Package 'summarySCI'

November 13, 2026

Title Produces Publication-Ready Summary Tables and Plots

Version 1.0.0

Description Produces tables with descriptive statistics for continuous, categorical and dichotomous variables. It is largely based on the package 'gtsummary'; Sjoberg DD et al. (2021) <doi:10.32614/RJ-2021-053>. The tables can be accompanied by matching boxplots, line plots and stacked barplots, also over time.

License LGPL-3

Encoding UTF-8

RoxygenNote 7.3.2

Depends R (>= 4.1)

Imports Hmisc, cardx, dplyr, gtsummary (>= 2.3.0), forcats, labelled, purrr, rlang, tidyr, flextable, officer, graphics, grDevices, stats, utils, tools

Suggests broom (>= 1.0.8), knitr, survival, testthat (>= 3.0.0), rmarkdown

VignetteBuilder knitr

URL https://sakk-statistics.github.io/summarySCI/, https://github.com/SAKK-Statistics/summarySCI

BugReports https://github.com/SAKK-Statistics/summarySCI/issues

Config/testthat/edition 3

NeedsCompilation no

Author Sämi Schär [aut], Charlotte Micheloud [cre, aut], Tudor Jumuga [aut]

Maintainer Charlotte Micheloud <Charlotte.Micheloud@swisscancerinstitute.ch>

Repository CRAN

Date/Publication 2026-11-13 16:10:08 UTC

Contents

boxplotSCI . . . 2
hemoglobin_data . . . 6
lineplotSCI . . . 6
plotByVisit . . . 10
save_plot . . . 12
save_plots . . . 13
sci_plot . . . 14
stackedbarplotSCI . . . 14
summarizeByVisitSCI . . . 19
summarizeSCI . . . 22
summaryByVisit . . . 24
summaryByVisitCategorical . . . 26
summaryByVisitContinuous . . . 28
summaryLevels . . . 30
summaryTable . . . 32
tidySummaryByVisit . . . 35

Index **36**

`boxplotSCI` *Boxplot over time*

Description

Generates boxplots over time for one or multiple sub-groups.

Usage

```
boxplotSCI(
  data,
  var,
  order = NULL,
  visit = NULL,
  group = NULL,
  group_names = NULL,
  group_names_short = NULL,
  main = "",
  xlab = NULL,
  ylab = "Variable",
  ylim = NULL,
  y_ticks = NULL,
  col = NULL,
  cex_lab = 1,
  cex_axis = 1,
  cex_n_patients = 0.7,
  boxwex = NULL,
```

```
  shift = NULL,
  mar_custom = NULL,
  xlab_line = NULL,
  ylab_line = 3,
  xlab_position = NULL,
  lines = NULL,
  lty = NULL,
  col_lines = NULL,
  legend_position = "topright",
  legend_offset = c(0, 0),
  legend_cex = 1,
  legend_bty = "n",
  risk_table = NULL,
  missing_nr = 0,
  color_time = FALSE,
  stat_footnote = FALSE,
  ...
)
```

Arguments

`data`	A data frame or tibble containing the data to be plotted.
`var`	Name of the column in `data` indicating the measurement to be plotted. Supports bare or quoted column names.
`order`	Name of the column in `data` containing the time points of interest. The time points can be either numeric, character, or a factor. If the time points are factors, the factor levels must be in the correct chronological order for plotting. If left empty, the function will assume a single time point. If character, they need to be able to be ordered by numerical logic.
`visit`	Name of the column in `data` containing descriptive labels for the time points. Optional.
`group`	Name of the column in `data` containing the grouping variables (e.g., treatment arms). If left empty, the function will assume a single group.
`group_names`	A character vector containing the names of the groups to be displayed in the legend. The order must match `levels(factor(data$group))`. Default is `NULL`, which falls back to the group names found in the data.
`group_names_short`	A character vector containing short names or abbreviations for the groups, to be displayed below the x-axis near the patient counts. Optional. Defaults to `group_names`.
`main`	Title of the plot. Defaults to `""` (no title).
`xlab`	A label for the x-axis. Default is `"Time"` if there are multiple time points, otherwise defaults to `"Group"`.
`ylab`	A label for the y-axis. Default is `"Variable"`.
`ylim`	The y-axis limits of the plot as a numeric vector of length 2. Default is `NULL`, which adds a 10% padding buffer above and below the data range.

