

Package ‘crosstable’

March 8, 2021

Title Crosstables for Descriptive Analyses

Version 0.2.1

Description Create descriptive tables for continuous and categorical variables.

Apply summary statistics and counting function, with or without a grouping variable, and create beautiful reports using 'rmarkdown' or 'officer'.

You can also compute statistical tests and effect sizes if needed.

License GPL-3

URL <https://danchaltiel.github.io/crosstable/>,

<https://github.com/DanChaltiel/crosstable/>

BugReports <https://github.com/DanChaltiel/crosstable/issues/>

Depends R (>= 3.1.0)

Imports checkmate, dplyr (>= 1.0.0), ellipsis, flextable (>= 0.5.8),
forcats, glue, lifecycle, nortest, officer, purrr, rlang (>=
0.4.7), stats, stringr, survival, tibble, tidyr, tidyselect

Suggests covr, crayon, xml2, DescTools, digest, gt, expss, ggplot2,
gmodels, Hmisc, jsonlite, knitr, openxlsx, rmarkdown, sloop,
stringi, systemfonts, testthat, withr, waldo,

VignetteBuilder knitr

RdMacros lifecycle

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

Config/testthat/edition 3

Config/testthat/parallel true

Config/testthat/start-first watcher, parallel*

NeedsCompilation no

Author Dan Chaltiel [aut, cre] (<<https://orcid.org/0000-0003-3488-779X>>),
David Hajage [ccp]

Maintainer Dan Chaltiel <dan.chaltiel@gmail.com>

Repository CRAN

Date/Publication 2021-03-08 09:20:02 UTC

R topics documented:

apply_labels	3
as_gt.crosstable	3
as_workbook	5
body_add_crosstable	6
body_add_crosstable_footnote	7
body_add_gg2	7
body_add_img2	8
body_add_list	9
body_add_normal	10
body_add_table_legend	11
body_add_title	13
compact	14
confint_numeric	15
crosstable	16
crosstable_effect_args	18
crosstable_options	19
crosstable_test_args	20
cross_summary	21
display_effect	22
display_test	22
docx_bookmarks2	23
effect_summary	23
effect_survival	24
effect_tabular	25
format_fixed	26
get_label	27
import_labels	28
iris2	29
mtcars2	30
N	31
na	32
peek	32
plim	33
remove_labels	33
rename_dataframe_with_labels	34
set_label	35
summaryFunctions	36
test_correlation_auto	37
test_summarize_auto	38
test_summarize_linear_contrasts	39
test_survival_logrank	39
test_tabular_auto	40
write_and_open	41

apply_labels	<i>Batch set variable labels</i>
--------------	----------------------------------

Description

This function is a copycat of from expss package v0.10.7 (slightly modified) to avoid having to depend on expss. See [expss::apply_labels\(\)](#) for more documentation. Note that this version is not compatible with `data.table`.

Usage

```
apply_labels(data, ..., warn_missing = FALSE)
```

Arguments

data	data.frame/list
...	named arguments
warn_missing	if TRUE, throw a warning if some names are missing

Value

An object of the same type as `.data`, with labels

Examples

```
library(crosstable)
iris %>%
  apply_labels(Sepal.Length="Length of Sepal",
              Sepal.Width="Width of Sepal") %>%
  crosstable()
```

as_gt.crosstable	<i>Converts a crosstable object into a formatted gt table.</i>
------------------	--

Description

Converts a crosstable object into a formatted gt table.

Method to convert an object to a gt table

Default method to convert an object to a gt table

Usage

```
## S3 method for class 'crosstable'
as_gt(
  x,
  show_test_name = TRUE,
  by_header = NULL,
  keep_id = FALSE,
  generic_labels = list(id = ".id", variable = "variable", value = "value", total =
    "Total", label = "label", test = "test", effect = "effect"),
  ...
)

as_gt(x, ...)
```

Default S3 method:
as_gt(x, ...)

Arguments

x	object to be converted
show_test_name	in the test column, show the test name
by_header	a string to override the by header
keep_id	whether to keep the .id column
generic_labels	names of the crosstable default columns
...	arguments for custom methods

Value

a formatted gt table

Methods (by class)

- crosstable: For crosstables
- default: default function

Author(s)

Dan Chaltiel

See Also

[as_flextable.crosstable\(\)](#)
[gt::gt\(\)](#)

Examples

```
xx = mtcars2 %>% dplyr::select(1:9)
crosstable(xx) %>% as_gt
crosstable(xx, by=am) %>% as_gt
crosstable(xx, by=am, test=TRUE, total=TRUE, effect=TRUE) %>% as_gt(keep_id=TRUE)
```

as_workbook	<i>Converts a crosstable object into a formatted, savable openxlsx workbook.</i>
-------------	--

Description

Converts a crosstable object into a formatted, savable openxlsx workbook.

Usage

```
as_workbook(
  x,
  show_test_name = TRUE,
  by_header = NULL,
  keep_id = FALSE,
  generic_labels = list(id = ".id", variable = "variable", value = "value", total =
    "Total", label = "label", test = "test", effect = "effect"),
  ...
)
```

Arguments

x	the result of <code>crosstable()</code>
show_test_name	in the test column, show the test name
by_header	a string to override the by header
keep_id	whether to keep the .id column
generic_labels	names of the crosstable default columns
...	unused

Value

an openxlsx workbook containing the crosstable

Author(s)

Dan Chaitiel

Examples

```
library(openxlsx)
target = tempfile(fileext=".xlsx")
x=crosstable(mtcars2, c(mpg, vs, gear), total=TRUE, test=TRUE)
x %>%
  as_workbook(keep_id=TRUE) %>%
  saveWorkbook(file=target, overwrite = TRUE)
if(interactive()) browseURL(target)
```

body_add_crosstable *Add a crosstable to an officer document*

Description

`body_add_crosstable()` adds such a flextable an officer document.

