be rotated 90 degrees clockwise
cleancur	logical indicating if the available plots should be deleted before creating new ones
titlesub	character string to define the subtext after title in footnotesize
shorttitle	character string to define the title in a shorter version of the plot which will be used in the TOF in a latex document
...	additional arguments passed through to <code>ltx_doc()</code> . Most important are <code>template</code> , <code>rendlist</code> , <code>compile</code> and <code>show</code>

**Value**

The function returns a latex file (or writes output to console)

**Examples**

```

# It is convenient to have an object for the plot argument
## Not run:
data(Theoph)
library(ggplot2)
pl <- ggplot(Theoph,aes(Time,conc)) + geom_line() + facet_wrap(~Subject)
ltx_plot(pl,out=tempfile(fileext=".tex"))

# Base plots work a bit different and can be placed
# in the function directly or wrapped in a function
pl <- function() {
  plot(conc~Time,data=Theoph)
  title(main="a plot")
}
ltx_plot(pl(),out=tempfile(fileext=".tex"))
# In case of big data it can be more convenient to have a png included
ltx_plot(plot(rnorm(1e6)),out=tempfile(fileext=".tex"),
         outfmt="png",pwidth=2000,pheight=1200)

## End(Not run)

```

---

ltx\_table

*Creates a latex table*


---

**Description**

This function creates a latex table. The function calls a combination of functions within the R3port package to create an overall table and writes it to a file or console

**Usage**

```

ltx_table(
  dfrm,
  x,
  y,
  var,
  fill = "",
  uselabel = TRUE,
  yhead = FALSE,
  footnote = "",
  tablenote = "",
  mancol = NULL,
  size = "\\footnotesize",
  title = "table",
  titlepr = NULL,
  xabove = FALSE,
  group = NULL,
  xrepeat = FALSE,

```

```

    hyper = TRUE,
    out = NULL,
    rawout = paste0(out, ".rawtex"),
    convchar = TRUE,
    tabenv = "longtable",
    label = NULL,
    flt = "h",
    ...
)

```

### Arguments

dfrm	the data frame for which the table should be generated
x	vector of x variable(s) in the data frame
y	vector of y variable(s) in the data frame (will be cast to generate long format)
var	variable within the data frame with the values to be placed in the table
fill	character vector of one indicating the character to use in case of missing values
uselabel	logical indicating if labels should be used for the x variable(s). If set to TRUE, the function will try to use the label attribute for the display of x variable(s).
yhead	logical indicating if the y variable should also be set as header in the table.
footnote	character string with the footnote to be placed in the footer of the page (LaTeX coding can be used for example to create line breaks)
tablenote	character string with the table note to be placed directly below the table (LaTeX coding can be used for example to create line breaks)
mancol	character string to define manual column alignment. in case argument is NULL, a sensible default will be set.
size	character string to define the font size of the table
title	character string to define the title of the table which will be added to the caption
titlepr	character string to define the prefix of the table title. Can be used to create custom table numbering
xabove	logical indicating if the first unique x variable should be placed in the table row above. Mostly used to save space on a page
group	number indicating which x variables should be grouped (displayed in table with a certain white space) and interpreted as x[1:group]
xrepeat	logical indicating if duplicate x values should be repeated in the table or not
hyper	logical indicating if a hypertarget should be set used for bookmarks
out	filename for the output latex file (if NULL it will print to console)
rawout	character string with the name of the raw latex file to generate (e.g. only table with no preamble and document ending) In case NULL no raw output will be generated. In order to combine results the filename should end in .rawtex
convchar	logical indicating if special characters should be masked
tabenv	character with the table environment to use. Currently "longtable" and "tabular" are supported

label	character with the label to add after the caption for referencing the table in text
flt	character with the type of floating environment to use (only applicable for tabular environment)
...	additional arguments passed through to <code>ltx_doc()</code> . Most important are template, rendlist, compile and show

### Value

The function returns a latex file (or writes output to console)