y_ticks	A numeric vector specifying the explicit tick positions on the y-axis. Default is NULL, which places about 5 ticks at round values within the y-axis limits, as in the other plots of the package.
col	A character vector of hex colors or color names to use for the boxes. Default is NULL, which automatically applies the SCI color palette.
cex_lab	A numeric value specifying the font size of the main axis labels. Default is 1.
cex_axis	A numeric value specifying the font size of the axis tick marks. Default is 1.
cex_n_patients	A numeric value specifying the font size of the sample size labels and numbers displayed beneath the horizontal axis. Default is 0.7.
boxwex	Scale factor for the width of the boxes. Default is NULL, which scales the box flexibly to the size of total timepoints and groups.
shift	A numeric value indicating the distance between adjacent boxes within a time-point cluster. Default is NULL, which dynamically sets spacing to boxwex * 1.15.
mar_custom	A numeric vector indicating the plot margins to be used in the form c(bottom, left, top, right). Default is NULL, which dynamically calculates spacing so that stacked sample-size rows are never clipped.
xlab_line	A numeric value indicating the margin line offset for the x-axis label. Default is NULL, which calculates an adaptive offset based on the number of groups.
ylab_line	A numeric value indicating the margin line offset for the y-axis label. Default is 3.0.
xlab_position	Horizontal position coordinate adjustment for the sample size group headers down the left-hand margin. Default is NULL, which auto-aligns them.
lines	A numeric vector indicating the vertical coordinates where horizontal reference lines will be plotted. Optional.
lty	A numeric vector of length 1 or of the same length as lines indicating the line type to be plotted. Default is 1 (solid).
col_lines	A character vector of length 1 or of the same length as lines indicating the color of the reference lines. Default is "grey".
legend_position	Position of the legend box. Options include standard base R keywords (e.g., "topright", "topleft", "bottom") or manual numeric coordinate pairs c(x, y). Default is "topright".
legend_offset	A numeric vector of length 2 specifying an inset to nudge the legend position away from plot elements. Default is c(0, 0).
legend_cex	A numeric value specifying the font and element scaling size inside the legend box. Default is 1.
legend_bty	The type of box border to be drawn around the legend. Allowed values are "n" (transparent/no border, the default) and "o" (standard box frame).
risk_table	A logical toggle switch enabling the patient numbers (non-missing observations of var) beneath the x-axis: one row per group over time, or one number under each box for a single time point. Default is NULL, which shows them over time but not for a single time point; TRUE shows them in both cases, FALSE in neither.

`missing_nr` A numeric or character value indicating how zero-count patient sizes on the x-axis should be displayed. Default is `0`. Can also be set to `NA`.

`color_time` A logical toggle switch enabling legacy timepoint colorization. Supports legacy use, where, for a single timepoint, multiple groups were used as "timepoint" variable. If set to `TRUE`, it treats timepoints as individual categories,applying sequential coloring and rendering a matching tracking legend. Default is `FALSE`.

`stat_footnote` Show what the plot displays as a footnote in the bottom-right corner, beneath the x-axis title: `TRUE` for the default text (e.g. the definition of box and whiskers), a character string for a text of your own, or `FALSE` (the default) for none.

`...` Further graphical parameters passed on to `bxp()`, which draws the boxes for `boxplot()`, e.g. `lwd`, `border`, `outline = FALSE` (no points for outliers) or `range` (the whisker length, see above; `0` for whiskers to the minimum and maximum).

Details

The box spans the first to the third quartile with the median inside, computed as in the summary tables (`summaryTable()`, `summaryByVisit()`: `quantile(type = 2)`), so that a boxplot and a `"median_IQR"` table show the same numbers. (`boxplot()` itself draws Tukey's hinges, which can differ slightly.) The whiskers reach the most extreme values within `range` (1.5 by default) times the box length from the box; values beyond are drawn as points.

Examples

```
library(summarySCI)
data("hemoglobin_data")
boxplotSCI(data = hemoglobin_data,
                 var = hb,
                  order = visit_nr,
                  group = Arm,
                  group_names = c("Treatment (Arm A)", "Control (Arm B)"),
                  group_names_short = c("A", "B"),
                  # Title
                  main = "Hemoglobin over time",
                  # Col of boxes
                  col = c("lightblue", "orange"),
                  # X axis label
                  xlab = "Visits",
                  # X axis position
                  xlab_position = 0,
                  # Y axis label
                  ylab = "Score",
                  lines = c(90, 120, 160),
                  # Col of lines
                  col_lines = c("red", "grey", "red"),
                  # Type of lines
                  lty = c(2, 3, 2)
)
```

hemoglobin_data *Hemoglobin data (to illustrate* `boxplotSCI()` *and* `summaryByVisit()`*)*

Description

A dataset containing the hemoglobin value of 169 patients over 9 visits. Adapted from SAKK 08/14.

Usage

```
data(hemoglobin_data)
```

Format

`hemoglobin_data`**:**

A data frame with 1307 rows and 5 columns:

ID Patient ID

Arm Experimental arm ("Arm A", "Arm B")

visit Visit (character, "Cycle 1" to "Cycle 9")

visit_nr Visit number (numeric, 1 to 9)

hb Hemoglobin value (g/L)

Source

SAKK

lineplotSCI *Line plot of a summary statistic over time, by group*

Description

Generates line plots of a central-tendency statistic over time for one or multiple sub-groups, optionally with a dispersion band (ribbon or whiskers) and a table of patient numbers beneath the horizontal axis.

Usage

```
lineplotSCI(
  data,
  var,
  order,
  group = NULL,
  visit = NULL,
```

```
  n_order = NULL,
  group_names = NULL,
  group_names_short = NULL,
  stat = c("median", "mean", "geomean"),
  interval = c("iqr", "range", "ci", "sd", "se", "gsd", "none"),
  interval_type = c("whiskers", "ribbon"),
  alpha = 0.05,
  dodge = 0,
  connect = TRUE,
  connect_shade = 0.15,
  main = "",
  xlab = "Time",
  ylab = "Variable",
  xlim = NULL,
  ylim = NULL,
  y_ticks = NULL,
  col = NULL,
  cex_lab = 1,
  cex_axis = 1,
  cex_n_patients = 0.7,
  risk_table = FALSE,
  missing_nr = 0,
  rotate_labels = FALSE,
  mar_custom = NULL,
  xlab_line = NULL,
  ylab_line = 3,
  xlab_position = NULL,
  lines = NULL,
  lty = 1,
  col_lines = "grey",
  legend_position = "topright",
  legend_title = NULL,
  legend_offset = c(0, 0),
  legend_cex = 1,
  legend_bty = "n",
  stat_footnote = FALSE,
  ...
)
```