Usage

```
body_add_crosstable(
  doc,
  x,
  body_fontsize = NULL,
  header_fontsize = ceiling(body_fontsize * 1.2),
  ...
)
```

Arguments

doc	a rdocx object, created by <code>officer::read_docx()</code>
x	a crosstable object
body_fontsize	fontsize of the body
header_fontsize	fontsize of the header
...	further arguments passed to <code>as_flextable.crosstable()</code>

Value

The docx object doc

Examples

```
#Officer
library(officer)
mytable = crosstable(mtcars2)
doc = read_docx() %>%
  body_add_crosstable(mytable) %>%
  body_add_break %>%
```

```

body_add_crosstable(mytable, compact=TRUE)

dfile = tempfile(fileext=".docx")
print(doc, target = dfile)
if(interactive()) browseURL(dfile)

```

body_add_crosstable_footnote

Adds a standard footnote explaining the abbreviations used in a crosstable

Description

Use it below [body_add_crosstable\(\)](#). Footnote: Med: median, IQR: interquartile range, Std: standard deviation. Percentages are expressed in column.

Usage

```
body_add_crosstable_footnote(doc)
```

Arguments

doc a rdocx object

Value

The docx object doc

body_add_gg2 *Alternative to [officer::body_add_gg\(\)](#) which uses ggplot syntax*

Description

Alternative to [officer::body_add_gg\(\)](#) which uses ggplot syntax

Usage

```

body_add_gg2(
  doc,
  value,
  width = 6,
  height = 5,
  units = getOption("crosstable_units", "in"),
  style = getOption("crosstable_style_image", doc$default_styles$paragraph),
  res = 300,
  ...
)

```

Arguments

doc	an rdocx object
value	ggplot object
width, height	width and height. Can be abbreviated to w and h.
units	units for width and height
style	paragraph style
res	resolution of the png image in ppi (passed to the argument dpi of <code>ggplot2::ggsave()</code>)
...	other arguments to be passed to <code>ggplot2::ggsave()</code>

Value

The docx object doc

Examples

```
if(require("ggplot2") && capabilities(what = "png")){
  library(officer)
  p = ggplot(data = iris ) +
    geom_point(mapping = aes(Sepal.Length, Petal.Length))

  options(crosstable_units="cm")
  options(crosstable_style_image="centered")
  doc = read_docx() %>%
    body_add_normal("Text before") %>%
    body_add_gg2(p, w=14, h=10, scale=1.5) %>% #or units="cm" instead of using options
    body_add_normal("Text after")
  #write_and_open(doc)
}
```

body_add_img2

Alternative to `officer::body_add_img()` which adds a units choice

Description

Alternative to `officer::body_add_img()` which adds a units choice

Usage

```
body_add_img2(
  doc,
  src,
  width,
  height,
  units = getOption("crosstable_units", "in"),
  ...
)
```


Arguments

doc	an rdocx object
src	image filename, the basename of the file must not contain any blank.
width, height	width and height. Can be abbreviated to w and h.
units	units for width and height
...	other arguments to be passed to <code>officer::body_add_img()</code>

Value

The docx object doc

See Also

[body_add_gg2\(\)](#)

Examples

```
img.file = file.path( R.home("doc"), "html", "logo.jpg" )
if(file.exists(img.file)){
  library(officer)
  options(crosstable_units="cm")
  doc = read_docx() %>%
    body_add_normal("This is the R logo.") %>%
    body_add_img2(img.file, h=7.6, w=10, style="centered") #or units="cm" without options
  #write_and_open(doc)
}
```

body_add_list	<i>Add a list to an officer document</i>
---------------	--

Description

Add a list to an officer document

Usage

```
body_add_list(doc, value, ordered = FALSE, style = NULL, ...)
```

```
body_add_list_item(doc, value, ordered = FALSE, style = NULL, ...)
```

Arguments

doc	a docx object
value	a character (<code>body_add_list()</code>) or a string (<code>body_add_list_item</code>)
ordered	if TRUE, adds an ordered list, if FALSE, adds a bullet list

style specify the style manually, overriding ordered. A better way is to set options `crosstable_style_list_ordered` and `crosstable_style_list_unordered` globally.

... passed on to `officer::body_add_par()`

Details

Ordered lists and bullet lists are not supported by the default officer template (see <https://github.com/davidgohel/officer/issues>). You have to manually set custom styles matching those list in a custom Word template file. Then, you can use either the `style` argument or `crosstable` options. See examples for more details.

Value

The docx object `doc`

Author(s)

Dan Chahiel

Examples

```
## Not run:
#For this example to work, `my_template.docx` should include styles named
#`ordered_list` and `unordered_list`

library(officer)
library(crosstable)
options(crosstable_style_list_ordered="ordered_list")
options(crosstable_style_list_unordered="unordered_list")

read_docx("my_template.docx") %>%
  body_add_list(c("Numbered item 1", "Numbered item 2"), ordered = TRUE) %>%
  body_add_list_item("Numbered item 3", ordered = TRUE) %>%
  body_add_list(c("Bullet item 1", "Bullet item 2"), ordered = FALSE) %>%
  body_add_list_item("Bullet item 3", ordered = FALSE) %>%
  write_and_open()

## End(Not run)
```

body_add_normal

Add a new paragraph with default style

Description

Add a new paragraph in an officer document with default style. Variables can be inserted as multiple strings (`paste()` style) or enclosed by braces (`glue()` style). References to any bookmark can be inserted using the syntax "`@ref(bookmark)`". See an example in [body_add_table_legend\(\)](#).

Usage

```
body_add_normal(doc, ..., .sep = "")
```

Arguments

doc	the doc object (created with the read_docx function of officer package)
...	one or several character strings, pasted using .sep. As with glue::glue(), expressions enclosed by braces will be evaluated as R code. If more than one variable is passed, all should be of length 1.
.sep	Separator used to separate elements.