### Examples

```
## Not run:
data(Indometh)
Indometh$id <- as.numeric(as.character(Indometh$Subject))
Indometh$trt <- ifelse(Indometh$id<4,"trt 1","trt 2")

ltx_table(Indometh,x=c("trt","time"),y="id",var="conc",
          out=tempfile(fileext=".tex"),xabove=TRUE)

# Usage of multiple y values
ltx_table(Indometh,x="time",y=c("trt","id"),var="conc",
          out=tempfile(fileext=".tex"))

# Some examples for different options
ltx_table(Indometh,x=c("time","trt"),y="id",var="conc",
          out=tempfile(fileext=".tex"),yhead=TRUE,
          group=1,titlepr="TBL01",title="Dummy table",
          footnote="this table is not very informative")

## End(Not run)
```

---

ltx\_table\_design

*Designs a table based on a object returned by the table\_prep function*


---

### Description

This function designs the a latex table based on the data frame list returned by the table\_prep function.

### Usage

```
ltx_table_design(
  dfl,
  uselabel = TRUE,
  yhead = FALSE,
  footnote = "",
  tablenote = "",
```

```

mancol = NULL,
size = "\\normalsize",
title = "table",
titlepr = NULL,
xabove = TRUE,
group = NULL,
xrepeat = FALSE,
hyper = TRUE,
tabenv = "longtable",
label = NULL,
flt = "h"
)

```

### Arguments

<code>df1</code>	list generated by the <code>table_prep</code> function which serves as the base of the table to be generated
<code>uselabel</code>	logical indicating if labels should be used for the x variable(s). If set to <code>TRUE</code> , the function will try to use the label attribute for the display of x variable(s).
<code>yhead</code>	logical indicating if the y variable should also be set as header in the table.
<code>footnote</code>	character string with the footnote to be placed in the footer of the page (LaTeX coding can be used for example to create line breaks)
<code>tablenote</code>	character string with the table note to be placed directly below the table (LaTeX coding can be used for example to create line breaks)
<code>mancol</code>	character string to define manual column alignment. in case argument is <code>NULL</code> , a sensible default will be set.
<code>size</code>	character string to define the font size of the table
<code>title</code>	character string to define the title of the table which will be added to the caption
<code>titlepr</code>	character string to define the prefix of the table title. Can be used to create custom table numbering
<code>xabove</code>	logical indicating if the first unique x variable should be placed in the table row above. Mostly used to save space on a page
<code>group</code>	number indicating which x variables should be grouped (displayed in table with a certain white space) and interpreted as <code>x[1:group]</code>
<code>xrepeat</code>	logical indicating if duplicate x values should be repeated in the table or not
<code>hyper</code>	logical indicating if a <code>hypertarget</code> should be set used for bookmarks
<code>tabenv</code>	character with the table environment to use. Currently "longtable" and "tabular" are supported
<code>label</code>	character with the label to add after the caption for referencing the table in text
<code>flt</code>	character with the type of floating environment to use (only applicable for tabular environment)

### Details

This function designs a latex pivot table based on the results of the `table_prep` output. This means that the function Should always be used in conjunction with this function.

**Value**

The function returns a vector that defines the entire latex table. This vector can be adapted manually however it is intended to be used in a print function to add to a latex document.

**Examples**

```
## Not run: ltx_table_design(lstobject)
```

---

means	<i>Calculate summary statistics on a data frame</i>
-------	-----------------------------------------------------

---

**Description**

This function calculates summary statistics on a data frame. It is intended to use to calculate a standard number of descriptive statistics which are defined in the packs argument

**Usage**

```
means(dfrm, variable, by, total = NULL, pack = 1, dig = 2, alpha = 0.1)
```

**Arguments**

dfrm	the data frame to calculate the statistics on
variable	character with the variable within the data frame for which the statistics should be calculated
by	character vector of variable(s) within the data frame to stratify the statistics on
total	character vector of variable(s) within the data frame to calculate totals on
pack	numeric indicating number of the pack or name of the function with different descriptive statistics (see details)
dig	the number of digits to use for output statistics (except for N which is always displayed as integer)
alpha	the alpha in case the statistics pack calculates confidence limits