Arguments

data	A data frame or tibble containing the data to be plotted.
var	Name of the column in data indicating the measurement to be plotted. Supports bare or quoted column names.
order	Name of the column in data containing the time points of interest. The time points must be numeric or coercible to numeric, since they are used as the plotting coordinates on the x-axis. Rows with a missing time point are dropped.
group	Name of the column in data containing the grouping variable (e.g., treatment

arms). Supports bare or quoted column names. Rows with a missing group are dropped. Default is NULL, which draws a single line, labelled with var (or group_names).

visit — Descriptive labels for the time points. Either the name of a column in data containing the labels (bare or quoted; the first label observed at each time point is used), or a character vector with one entry per tick position. Default is NULL, which prints the time values themselves.

n_order — A numeric value indicating the number of tick positions on the x-axis. Default is NULL, which places one tick at each distinct value of order. Set explicitly to reserve ticks for time points with no observed data, in which case ticks are drawn at 1:n_order.

group_names — A character vector containing the names of the groups to be displayed in the legend. The order must match levels(factor(data$group)). Default is NULL, which falls back to the group names found in the data.

group_names_short — A character vector containing short names or abbreviations for the groups, to be displayed below the x-axis near the patient counts. Optional. Defaults to group_names.

stat — The summary statistic plotted as the point estimate: "median" (the default), "mean" or "geomean" (geometric mean, for positive values: as in the tables, groups and time points with a value of zero or below are left out, with a warning).

interval — The quantity spanned by the error band around the point estimate. One of "iqr" (the default, 25th to 75th percentile, computed as in the summary tables, i.e. quantile(type = 2)), "range" (observed minimum to maximum), "ci" (a two-sided confidence interval at level 1 - alpha), "sd" (point estimate +/- one standard deviation), "se" (point estimate +/- one standard error), "gsd" (geometric mean divided and multiplied by the geometric standard deviation, exp(sd(log(x))); only with stat = "geomean"), or "none" to plot point estimates alone. With stat = "geomean", only "gsd", "iqr", "range" and "none" are available.

interval_type — The style in which the error band is drawn. Either "whiskers" (the default; vertical bars with caps) or "ribbon" (a shaded polygon with a dashed border). Ignored when interval is "none".

alpha — The two-sided significance level used when interval = "ci". Default is 0.05, giving 95% confidence intervals. For stat = "mean" the interval is the usual t-interval; for stat = "median" it is the distribution-free interval based on the binomial order statistics, which is undefined for very small cells and returned as NA there.

dodge — A numeric value giving the total horizontal offset applied across groups, in x-axis units, so that overlapping error bars remain readable. Default is 0 (all groups drawn at the same x position).

connect — A logical toggle switch controlling whether the point estimates within a group are joined by a line. Default is TRUE.

connect_shade — A numeric value between 0 and 1 giving how much lighter the connecting lines are drawn than the point estimates. The color is mixed with white rather than

	made transparent, so the lines stay opaque where they cross error bars or a ribbon. Default is `0.15` (15% lighter); `0` draws the lines in the group color.
`main`	Title of the plot. Defaults to `""` (no title).
`xlab`	A label for the x-axis. Default is `"Time"`.
`ylab`	A label for the y-axis. Default is `"Variable"`.
`xlim`	The x-axis limits of the plot as a numeric vector of length 2. Default is `NULL`, which spans all tick positions and observed time points.
`ylim`	The y-axis limits of the plot as a numeric vector of length 2. Default is `NULL`, which adds a 10% padding buffer above and below the data range (band included, where one is drawn).
`y_ticks`	A numeric vector specifying the explicit tick positions on the y-axis. Default is `NULL`, which falls back to automatic tick placement.
`col`	A character vector of hex colors or color names to use for the groups. Recycled if shorter than the number of groups. Default is `NULL`, which automatically applies the SCI color palette.
`cex_lab`	A numeric value specifying the font size of the main axis labels. Default is `1`.
`cex_axis`	A numeric value specifying the font size of the axis tick marks. Default is `1`.
`cex_n_patients`	A numeric value specifying the font size of the sample size labels and numbers displayed beneath the horizontal axis. Default is `0.7`.
`risk_table`	A logical toggle switch enabling the table of patient numbers (non-missing observations of `var` per group and time point) beneath the x-axis. Default is `FALSE`.
`missing_nr`	A numeric or character value indicating how zero-count patient sizes on the x-axis should be displayed. Default is `0`. Can also be set to `NA`, which leaves the cell blank.
`rotate_labels`	A logical toggle switch rotating the x-axis tick labels by 90 degrees. Default is `FALSE`. Bottom margin and label offsets are adjusted accordingly.
`mar_custom`	A numeric vector indicating the plot margins to be used in the form `c(bottom, left, top, right)`. Default is `NULL`, which dynamically calculates spacing so that stacked sample-size rows are never clipped.
`xlab_line`	A numeric value indicating the margin line offset for the x-axis label. Default is `NULL`, which calculates an adaptive offset based on the number of groups and on `rotate_labels`.
`ylab_line`	A numeric value indicating the margin line offset for the y-axis label. Default is `3.0`.
`xlab_position`	Horizontal position coordinate adjustment for the sample size group headers down the left-hand margin. Default is `NULL`, which auto-aligns them just outside the plot region.
`lines`	A numeric vector indicating the vertical coordinates where horizontal reference lines will be plotted. Optional.
`lty`	A numeric vector of length 1 or of the same length as `lines` indicating the line type to be plotted. Default is `1` (solid).
`col_lines`	A character vector of length 1 or of the same length as `lines` indicating the color of the reference lines. Default is `"grey"`.