Value

a new doc object
The docx object doc

Author(s)

Dan Chaltiel

Examples

```
library(officer)
library(crosstable)

info_rows = c("Also, table iris has {nrow(iris)} rows.",
              "And table mtcars has {nrow(mtcars)} rows.")
doc = read_docx() %>%
  body_add_normal("Table iris has", ncol(iris), "columns.", .sep=" ") %>% #paste style
  body_add_normal("However, table mtcars has {ncol(mtcars)} columns") %>% #glue style
  body_add_normal(info_rows) #vector style
#write_and_open(doc)
```

body_add_table_legend *Add a table legend to an officer document*

Description

Add a table legend to an officer document

Usage

```
body_add_table_legend(
  doc,
  legend,
  bookmark = NULL,
  legend_style = getOption("crosstable_style_legend", "Table Caption"),
```

```

    style = getOption("crosstable_style_strong", "strong"),
    legend_name = "Table",
    seqfield = "SEQ Table \\* Arabic"
)

body_add_figure_legend(
  doc,
  legend,
  bookmark = NULL,
  legend_style = getOption("crosstable_style_legend", "Image Caption"),
  style = getOption("crosstable_style_strong", "strong"),
  legend_name = "Figure",
  seqfield = "SEQ Figure \\* Arabic"
)

```

Arguments

doc	a docx object
legend	the table legend. As with <code>glue::glue()</code> , expressions enclosed by braces will be evaluated as R code.
bookmark	the id of the bookmark. This is the id that should then be called in <code>body_add_normal()</code> using "\\@ref(id)".
legend_style	style of of the whole legend. May depend on the docx template
style	style of the number. May depend on the docx template (default to strong)
legend_name	name before the numbering. Useful for translation
seqfield	Keep default. Otherwise, you may figure it out doing this: in a docx file, insert a table legend, right click on the inserted number and select "Toggle Field Codes". This argument should be the value of the field, with extra escaping.

Value

The docx object doc

Warning

At first, the legends added with `body_add_table_legend()` or `body_add_figure_legend()` have no numbers. You have to manually update the references in MS Word: select all (Ctrl+A), then update (F9). You might have to do this several times. More info on <https://ardata-fr.github.io/officeverse/faq.html#update-fields>.

Author(s)

Dan Chaltiel

Examples

```

library(officer)
p=ggplot2::quickplot(x=Sepal.Length, y=Sepal.Width, color=Species, data=iris)
x=read_docx() %>%
  body_add_normal("As you can see in Table \@ref(tab1) and in Figure \@ref(fig1), ",
                 "the iris dataset is about flowers.") %>%
  body_add_normal() %>%
  body_add_table_legend("Iris dataset", bookmark="tab1") %>%
  body_add_crosstable(crosstable(iris)) %>%
  body_add_gg(p) %>%
  body_add_figure_legend("Iris plot", bookmark="fig1")
write_and_open(x)
#press Ctrl+A then F9 twice for the reference to appear.

```

body_add_title	<i>Add a title to an officer document</i>
----------------	---

Description

Add a title to an officer document

Usage

```

body_add_title(
  doc,
  value,
  level = 1,
  style = getOption("crosstable_style_heading", "heading")
)

```

Arguments

doc	the doc object (created with the read_docx function of officer package)
value	a character string
level	the level of the title. See styles_info(doc) to know the possibilities.
style	the name of the title style. See styles_info(doc) to know the possibilities.

Value

The docx object doc

Author(s)

Dan Chaltiel

Examples

```

library(officer)
library(crosstable)
library(dplyr)
doc = read_docx()
doc = doc %>%
  body_add_title("La table iris", 1) %>%
  body_add_title("Description", 2) %>%
  body_add_normal("La table iris a ", ncol(iris), " colonnes.")
#write_and_open(doc)

```

compact

*Generic function to compact a table (publication formatting)***Description**

Generic function to compact a table (publication formatting)

Usage

```

## S3 method for class 'data.frame'
compact(
  data,
  name_from,
  name_to = "variable",
  wrap_cols = NULL,
  rtn_flextable = FALSE,
  ...
)

## S3 method for class 'crosstable'
compact(data, name_from = c("label", ".id"), name_to = "variable", ...)

```

Arguments

<code>data</code>	the object to compact
<code>name_from</code>	name of the column to be collapsed when compacting
<code>name_to</code>	name of the column that will receive the collapsed column. Will be created if it doesn't exist.
<code>wrap_cols</code>	name of the columns to wrap
<code>rtn_flextable</code>	whether to return a formatted <code>flextable()</code> object or a simple <code>data.frame</code>
<code>...</code>	additional arguments (not used)

Value

a compacted `data.frame`

Examples

```
#dataframes
x=iris[c(1:5,51:55,101:105),]
compact(x, name_from="Species")
compact(x, name_from="Species", name_to="Petal.Length")

#crosstables
x=crosstable(mtcars2, c(displ, hp, am), by=vs, test=TRUE, effect=TRUE)
compact(x)
compact(x, name_from=".id")
```

confint_numeric	<i>Confidence interval of a numeric vector</i>
-----------------	--

Description

Not an S3 method, which might have conflicted with [stats::confint](#).

Usage

```
confint_numeric(object, level = 0.95, B = 0)
```

Arguments

object	a vector, numeric or equivalent (date, logical...)
level	the confidence level required
B	if >0, the number of bootstraps

Value

the vector [conf_inf, conf_sup]

Examples

```
confint_numeric(iris$Sepal.Length)
confint_numeric(mtcars2$hp_date)
confint_numeric(mtcars2$hp_date, level=0.99)
```

 crosstable

Easily describe datasets

Description

Generate a descriptive table of all chosen columns, as contingency tables for categorical variables and as calculation summaries for numeric variables. If the `by` argument points to a categorical variable, `crosstable` will output a description of all columns for every level. Else, if it points to a numeric variable, `crosstable` will calculate correlation coefficients with all other selected numeric columns. Finally, if it points to a `Surv` object, `crosstable` will describe the survival at different times.

Usage

```
crosstable(
  data,
  cols = NULL,
  ...,
  by = NULL,
  total = c("none", "row", "column", "both"),
  margin = c("row", "column", "cell", "none", "all"),
  percent_digits = 2,
  showNA = c("ifany", "always", "no"),
  label = TRUE,
  funs = c(` ` = cross_summary),
  funs_arg = list(),
  cor_method = c("pearson", "kendall", "spearman"),
  test = FALSE,
  test_args = crosstable_test_args(),
  unique_numeric = 3,
  date_format = NULL,
  effect = FALSE,
  effect_args = crosstable_effect_args(),
  times = NULL,
  followup = FALSE,
  .vars
)
```

Arguments

<code>data</code>	a <code>data.frame</code>
<code>cols</code>	the variables to describe. Can be a character or name vector, a <code>tidyselect</code> helper, a (lambda) function that returns a logical, or a formula. See examples or <code>vignette("crosstable-selecti")</code> for more details.
<code>...</code>	more variables to describe. Cannot be a lambda function nor a formula.
<code>by</code>	the variable to group on. Character or name.