**Details**

The function calculates multiple statistics of a data frame that can be stratified. Furthermore, the statistics for a total can also be generated. Currently the function has the possibility to calculate 3 different sets of statistics which are commonly used within the field of clinical data analysis:

1. N, Mean, Median, SD, Min, Max
2. Ntot, N, Nmiss, Mean, Median, SD, Min, Max, CLM (different N values are given to identify the number of missing values)
3. N, Mean (SD), Median, Range (Both Mean (SD) and Range are concatenated to generate a dense overview of statistics)

All statistics are calculated on using `na.rm` set to `TRUE`, meaning that NA values are removed before calculating the statistics. A predefined set of statistics is chosen and not the possibility to implement user defined descriptive statistics. The reason for this is that there are many other (more simple) options to do this within R.

### Value

The function returns a dataframe with calculated descriptive statistics

### Examples

```
data(Indometh)
means(Indometh,"conc","time",total="time",pack=3)
```

---

R3port

*Report functions for HTML and PDF files*

---

### Description

The R3port package is written to easily create pdf and html documents including tables, listings and plots.

### Details

The package makes use of the option system within R. The way it was implemented is to always use an option in case it was set in `options()`. If an option is `NULL`, the function will evaluate the argument. The following options can be set within the package:

- `show` (see `ltx_doc()`, `html_doc()`)
- `template` (see `ltx_doc()`, `html_doc()`)
- `css` (see `html_doc()`)
- `compile` (see `ltx_doc()`)
- `orientation` (see `ltx_doc()`)
- `pwidth` (see `ltx_plot()`, `html_plot()`)
- `pheight` (see `ltx_plot()`, `html_plot()`)

### Author(s)

**Maintainer:** Richard Hooijmaijers <[richardhooijmaijers@gmail.com](mailto:richardhooijmaijers@gmail.com)>

Other contributors:

- Richard Hooijmaijers [copyright holder]

### See Also

Useful links:

- <https://github.com/RichardHooijmaijers/R3port>

---

table_prep	<i>Prepares the data for pivotal tabulation</i>
------------	-------------------------------------------------

---

### Description

This function prepares the data pivotal for tabulation. It is intended to use as a first step in the creation of a report pivotal table. It is not likely that this function will be used as stand-alone

### Usage

```
table_prep(dfrm, x, y, var, fill = "", type = "latex", convchar = TRUE)
```

### Arguments

dfrm	the data frame for which the table should be generated
x	vector of x variable(s) in the data frame
y	vector of y variable(s) in the data frame (will be cast to generate long format)
var	variable within the data frame with the values to be placed in the table
fill	character vector of one indicating the character to use in case of missing values
type	character vector of one indicating the type of table to generate, see details
convchar	logical indicating if special characters should be masked

### Details

This function only prepares the data for pivotal tabulation. The base of the function is the dcast function from plyr to reshape the data and the generation of a table header object to indicate how headers should be placed within the table. The function also specifies a type argument which includes "latex" or "html", this is necessary to identify specific character handling (convchar) that differs between HTML and LaTeX.

### Value

The function returns a list with the original dataframe, the table header, table data and other table specifications

### Examples

```
data(Indometh)
table_prep(Indometh, "time", "Subject", "conc")
```

# Index

`freq`, [2](#)

`html_combine`, [3](#)

`html_doc`, [4](#)

`html_doc()`, [4](#), [7](#), [8](#), [10](#), [14](#), [24](#)

`html_list`, [6](#)

`html_plot`, [7](#)

`html_plot()`, [24](#)

`html_table`, [6](#), [9](#)

`html_table_design`, [11](#)

`ltx_combine`, [12](#)

`ltx_doc`, [13](#)

`ltx_doc()`, [5](#), [13](#), [16](#), [18](#), [21](#), [24](#)

`ltx_list`, [15](#)

`ltx_plot`, [17](#)

`ltx_plot()`, [24](#)

`ltx_table`, [15](#), [19](#)

`ltx_table_design`, [21](#)

`means`, [23](#)

`R3port`, [24](#)

`R3port-package (R3port)`, [24](#)

`table_prep`, [25](#)

`whisker::whisker.render`, [5](#), [14](#)