`legend_position`
: Position of the legend box. Options include standard base R keywords (e.g., `"topright"`, `"topleft"`, `"bottom"`) or manual numeric coordinate pairs `c(x, y)`. Default is `"topright"`. NULL removes the legend.

`legend_title`
: Title of the legend box. Default is `NULL` (no title).

`legend_offset`
: A numeric vector of length 2 specifying an `inset` to nudge the legend position away from plot elements. Default is `c(0, 0)`. Ignored when `legend_position` is given as coordinates.

`legend_cex`
: A numeric value specifying the font and element scaling size inside the legend box. Default is `1`.

`legend_bty`
: The type of box border to be drawn around the legend. Allowed values are `"n"` (transparent/no border, the default) and `"o"` (standard box frame).

`stat_footnote`
: Show what the plot displays as a footnote in the bottom-right corner, beneath the x-axis title: `TRUE` for the default text (e.g. `"Median (Q1, Q3)"`), a character string for a text of your own, or `FALSE` (the default) for none.

`...`
: Further graphical parameters passed on to `plot()`.

Value

Invisibly, a list with the group-by-time matrices `centre`, `lower`, `upper` (both all-`NA` when `interval` is `"none"`) and `n`, the number of non-missing observations per cell.

`plotByVisit` *Plot a summaryByVisit table*

Description

Plots one variable of a table produced by `summaryByVisit()` over the visits, using the numbers printed in the table rather than the raw data. Continuous variables are drawn as a line plot of the point estimate with its interval (see `lineplotSCI()`), categorical variables as stacked bars (see `stackedbarplotSCI()`).

Usage

```
plotByVisit(
  tbl,
  var = NULL,
  stat_cont = NULL,
  missing = FALSE,
  missing_text = "Missing",
  ...
)
```

Arguments

tbl	A table returned by summaryByVisit() with as_flex_table = FALSE (the default TRUE returns a flextable, from which the numbers cannot be read back).
var	The variable to be plotted, given as its label in the table (the bold header row, e.g. "Hemoglobin") or its position among the variables of the table. May be omitted if the table holds a single variable.
stat_cont	The statistic of continuous variables, as passed to summaryByVisit(). Default is NULL, which reads it from the table footnote; if the footnote is not recognised, plotByVisit() stops and asks for stat_cont rather than guess. A stat_cont that does not fit the printed cells (e.g. "mean_sd" for a median (range) table) is an error too.
missing	A logical toggle switch adding the missing-value rows of a categorical variable as a segment of their own. Default is FALSE, which leaves them out, so that percentages are relative to the non-missing observations.
missing_text	The label of the missing-value rows in the table, as passed to summaryByVisit(). Default is "Missing".
...	Further arguments controlling the appearance of the plot, as documented in lineplotSCI() (continuous variables; e.g. interval_type, risk_table, col, legend_position) or stackedbarplotSCI() (categorical variables; e.g. scale, percent_labels, risk_table). For continuous variables, xs sets the positions of the visits on the x-axis (one number per visit, e.g. the study days); by default they are evenly spaced.

Details

The statistic of continuous variables is read from the table footnote and determines the error bars:

- "median_range": median with whiskers from minimum to maximum;
- "median_IQR": median with whiskers from the first to the third quartile;
- "mean_sd" / "mean_se": mean +/- one standard deviation / standard error;
- "geomMean_sd": geometric mean divided and multiplied by the geometric standard deviation.

Since the numbers are read from the printed table, they carry its rounding (digits_cont / digits_cat). Visits appear on the x-axis in the order of the table rows.

Value

Invisibly, a list with

- type: "continuous" or "categorical";
- summary: the rows of tidySummaryByVisit() belonging to var;
- for continuous variables, the group-by-visit matrices centre, lower, upper and n;
- for categorical variables, the arrays counts and percent indexed by level, group and visit, and the group-by-visit matrix n of denominators.

Examples

```
library(summarySCI)
data("hemoglobin_data")
hb <- subset(hemoglobin_data, visit_nr <= 3)   # first 3 cycles, to keep it short
attr(hb$hb, "label") <- "Hemoglobin"           # subset() drops the label
tbl <- summaryByVisit(hb, vars = "hb",
                      group = "Arm", visit = "visit", stat_cont = "median_IQR",
                      as_flex_table = FALSE)
plotByVisit(tbl, var = "Hemoglobin", risk_table = TRUE)


# a categorical variable
hb$hb_cat <- cut(hb$hb, c(-Inf, 120, 140, Inf),
                 labels = c("< 120", "120-140", "> 140"))
tbl <- summaryByVisit(hb, vars = c("hb", "hb_cat"),
                      group = "Arm", visit = "visit", stat_cont = "median_IQR",
                      as_flex_table = FALSE)
plotByVisit(tbl, var = "hb_cat", legend_title = "Hemoglobin")
```

`save_plot` *Save a plot to a file*

Description

Draws a plot of `summarizeSCI()` into a file. The format follows the file extension: `.png`, `.jpeg`/`.jpg`, `.tiff`, `.pdf` or `.svg`.