<code>total</code>	one of ["none", "row", "column" or "both"] to indicate whether to add total rows and/or columns. Default to none.
<code>margin</code>	one of ["row", "column", "cell", "none" or "all"] to indicate which proportions should be computed in frequency tables. Default to row.
<code>percent_digits</code>	number of digits for percentages
<code>showNA</code>	whether to show NA in categorical variables (one of c("ifany", "always", "no"), like in <code>table()</code>)
<code>label</code>	whether to show labels. See import_labels or set_label for how to add labels to the dataset columns.
<code>funs</code>	functions to apply to numeric variables. Default to cross_summary .
<code>funs_arg</code>	additional parameters for funs, e.g. <code>digits</code> (the number of decimal places) for the default cross_summary . Ultimately, these arguments are passed to format_fixed .
<code>cor_method</code>	one of ["pearson", "kendall", or "spearman"] to indicate which correlation coefficient is to be used.
<code>test</code>	whether to perform tests
<code>test_args</code>	See crosstable_test_args to override default testing behaviour.
<code>unique_numeric</code>	the number of non-missing different levels a variable should have to be considered as numeric
<code>date_format</code>	if <code>x</code> is a vector of Date or POSIXt, the format to apply (see strptime for formats)
<code>effect</code>	whether to compute a effect measure
<code>effect_args</code>	See crosstable_effect_args to override default behaviour.
<code>times</code>	when using formula with survival::Surv() objects, which times to summarize
<code>followup</code>	when using formula with survival::Surv() objects, whether to display follow-up time
<code>.vars</code>	deprecated

Details

Can be formatted as an HTML table using [as_flextable\(\)](#).

Value

A data.frame of class `crosstable`

See Also

<https://danchaltiel.github.io/crosstable/>, [as_flextable](#), [import_labels](#)

Examples

```

#whole table
crosstable(iris)
crosstable(mtcars)
crosstable(mtcars2)

#tidyselection, custom functions
library(dplyr)
crosstable(mtcars2, ends_with("t"), starts_with("c"), by=vs,
           funs=c(mean, quantile), funs_arg = list(probs=c(.25,.75)))

#margin and totals
crosstable(mtcars2, disp, vs, by=am,
           margin=c("row", "col"), total = "both")

#predicate selection, correlation, testing
crosstable(mtcars2, where(is.numeric), by=hp, test=TRUE)

#lambda selection, effect calculation
crosstable(mtcars2, ~is.numeric(.x) && mean(.x)>50, by=vs, effect=TRUE)

#Dates
mtcars2$my_date = as.Date(mtcars2$hp , origin="2010-01-01") %>% set_label("Some nonsense date")
crosstable(mtcars2, my_date, by=vs, date_format="%d/%m/%Y")

#Survival data (using formula UI)
library(survival)
crosstable(am1, Surv(time, status) ~ x, times=c(0,15,30,150), followup=TRUE)

```

crosstable_effect_args

Default arguments for calculating and displaying effects in [crosstable\(\)](#)

Description

This helper function provides default parameters for defining how the effect sizes should be computed. It belongs to the `effect_args` argument of the `crosstable()` function. See [effect_summary](#), [effect_tabular](#), and [effect_survival](#) for more insight.

Usage

```
crosstable_effect_args()
```

Value

A list with testing parameters:

- `effect_summarize` - a function of three arguments (continuous variable, grouping variable and `conf_level`), used to compare continuous variable. Returns a list of five components: `effect` (the effect value(s)), `ci` (the matrix of confidence interval(s)), `effect.name` (the interpretation(s) of the effect value(s)), `effect.type` (the description of the measure used) and `conf_level` (the confidence interval level). See [diff_mean_auto\(\)](#), [diff_mean_student\(\)](#), [diff_mean_boot\(\)](#), or [diff_median\(\)](#) for some examples of such functions. Users can provide their own function.
- `effect_tabular` - a function of three arguments (two categorical variables and `conf_level`) used to measure the associations between two factors. Returns a list of five components: `effect` (the effect value(s)), `ci` (the matrix of confidence interval(s)), `effect.name` (the interpretation(s) of the effect value(s)), `effect.type` (the description of the measure used) and `conf_level` (the confidence interval level). See [effect_odds_ratio\(\)](#), [effect_relative_risk\(\)](#), or [effect_risk_difference\(\)](#) for some examples of such functions. Users can provide their own function.
- `effect_survival` - a function of two argument (a formula and `conf_level`), used to measure the association between a censored and a factor. Returns the same components as created by `effect_summarize`. See [effect_survival_coxph\(\)](#). Users can provide their own function.
- `conf_level` - the desired confidence interval level
- `digits` - the decimal places
- `show_effect` - a function to format the effect. See [display_effect\(\)](#).

Author(s)

Dan Chaltiel

crosstable_options *Options for the package crosstable*

Description

Here is a comprehensive list of all options that you can set globally.

For flextables:

- `crosstable_autofit`, `crosstable_compact`, and `crosstable_show_test_name`: default arguments for [as_flextable\(\)](#).
- `crosstable_compact_padding`: left-padding for non-headers rows when `compact=TRUE`.
- `crosstable_fontsize_body`, `crosstable_fontsize_header`, and `crosstable_fontsize_subheaders`: font sizes for, respectively for normal, header and subheader rows. Subheaders are only considered when `compact=TRUE`.
- `crosstable_wrap_id`: if `id` contains no spaces, wrap it with this maximum number of characters.

For specifying styles in officer docx output:

- `crosstable_style_normal`

- `crosstable_style_character`: used in cross-references
- `crosstable_style_heading`
- `crosstable_style_strong`
- `crosstable_style_image`
- `crosstable_style_legend`
- `crosstable_style_list_ordered` and `crosstable_style_list_unordered`, mandatory for `body_add_list()` to work.

Verbosity:

- `crosstable_verbosity_autotesting`: one of default, quiet, or verbose

Misc:

- `crosstable_only_round`: default argument for `format_fixed()`
- `crosstable_units`: default units in `body_add_gg2()` and `body_add_img2()`

`crosstable_test_args` *Default arguments for calculating and displaying tests in `crosstable()`*

Description

This is the starting point for refining the testing algorithm used in `crosstable`. Users can provide their own functions for `test.~`.

Usage

```
crosstable_test_args()
```

Value

A list with testing parameters:

- `test_summarize` - a function of two arguments (continuous variable and grouping variable), used to compare continuous variable. Must return a list of two components: `p.value` and `method`. See `test_summarize_auto` or `test_summarize_linear_contrasts` for some examples of such functions.
- `test_tabular` - a function of two arguments (two categorical variables), used to test association between two categorical variables. Must return a list of two components: `p.value` and `method`. See `test_tabular_auto` for example.
- `test_correlation` - a function of three arguments (two continuous variables plus the correlation method), used to test association between two continuous variables. Like `cor.test`, it must return a list of at least `estimate`, `p.value`, and `method`, with also `conf.int` optionally. See `test_correlation_auto` for example.

- `test_survival` - a function of one argument (the formula `surv~by`), used to compare survival estimations. Must return a list of two components: `p.value` and `method`. See [test_survival_logrank](#) for example.
- `display_test` - function used to display the test result. See [display_test](#).
- `plim` - number of digits for the p value
- `show_method` - whether to display the test name (logical)

Author(s)

Dan Chaltiel

See Also

[test_summarize_auto](#), [test_tabular_auto](#), [test_survival_logrank](#), [test_summarize_linear_contrasts](#), [display_test](#)

Examples

```
library(dplyr)
my_test_args=crosstable_test_args()
my_test_args$test_summarize = test_summarize_linear_contrasts
iris %>%
  mutate(Petal.Width.qt = paste0("Q", ntile(Petal.Width, 5)) %>% ordered()) %>%
  crosstable(Petal.Length ~ Petal.Width.qt, test=TRUE, test_args = my_test_args)
```

cross_summary

Summarize a numeric vector

Description

Summarize a numeric vector with min, max, mean, sd, median, IQR, n and missings.

Usage

```
cross_summary(x, dig = 1, ...)
```

Arguments

<code>x</code>	a numeric vector
<code>dig</code>	number of digits
<code>...</code>	params to pass on to format_fixed() : <code>zero_digits</code> and <code>only_round</code>

Value

a list of named functions

Examples

```

cross_summary(iris$Sepal.Length)
cross_summary(iris$Petal.Width, dig=3)
cross_summary(mtcars2$hp_date)
cross_summary(mtcars2$qsec_posix, date_format="%d/%m %H:%M")

```

display_effect	<i>Default function to display the effect</i>
----------------	---

Description

User can provide their own custom version in [crosstable_effect_args\(\)](#)

Usage

```
display_effect(effect, digits = 4)
```

Arguments

effect	effect
digits	digits

Value

a character vector

display_test	<i>Default function to display a test result</i>
--------------	--

Description

Default function to display a test result

Usage

```
display_test(test, digits = 4, method = TRUE)
```

Arguments

test	test
digits	number of digits
method	display method

Value

a string

docx_bookmarks2	<i>List Word bookmarks, including the ones in header and footer</i>
-----------------	---

Description

This is a correction of `officer::docx_bookmarks()`. See [this PR](#).

Usage

```
docx_bookmarks2(x, return_vector = FALSE)
```

Arguments

`x` an rdocx object
`return_vector` use TRUE for compatibility with `officer::docx_bookmarks()`

Value

a list with all bookmarks

Author(s)

Dan Chaltiel

effect_summary	<i>Effect measure for association between one continuous and one categorical variable</i>
----------------	---

Description

User can either use or extend these functions to parametrize effect calculation.

Usage

```
diff_mean_auto(x, g, conf_level = 0.95, R = 500)
```

```
diff_mean_boot(x, g, conf_level = 0.95, R = 500)
```

```
diff_median(x, g, conf_level = 0.95, R = 500)
```

```
diff_mean_student(x, g, conf_level = 0.95)
```

Arguments

x	a numeric vector
g	another vector (of exactly 2 unique levels)
conf_level	confidence interval level
R	number of bootstrap replication

Value

A list with five components: effect, ci, effect.name, effect.type, and conf_level

Functions

- `diff_mean_auto`: **(Default)** calculate a specific "difference in means" effect based on normality (Shapiro or Anderson test) and variance homogeneity (Bartlett test)
- `diff_mean_boot`: calculate a "difference in means" effect by bootstrapping
- `diff_median`: calculate a "difference in medians" effect by bootstrapping
- `diff_mean_student`: calculate a "difference in means" effect using `t.test` confidence intervals

See Also

[crosstable_effect_args\(\)](#)

effect_survival	<i>Effect measure for association between one censored variable and one categorical variable</i>
-----------------	--

Description

Effect measure for association between one censored variable and one categorical variable

Usage

```
effect_survival_coxph(formula, conf_level = 0.95)
```

Arguments

formula	a formula
conf_level	the confidence level required

Value

a list with two components: p.value and method

effect_tabular	<i>Effect measure for association between two categorical variables</i>
----------------	---

Description

User can either use or extend these functions to parametrize effect calculation.

Usage

```
effect_odds_ratio(x, y, conf_level = 0.95)
```

```
effect_relative_risk(x, y, conf_level = 0.95)
```

```
effect_risk_difference(x, y, conf_level = 0.95)
```

Arguments

x vector (of exactly 2 unique levels)

y another vector

conf_level confidence interval level

Value

A list with five components: effect, ci, effect.name, effect.type, and conf_level

Functions

- effect_odds_ratio: (**Default**) calculate the odds ratio
- effect_relative_risk: calculate the relative risk
- effect_risk_difference: calculate the risk difference

See Also

[crosstable_effect_args\(\)](#)

format_fixed	<i>Format numbers with the exact same number of decimals, including trailing zeros</i>
--------------	--

Description

Format numbers with the exact same number of decimals, including trailing zeros

Usage

```
format_fixed(
  x,
  digits = 1,
  zero_digits = 1,
  date_format = NULL,
  only_round = getOption("crosstable_only_round", FALSE),
  ...
)
```