Usage

```
save_plot(x, file, width = 8, height = 5, res = 300)
```

Arguments

`x`	A `sci_plot` object, e.g. `res$plots$hb`.
`file`	Path of the file to write.
`width, height`	Size of the plot in inches. Default is 8 by 5.
`res`	Resolution in dots per inch, for the pixel formats. Default is 300.

Value

The path of the file, invisibly.

save_plots *Save plots to files*

Description

Writes the plots of a summarizeSCI() or summarizeByVisitSCI() result, or a single plot such as res$plots$hb, each into its own file.

Usage

```
save_plots(
  x,
  file_name = NULL,
  file_path = ".",
  format = "png",
  width = 8,
  height = 5,
  res = 300
)
```

Arguments

x	The result of summarizeSCI() or summarizeByVisitSCI() (made with type = "both" or "plot"), or a single plot, e.g. res$plots$hb.
file_name	Name(s) of the files, without extension. Either one name per plot, or a single name that prefixes the variable names when x holds several plots (e.g. "fig" gives fig_hb.png, fig_hb_cat.png). Default is NULL, which names each file after its variable (hb.png).
file_path	Folder to write the files into. Created if it does not exist. Default is the working directory.
format	File format: "png" (the default), "jpeg", "tiff", "pdf" or "svg".
width, height	Size of the plot in inches. Default is 8 by 5.
res	Resolution in dots per inch, for the pixel formats. Default is 300.

Value

The paths of the files, invisibly.

Examples

```
## Not run:
save_plots(res)                                    # hb.png, hb_cat.png
save_plots(res, file_name = "fig", file_path = "figures", format = "pdf")
save_plots(res$plots$hb, file_name = "hemoglobin")  # hemoglobin.png

## End(Not run)
```

sci_plot *Plot objects*

Description

The plotting functions of the package draw with base R graphics, which produce no object to keep. The wrappers `summarizeSCI()` therefore store each plot as a `sci_plot`: the instructions to draw it, together with the numbers it shows. Printing a `sci_plot` draws it on the current device, and `save_plot()` draws it into a file of the requested size, so that the layout is always worked out for the size it is drawn at.

Usage

```
## S3 method for class 'sci_plot'
print(x, ...)

## S3 method for class 'sci_plot'
plot(x, ...)
```

Arguments

`x`	A `sci_plot` object.
`...`	Ignored.

Value

`print()` and `plot()` return `x` invisibly.

stackedbarplotSCI *Stacked barplot over time*

Description

Generates stacked barplots of a categorical variable over time for one or multiple sub-groups. Each bar shows the distribution of the levels of `var` at one time point in one group, optionally with the percentages printed inside the segments and a table of patient numbers beneath the horizontal axis.

Usage

```
stackedbarplotSCI(
  data,
  var,
  order = NULL,
  visit = NULL,
  group = NULL,
```

```
  id = NULL,
  group_names = NULL,
  group_names_short = NULL,
  level_names = NULL,
  scale = c("percent", "count"),
  missing = FALSE,
  missing_text = "Missing",
  main = "",
  xlab = NULL,
  ylab = NULL,
  ylim = NULL,
  y_ticks = NULL,
  col = NULL,
  greyscale = FALSE,
  percent_labels = TRUE,
  col_percent = NULL,
  cex_percent = NULL,
  digits = 0,
  bar_width = NULL,
  cex_lab = 1,
  cex_axis = 1,
  cex_n_patients = 0.7,
  cex_groups = NULL,
  risk_table = FALSE,
  missing_nr = 0,
  rotate_labels = FALSE,
  mar_custom = NULL,
  xlab_line = NULL,
  ylab_line = 3,
  xlab_position = NULL,
  legend_position = "top",
  legend_title = NULL,
  legend_offset = c(0, 0),
  legend_cex = 1,
  legend_bty = "n",
  stat_footnote = FALSE,
  ...
)
```

Arguments

`data`	A data frame or tibble containing the data to be plotted.
`var`	Name of the column(s) in `data` to be stacked. A single column supports bare or quoted names; several columns are given as a character vector.
`order`	Name of the column in `data` containing the time points of interest. Numeric time points are sorted numerically; factors keep their level order; character values are sorted by the number they contain (e.g. `"Visit 2"` before `"Visit 10"`), or by order of appearance if they contain none. If left empty, a single time point is

assumed and one bar is drawn per group. Rows with a missing time point are dropped.

`visit` Name of the column in `data` containing descriptive labels for the time points (the first label observed at each time point is used). Optional.

`group` Name of the column in `data` containing the grouping variable (e.g., treatment arms). If left empty, a single group is assumed. Rows with a missing group are dropped.

`id` Name of the column in `data` identifying the patient. Only used when `var` names several Yes/No columns, as the denominator of the percentages. Default is `NULL`, which counts rows instead.

`group_names` A character vector containing the names of the groups to be displayed on the x-axis. The order must match `levels(factor(data$group))`. Default is `NULL`, which falls back to the group names found in the data.

`group_names_short`
A character vector containing short names or abbreviations for the groups, displayed beneath the bars and in the table of patient numbers. Optional. Defaults to `group_names`.

`level_names` A character vector containing the names of the stacked segments to be displayed in the legend. The order must match the levels of `var` (or the columns in `var`). Default is `NULL`, which uses the levels (or column names) themselves.

`scale` The quantity on the y-axis. Either `"percent"` (the default) or `"count"`.

`missing` A logical toggle switch adding missing values of `var` as a segment of their own, on top of the stack, hatched in grey on white rather than coloured. Default is `FALSE`, which drops them. Only used when `var` is a single column.