Arguments

x	a numeric vector to format
digits	number of decimals
zero_digits	number of significant digits for values rounded to 0 (can be set to NULL to keep the original 0 value)
date_format	if x is a vector of Date or POSIXt, the format to apply (see strptime for formats)
only_round	if TRUE, format_fixed simply returns the rounded value. Can be set globally with options("crosstable_only_round"=TRUE).
...	unused

Value

a character vector of formatted numbers

Author(s)

Dan Chaitiel

Examples

```
x = c(1, 1.2, 12.78749, pi, 0.00000012)
format_fixed(x, digits=3) #default zero_digits=1
format_fixed(x, digits=3, zero_digits=2)
format_fixed(x, digits=3, zero_digits=NULL)

x_sd = sd(iris$Sepal.Length/10000, na.rm=TRUE)
format_fixed(x_sd, dig=6)
```

```

format_fixed(x_sd, dig=3, zero_digits=2) #default only_round=FALSE
format_fixed(x_sd, dig=3, zero_digits=2, only_round=TRUE)
options("crosstable_only_round"=TRUE)
format_fixed(x_sd, dig=3, zero_digits=2) #override default
options("crosstable_only_round"=NULL)

```

get_label	<i>Get label if wanted and available, or default (name) otherwise</i>
-----------	---

Description

Get label if wanted and available, or default (name) otherwise

Usage

```
get_label(x, default = names(x), object = FALSE, simplify = TRUE)
```

Arguments

x	labelled object. If x is a list/data.frame, get_label() will return the labels of all children recursively
default	value returned if there is no label. Default to names(x).
object	if x is a list/data.frame, object=TRUE will force getting the labels of the object instead of the children
simplify	if x is a list and object=FALSE, simplify the result to a vector

Value

A character vector if simplify==TRUE, a list otherwise

See Also

[set_label\(\)](#), [import_labels\(\)](#), [remove_label\(\)](#), [Hmisc::label\(\)](#), [expss::var_lab\(\)](#)

Examples

```

xx=mtcars2 %>%
  set_label("The mtcars2 dataset", object=TRUE)
xx$cyl=remove_label(xx$cyl)

#vectors
get_label(xx$mpg) #label="Miles/(US) gallon"
get_label(xx$cyl) #default to NULL (as names(xx$cyl)==NULL)
get_label(xx$cyl, default="Default value")

#data.frames
get_label(xx)
get_label(xx, object=TRUE)

```

```
data.frame(name=names(xx), label=get_label(xx, default=NA)) #cyl is NA

#lists
get_label(list(xx$cyl, xx$mpg))
get_label(list(foo=xx$cyl, bar=xx$mpg))
get_label(list(foo=xx$cyl, bar=xx$mpg), default="Default value")
```

import_labels	<i>Import labels</i>
---------------	----------------------

Description

import_labels imports labels from a data.frame (data_label) to another one (.tbl). Works in synergy with [save_labels\(\)](#).

save_labels saves the labels from a data.frame in a temporary variable that can be retrieve by import_labels.

Usage

```
import_labels(
  .tbl,
  data_label,
  name_from = "name",
  label_from = "label",
  verbose_name = FALSE,
  verbose_label = FALSE,
  verbose = deprecated()
)

save_labels(.tbl)
```

Arguments

.tbl	the data.frame to labellize
data_label	a data.frame from which to import labels. If missing, the function will take the labels from the last dataframe on which save_labels() was called.
name_from	in data_label, which column to get the variable name (default to name)
label_from	in data_label, which column to get the variable label (default to label)
verbose_name	if TRUE, displays a warning if a variable name is not found in data_label
verbose_label	if TRUE, displays a warning if a label is not found in .tbl
verbose	deprecated

Value

A dataframe, as .tbl, with labels
 .tbl invisibly. Used only for its side effects.

See Also

[get_label\(\)](#), [set_label\(\)](#), [remove_label\(\)](#), [save_labels\(\)](#)

Examples

```
#import the labels from a data.frame to another
iris_label = data.frame(
  name=c("Sepal.Length", "Sepal.Width",
        "Petal.Length", "Petal.Width", "Species"),
  label=c("Length of Sepals", "Width of Sepals",
         "Length of Petals", "Width of Petals", "Specie name")
)
iris %>%
  import_labels(iris_label) %>%
  crosstable

#save the labels, use some dplyr label-removing function, then retrieve the labels
library(dplyr)
mtcars2 %>%
  save_labels() %>%
  transmute(dispatch=as.numeric(dispatch)+1) %>%
  import_labels(verbose_label=FALSE) %>% #
  crosstable(dispatch)
```

 iris2

Modified iris dataset

Description

Modified iris dataset so:

- every column is labelled (using label attribute from expss package, compatible with Hmisc package)
- Species column is considered as factor

See [iris](#) for more informations on the original "Edgar Anderson's Iris Data" dataset.

Usage

```
iris2
```

Format

A data frame with 150 observations on 5 variables with labels.

Source

```
library(dplyr)
iris2 = iris %>%
  mutate_at("Species", factor) %>%
  expss::apply_labels( #I also could have used [import_labels] or even `Hmisc::label`
    Species = "Specie",
    Sepal.Length = "Length of Sepal",
    Sepal.Width = "Width of Sepal",
    Petal.Length = "Length of Petal",
    Petal.Width = "Width of Petal"
  ) %>%
  as_tibble()
```

Examples

```
library(crosstable)
ct=crosstable(iris2, by=Species)
ct
as_flextable(ct)
```

mtcars2

Modified mtcars dataset

Description

Modified mtcars dataset so:

- every column is labelled (using label attribute from expss package, compatible with Hmisc package)
- gear and cyl columns are considered as numerical factors
- vs and am columns are considered as character vector

See [mtcars](#) for more informations on the original "Motor Trend Car Road Tests" dataset.

Usage

```
mtcars2
```

Format

A data frame with 32 observations on 11 variables with labels.

Source

```
library(dplyr)
mtcars2 = mtcars %>%
  mutate(vs=ifelse(vs==0, "vshaped", "straight"),
         am=ifelse(am==0, "auto", "manual")) %>%
  mutate_at(c("cyl", "gear"), factor) %>%
  expss::apply_labels( #I also could have used [import_labels] or even `Hmisc::label`
    mpg="Miles/(US) gallon",
    cyl="Number of cylinders",
    disp="Displacement (cu.in.)",
    hp="Gross horsepower",
    drat="Rear axle ratio",
    wt="Weight (1000 lbs)",
    qsec="1/4 mile time",
    vs="Engine",
    am="Transmission",
    gear="Number of forward gears",
    carb="Number of carburetors"
  )
```

Examples

```
library(crosstable)
ct=crosstable(mtcars2, by=vs)
ct
as_flextable(ct)
```

N

Return the number of non NA observations

Description

Return the number of non NA observations

Usage

`N(x)`

Arguments

`x` a vector

Value

integer, number of non NA observations

Author(s)

David Hajage

na	<i>Return the number of NA observations</i>
----	---

Description

Return the number of NA observations

Usage

```
na(x)
```

Arguments

x a vector

Value

integer, number of NA observations

Author(s)

David Hajage

peek	<i>Open a crosstable in a temporary Word document</i>
------	---

Description

This eases copy-pasting

Usage

```
peek(x, ...)
```

Arguments

x a crosstable
... passed on to `as_flextable.crosstable()`

Value

Nothing, called for its side effects

plim	<i>Format p values (alternative to format.pval())</i>
------	---

Description

Format p values (alternative to [format.pval\(\)](#))

Usage

```
plim(p, digits = 4)
```

Arguments

p	p values
digits	number of digits

Value

formatted p values

Author(s)

David Hajage

See Also

[format.pval\(\)](#), <https://stackoverflow.com/a/23018806/3888000>

remove_labels	<i>Remove all label attributes.</i>
---------------	-------------------------------------

Description

Use `remove_labels()` to remove the label from an object or to recursively remove all the labels from a collection of objects (such as a list or a `data.frame`).

This can be useful with functions reacting badly to labelled objects.

Usage

```
remove_labels(x)
```

Arguments

x	object to unlabel
---	-------------------

Value

An object of the same type as x, with no labels

See Also

[get_label](#), [set_label](#), [import_labels](#), [expss::unlab](#)

Examples

```
mtcars2 %>% remove_labels %>% crosstable(1:2) #no labels
mtcars2$hp %>% remove_labels %>% get_label #numeric
```

rename_dataframe_with_labels

Rename every column of a dataframe with its label

Description

Rename every column of a dataframe with its label

Usage

```
rename_dataframe_with_labels(df)
```

Arguments

df a data.frame

Value

A dataframe, as df, which names are copied from the label attribute

Examples

```
library(dplyr)
mtcars2 %>%
  select(1:5) %>%
  rename_dataframe_with_labels()
```

set_label	<i>Set the "label" attribute of an object</i>
-----------	---

Description

Set the "label" attribute of an object

Copy the label from one variable to another

Usage

```
set_label(x, value, object = FALSE)
```

```
copy_label_from(x, from)
```

Arguments

x	the variable to label
value	value of the label. If x is a list/data.frame, all the labels will be set recursively
object	if x is a list/data.frame, object=TRUE will force setting the labels of the object instead of the children
from	the variable whose label must be copied

Value

An object of the same type as x, with labels

An object of the same type as x, with the label of from

See Also

[get_label\(\)](#), [import_labels\(\)](#), [remove_label\(\)](#)

Examples

```
library(dplyr)
mtcars %>%
  mutate(mpg2=set_label(mpg, "Miles per gallon"),
         mpg3=mpg %>% copy_label_from(mpg2)) %>%
  crosstable(mpg, mpg2, mpg3)
```

summaryFunctions *Summary functions*

Description

Summary functions to use with `crosstable()` or anywhere else.

Usage

```
meansd(x, na.rm = TRUE, dig = 2, ...)
```

```
meanCI(x, na.rm = TRUE, dig = 2, level = 0.95, format = TRUE, ...)
```

```
mediqr(x, na.rm = TRUE, dig = 2, format = TRUE, ...)
```

```
minmax(x, na.rm = TRUE, dig = 2, ...)
```

```
nna(x)
```

Arguments

x	a numeric vector
na.rm	TRUE as default
dig	number of digits
...	params to pass on to <code>format_fixed()</code> : <ul style="list-style-type: none"> • <code>zero_digits</code> (default=1): the number of significant digits for values rounded to 0 (set to NULL to keep the original 0 value) • <code>only_round</code> (default=FALSE): use <code>round()</code> instead of <code>format_fixed()</code>
level	the confidence level required
format	a sugar argument. If FALSE, the function returns a list instead of a formatted string

Value

a character vector

Functions

- `meansd`: returns mean and std error
- `meanCI`: returns mean and confidence interval
- `mediqr`: returns median and IQR
- `minmax`: returns minimum and maximum
- `nna`: returns number of observations and number of missing values

Fixed format

These functions use `format_fixed()` which allows to have trailing zeros after rounded values. In the case when the output of rounded values is zero, the use of the `zero_digits` argument allows to keep some significant digits for this specific case only.

Author(s)

Dan Chaltiel
David Hajage

See Also

[format_fixed\(\)](#)

Examples

```
meansd(iris$Sepal.Length, dig=3)
meanCI(iris$Sepal.Length)
minmax(iris$Sepal.Length, dig=3)
mediqr(iris$Sepal.Length, dig=3)
nna(iris$Sepal.Length)

#arguments for format_fixed
x = iris$Sepal.Length/10000 #closer to zero

meansd(x, dig=3)
meansd(x, dig=3, zero_digits=NULL) #or NA
meansd(x, dig=3, only_round=TRUE)
options("crosstable_only_round"=TRUE)
meansd(x, dig=3, zero_digits=2)
options("crosstable_only_round"=NULL)
meanCI(mtcars2$x_date)

#dates
x = as.POSIXct(mtcars$qsec*3600*24 , origin="2010-01-01")
meansd(x)
minmax(x, date_format="%d/%m/%Y")
```

test_correlation_auto *test for correlation coefficients*

Description

test for correlation coefficients

Usage

```
test_correlation_auto(x, by, method)
```

Arguments

x	vector
by	another vector
method	"pearson", "kendall", or "spearman"

Value

the correlation test with appropriate method

test_summarize_auto *test for mean comparison*

Description

Compute a oneway.test (with equal or unequal variance) or a kruskal.test as appropriate.