`missing_text` Legend text for the missing-value segment. Default is `"Missing"`.

`main` Title of the plot. Defaults to `""` (no title).

`xlab` A label for the x-axis. Default is `NULL`, which uses `"Time"` if there are several time points, otherwise `"Group"`.

`ylab` A label for the y-axis. Default is `NULL`, which uses `"Percentage of patients"` or `"Number of patients"` depending on `scale`.

`ylim` The y-axis limits of the plot as a numeric vector of length 2. Default is `NULL`, which spans `0` to `100` for percentages (more if the stacks exceed 100%) and `0` to the tallest stack for counts.

`y_ticks` A numeric vector specifying the explicit tick positions on the y-axis. Default is `NULL`, which falls back to automatic tick placement.

`col` A character vector of hex colors or color names to use for the stacked segments, bottom to top. Recycled if shorter than the number of segments. The missing-value segment (see `missing`) is always hatched and takes no color. Default is `NULL`, which applies the SCI color palette without its lightest shade.

`greyscale` A logical toggle switch replacing the color palette by a sequential grey ramp, for print or black-and-white reproduction. Default is `FALSE`. Ignored if `col` is supplied.

`percent_labels` A logical toggle switch enabling the percentage labels printed inside the segments. Default is `TRUE`. Empty segments, and segments too thin to hold a label, are left unlabelled.

col_percent — A character value giving the color of the percentage labels. Default is NULL, which picks black or white per segment, whichever contrasts better with the fill.

cex_percent — A numeric value specifying the font size of the percentage labels. Default is NULL, which sizes them to fit the width of the bars (at most 0.8). A number is used as given, even if labels then overflow narrow bars.

digits — A numeric value indicating the number of decimal places shown in the percentage labels. Default is 0.

bar_width — A numeric value giving the width of each bar in x-axis units. Default is NULL, which fills about 70% of each time point with the cluster of group bars.

cex_lab — A numeric value specifying the font size of the main axis labels. Default is 1.

cex_axis — A numeric value specifying the font size of the axis tick marks. Default is 1.

cex_n_patients — A numeric value specifying the font size of the table of patient numbers. Default is 0.7.

cex_groups — A numeric value specifying the font size of the group labels beneath the bars (several groups over time). Default is NULL, which uses cex_n_patients and shrinks the labels until they fit the bars. A number is used as given, even if long labels then overlap.

risk_table — A logical toggle switch enabling the table of patient numbers (the denominator of each bar) beneath the x-axis. Default is FALSE.

missing_nr — A numeric or character value indicating how zero-count patient sizes on the x-axis should be displayed. Default is 0. Can also be set to NA, which leaves the cell blank.

rotate_labels — A logical toggle switch rotating the x-axis tick labels by 90 degrees. Default is FALSE.

mar_custom — A numeric vector indicating the plot margins to be used in the form c(bottom, left, top, right). Default is NULL, which dynamically calculates spacing for the labels, the table of patient numbers and the legend.

xlab_line — A numeric value indicating the margin line offset for the x-axis label. Default is NULL, which calculates an adaptive offset.

ylab_line — A numeric value indicating the margin line offset for the y-axis label. Default is 3.0.

xlab_position — Horizontal position of the row labels of the table of patient numbers. Default is NULL, which aligns them just outside the plot region.

legend_position — Position of the legend box. "top" (the default) places it in one row above the bars, which keeps their full width; "right" places it outside the plot region to the right of the bars; "none" or NULL suppresses it. Other base R keywords (e.g., "topright") place it inside the plot region, and a numeric pair c(x, y) gives explicit coordinates. The margins are sized to the legend text.

legend_title — Title of the legend box. Default is NULL (no title).

legend_offset — A numeric vector of length 2 specifying an inset to nudge a keyword-placed legend. Default is c(0, 0).

legend_cex — A numeric value specifying the font and element scaling size inside the legend box. Default is 1.

legend_bty	The type of box border to be drawn around the legend. Allowed values are "n" (no border, the default) and "o" (standard box frame).
stat_footnote	Show what the bars display as a footnote in the bottom-right corner, beneath the x-axis title: TRUE for the default text (e.g. "\% of patients with a non-missing value"), a character string for a text of your own, or FALSE (the default) for none.
...	Further graphical parameters passed on to plot().

Details

Two layouts of the data are supported:

- var names **one** categorical column (long format): its levels are stacked, and percentages are relative to the number of non-missing observations in the bar (or all observations if missing = TRUE).
- var names **several** columns coded "Yes" / "No" (wide format, e.g. one column per adverse event): each column becomes one segment holding the number of "Yes" answers, and percentages are relative to the number of patients in the bar (id, or rows if id is NULL). Missing values count as "No". Since a patient may answer "Yes" to several columns, the segments can add up to more than 100%.

Value

Invisibly, a list with counts and percent, arrays indexed by segment, group and time point, and n, the group-by-time matrix of denominators.

Examples

```
library(summarySCI)
data("hemoglobin_data")
hemoglobin_data$hb_cat <- cut(hemoglobin_data$hb, c(-Inf, 120, 140, Inf),
                              labels = c("< 120", "120-140", "> 140"))
stackedbarplotSCI(data = hemoglobin_data,
                  var = hb_cat,
                  order = visit_nr,
                  visit = visit,
                  group = Arm,
                  group_names_short = c("A", "B"),
                  legend_title = "Hemoglobin",
                  risk_table = TRUE)

# Several Yes/No columns, one bar per arm
set.seed(14)
ae <- data.frame(id = 1:50,
                 arm = sample(c("Arm A", "Arm B"), 50, replace = TRUE),
                 fatigue = sample(c("Yes", "No"), 50, replace = TRUE),
                 nausea  = sample(c("Yes", "No"), 50, replace = TRUE))
stackedbarplotSCI(ae, var = c("fatigue", "nausea"), group = arm, id = id,
                  scale = "count", legend_title = "Adverse events")
```

summarizeByVisitSCI *Summarize data by visit as a table and plots*

Description

Produces the table of summaryByVisit() and one plot per variable (or either of them alone, see type), the plots over the visits: a line plot or boxplots (boxplotSCI()) for continuous variables, and stacked barplots for categorical ones. The type of each plot follows the type the variable has in the table.

Usage

```
summarizeByVisitSCI(
  data,
  vars = NULL,
  group = NULL,
  labels = NULL,
  stat_cont = "median_range",
  stat_cat = "n_percent",
  visit = "visit",
  order = NULL,
  visitgroup = NULL,
  missing = TRUE,
  missing_percent = TRUE,
  missing_text = "Missing",
  continuous_as_categorical = NULL,
  type = c("both", "table", "plot"),
  plot_cont = c("lineplot", "boxplot"),
  plot_args = list(),
  as_flex_table = TRUE,
  border = TRUE,
  word_output = FALSE,
  file_path = paste0("SummaryByVisit_", format(Sys.Date(), "%Y%m%d"), ".docx"),
  ...
)
```

Arguments

data	A data frame or tibble containing the data to be summarized.
vars	Variables to include in the summary table. Need to be specified with quotes, e.g. "age" or c("age", "response"). Default to all variables present in the data except group.
group	A single column from data. Need to be specified with quotes, e.g. "treatment". Summary statistics will be stratified according to this variable. Default to NULL. A maximum of 3 groups are currently supported. Character variables and a character group are shown in natural order (e.g. "2" before "10"); factors keep their level order.

labels	A list containing the labels that should be used for the variables in the table, either named by variable (e.g. list(age = "Age (years)")) or unnamed in the order of vars. If NULL, labels are automatically taken from the dataset. If no label present, the variable name is taken.
stat_cont	Summary statistic to display for continuous variables. Options include "median_IQR", "median_range" (default), "mean_sd", "mean_se" and "geomMean_sd".
stat_cat	Summary statistic to display for categorical variables. Options include "n", "n_N" and "n_percent" (default).
visit	Name of the stratum for which summary statistics are displayed by line. Typically, this would be "visit".
order	A numerical variable defining the visit order. If NULL (default), visits are sorted naturally (e.g. "Visit 2" before "Visit 10"); if visit is a factor, its level order is used instead.
visitgroup	A grouping variable for the stratum for which summary statistics are displayed by line. Must be an ordered factor. Typically, this would be a visit group such as e.g., baseline, follow-up etc.
missing	Logical. If TRUE (default), the missing values are shown (only for categorical variables).
missing_percent	Indicates whether percentages for missings are shown (TRUE, default) or not (FALSE) for categorical variables. If "both", then both options are displayed next to each other.
missing_text	String indicating text shown on missing row. Default to "Missing".
continuous_as_categorical	A subset (vector) of continuous vars to be treated as categrical. Default is NULL.
type	What to produce: "both" (the default) the table and a plot per variable, "table" the table alone, or "plot" the plots alone. The table is built in every case, since it decides the type of each plot.
plot_cont	The plot for continuous variables: "lineplot" (the default) or "boxplot".
plot_args	A named list of further arguments for the plotting functions, e.g. list(risk_table = TRUE, dodge = 0.1). Each plot receives the arguments its function has, as in lineplotSCI(), boxplotSCI() and stackedbarplotSCI(), except those that compute the numbers (such as stat or interval), which come from the table; an argument no plot uses gives a warning. For example, stat_footnote = TRUE adds a footnote saying what each plot shows: for the line plot the statistic of stat_cont (e.g. "Median (Q1, Q3)").
as_flex_table	Logical. If TRUE (default) the gtsummary object is converted to a flextable object. Useful when rendering to Word.
border	Logical. If TRUE, a border will be drawn around the table. Only available if flex_table = TRUE. Default is TRUE.
word_output	Logical. If TRUE, the table is also saved in a word document.
file_path	Character string. Specify the path of the Word document containing the table. Only used when word_output is TRUE. Needs to end with ".docx". Default to "SummaryByVisit_YYYYMMDD.docx" (today's date) in the working directory.

... Further arguments passed on to summaryByVisit(), e.g. digits_cont, add_n or overall.

Details

The line plots and stacked bars are drawn from the numbers printed in the table, as plotByVisit() does, so that they show exactly what the table shows (with its rounding):

- stat_cont sets what the line plot shows: "median_range" the median with minimum and maximum, "median_IQR" the median with the quartiles, "mean_sd" / "mean_se" the mean +/- one standard deviation / standard error, and "geomMean_sd" the geometric mean divided and multiplied by the geometric standard deviation;
- the variable labels become the axis or legend titles, the column headers the group names, and the N columns the numbers of the risk table;
- missing / missing_percent decide whether missing values form a segment of their own (as they are counted in the table's percentages), and stat_cat = "n" shows counts instead of percentages;
- continuous_as_categorical variables are plotted as stacked bars, and with overall = TRUE the overall column is plotted as one more group.