Usage

```
test_summarize_auto(x, g)
```

Arguments

x	vector
g	another vector

Value

a list with two components: p.value and method

Author(s)

David Hajage, Dan Chaltiel

`test_summarize_linear_contrasts`*Test for linear trend across ordered factor with contrasts*

Description

Test for linear trend across ordered factor with contrasts

Usage

```
test_summarize_linear_contrasts(x, y)
```

Arguments

x	vector
y	ordered factor

Value

a list with two components: p.value and method

Author(s)

Dan Chaltiel

Examples

```
library(dplyr)
my_test_args=crosstable_test_args()
my_test_args$test_summarize = test_summarize_linear_contrasts
iris %>%
  mutate(Petal.Width.qt = paste0("Q", ntile(Petal.Width, 5)) %>% ordered()) %>%
  crosstable(Petal.Length ~ Petal.Width.qt, test=TRUE, test_args = my_test_args)
```

`test_survival_logrank` *test for survival comparison*

Description

Compute a logrank test

Usage

```
test_survival_logrank(formula)
```

Arguments

formula a formula

Value

a list with two components: p.value and method

Author(s)

David Hajage

test_tabular_auto *test for contingency table*

Description

Compute a `chisq.test`, a `chisq.test` with correction of continuity or a fisher test as appropriate

Usage

```
test_tabular_auto(x, y)
```

Arguments

x vector
y another vector

Value

a list with two components: p.value and method

Author(s)

David Hajage

write_and_open	<i>Alternative to default officer::print.rdocx() function. Write the file and try to open it right away.</i>
----------------	--

Description

As it tests if the file is writable, this function also prevents `officer::print.rdocx()` to abort the RStudio session.

Usage

```
write_and_open(doc, docx.file)
```

Arguments

doc	the docx object
docx.file	the name of the target file. If missing or NULL, the doc will open in a temporary file.

Value

Nothing, called for its side effects

Author(s)

Dan Chaltiel

Examples

```
library(officer)
library(crosstable)
mytable = crosstable(mtcars2)
doc = read_docx() %>%
  body_add_crosstable(mytable)

write_and_open(doc)
## Not run:
write_and_open(doc, "example.docx")

## End(Not run)
```

Index

- * **as_gt methods**
 - as_gt.crosstable, 3
- * **datasets**
 - iris2, 29
 - mtcars2, 30

- apply_labels, 3
- as_flextable(), 17, 19
- as_flextable.crosstable(), 4, 6
- as_gt (as_gt.crosstable), 3
- as_gt.crosstable, 3
- as_workbook, 5

- body_add_crosstable, 6
- body_add_crosstable(), 6, 7
- body_add_crosstable_footnote, 7
- body_add_figure_legend
 - (body_add_table_legend), 11
- body_add_figure_legend(), 12
- body_add_gg2, 7
- body_add_gg2(), 9, 20
- body_add_glued (body_add_normal), 10
- body_add_img2, 8
- body_add_img2(), 20
- body_add_list, 9
- body_add_list(), 20
- body_add_list_item (body_add_list), 9
- body_add_normal, 10
- body_add_normal(), 12
- body_add_table_legend, 11
- body_add_table_legend(), 10, 12
- body_add_title, 13

- compact, 14
- confint_numeric, 15
- copy_label_from (set_label), 35
- cross_summary, 17, 21
- crosstable, 16
- crosstable(), 5, 18, 20, 36
- crosstable_effect_args, 17, 18

- crosstable_effect_args(), 22, 24, 25
- crosstable_options, 19
- crosstable_test_args, 17, 20

- diff_mean_auto (effect_summary), 23
- diff_mean_auto(), 19
- diff_mean_boot (effect_summary), 23
- diff_mean_boot(), 19
- diff_mean_student (effect_summary), 23
- diff_mean_student(), 19
- diff_median (effect_summary), 23
- diff_median(), 19
- display_effect, 22
- display_effect(), 19
- display_test, 21, 22
- docx_bookmarks2, 23

- effect_odds_ratio (effect_tabular), 25
- effect_odds_ratio(), 19
- effect_relative_risk (effect_tabular), 25
- effect_relative_risk(), 19
- effect_risk_difference
 - (effect_tabular), 25
- effect_risk_difference(), 19
- effect_summary, 18, 23
- effect_survival, 18, 24
- effect_survival_coxph
 - (effect_survival), 24
- effect_survival_coxph(), 19
- effect_tabular, 18, 25
- exps::apply_labels(), 3
- exps::unlab, 34
- exps::var_lab(), 27

- flextable(), 14
- format.pval(), 33
- format_fixed, 17, 26
- format_fixed(), 20, 21, 36, 37

- get_label, 27, 34

get_label(), 29, 35
ggplot2::ggsave(), 8
glue::glue(), 12
gt::gt(), 4

Hmisc::label(), 27

import_labels, 17, 28, 34
import_labels(), 27, 35
iris, 29
iris2, 29

meanCI (summaryFunctions), 36
meansd (summaryFunctions), 36
mediqr (summaryFunctions), 36
minmax (summaryFunctions), 36
moystd (summaryFunctions), 36
mtcars, 30
mtcars2, 30

N, 31
na, 32
nna (summaryFunctions), 36

officer::body_add_gg(), 7
officer::body_add_img(), 8, 9
officer::body_add_par(), 10
officer::docx_bookmarks(), 23
officer::read_docx(), 6

peek, 32
plim, 33

remove_label (remove_labels), 33
remove_label(), 27, 29, 35
remove_labels, 33
rename_dataframe_with_labels, 34
round(), 36

save_labels (import_labels), 28
save_labels(), 28, 29
set_label, 17, 34, 35
set_label(), 27, 29
stats::confint, 15
strptime, 17, 26
summaryFunctions, 36
survival::Surv(), 17

test_args (crosstable_test_args), 20
test_correlation_auto, 20, 37
test_summarize_auto, 20, 21, 38
test_summarize_linear_contrasts, 20, 21, 39
test_survival_logrank, 21, 39
test_tabular_auto, 20, 21, 40
write_and_open, 41