Boxplots show more than the table holds, and are drawn from the data: from the rows the table summarizes, with the box at the quartiles of the table (see boxplotSCI()).

The visits appear in the order of the table. If order is given, its values also set the spacing of the visits in the line plots.

Value

An object of class summarizeByVisitSCI (and summarizeSCI), a list with table and plots; see summarizeSCI().

Examples

```
library(summarySCI)
data("hemoglobin_data")
hb <- subset(hemoglobin_data, visit_nr <= 2)   # first 2 cycles, to keep it short
res <- summarizeByVisitSCI(hb, vars = "hb",
                           group = "Arm", visit = "visit", order = "visit_nr",
                           stat_cont = "median_IQR")
res$plots$hb


# a categorical variable
hb$hb_cat <- cut(hb$hb, c(-Inf, 120, 140, Inf),
                 labels = c("< 120", "120-140", "> 140"))
res <- summarizeByVisitSCI(hb, vars = c("hb", "hb_cat"),
                           group = "Arm", visit = "visit", order = "visit_nr",
                           stat_cont = "median_IQR")
res$plots$hb_cat
```

summarizeSCI *Summarize data as a table and plots*

Description

Produces the summary table of summaryTable() and one plot per variable, or either of them alone (see type): a boxplot for continuous variables (boxplotSCI()) and a stacked barplot for categorical and dichotomous ones (stackedbarplotSCI()). The type of each plot follows the type the variable has in the table, so that table and plots always agree.

Usage

```
summarizeSCI(
  data,
  vars = NULL,
  group = NULL,
  labels = NULL,
  stat_cont = "median_range",
  stat_cat = "n_percent",
  missing = TRUE,
  missing_percent = TRUE,
  missing_text = "Missing",
  type = c("both", "table", "plot"),
  plot_args = list(),
  as_flex_table = TRUE,
  border = TRUE,
  word_output = FALSE,
  file_path = paste0("SummaryTable_", format(Sys.Date(), "%Y%m%d"), ".docx"),
  ...
)
```

Arguments

data	A data frame or tibble containing the data to be summarized.
vars	Variables to include in the summary table. Need to be specified with quotes, e.g. "age" or c("age", "response"). Default to all variables present in the data except group.
group	A single column from data. Need to be specified with quotes, e.g. "treatment". Summary statistics will be stratified according to this variable. Default to NULL.
labels	A list containing the labels that should be used for the variables in the table. If NULL, labels are automatically taken from the dataset. If no label present, the variable name is taken.
stat_cont	Summary statistic to display for continuous variables. Options include "median_IQR", "median_range" (default), "mean_sd", "mean_se" and "geomMean_sd". "geomMean_sd" shows the geometric mean with the geometric standard deviation, exp(sd(log(x))), both NA if a value is zero or negative. The geometric SD is a factor close to 1 and is shown with at least 2 decimals.

stat_cat	Summary statistic to display for categorical variables. Options include "n_percent" (default) and "n", and "n_N".
missing	Logical. If TRUE (default), the missing values are shown.
missing_percent	Indicates whether percentages for missings are shown (TRUE, default) or not (FALSE) for categorical variables. If "both", then both options are displayed next to each other.
missing_text	String indicating text shown on missing row. Default to "Missing".
type	What to produce: "both" (the default) the table and a plot per variable, "table" the table alone, or "plot" the plots alone. The table is built in every case, since it decides the type of each plot.
plot_args	A named list of further arguments for the plotting functions, e.g. list(risk_table = TRUE, col = c("grey", "black")). Each plot receives the arguments its function has, as in boxplotSCI() and stackedbarplotSCI(); an argument no plot uses gives a warning. For example, stat_footnote = TRUE adds a footnote saying what each plot shows (the box and whiskers, or the denominator of the percentages).
as_flex_table	Logical. If TRUE (default) the gtsummary object is converted to a flextable object. Useful when rendering to Word.
border	Logical. If TRUE, a border will be drawn around the table. Only available if flex_table = TRUE. Default is TRUE.
word_output	Logical. If TRUE, the table is also saved in a word document.
file_path	Character string. Specify the path of the Word document containing the table. Only used when word_output is TRUE. Needs to end with ".docx".
...	Further arguments passed on to summaryTable(), e.g. test, ci, overall or digits_cont.

Details

The plots are drawn from the same rows as the table, and the box of the boxplot spans the quartiles as the table computes them (see boxplotSCI()). The settings of the table carry over to the plots: the variable labels become the axis or legend titles, missing / missing_percent decide whether missing values form a segment of their own (as they are counted in the table's percentages), and stat_cat = "n" shows counts instead of percentages. Options that only concern the plots go into plot_args.

Value

An object of class summarizeSCI, a list with

- table: the table, a flextable or (with as_flex_table = FALSE) a gtsummary object; NULL with type = "plot". With word_output = TRUE it is also saved to file_path, as summaryTable() does;
- plots: a named list with one plot per variable, e.g. res$plots$hb; empty with type = "table". Printing a plot draws it; save_plots() writes them to files.

Printing the result shows the table and lists the plots.

Examples

```
library(summarySCI)
data("hemoglobin_data")
d <- hemoglobin_data[hemoglobin_data$visit_nr == 2, ]
d$hb_cat <- cut(d$hb, c(-Inf, 120, 140, Inf),
                labels = c("< 120", "120-140", "> 140"))
res <- summarizeSCI(d, vars = c("hb", "hb_cat"), group = "Arm")
res$table
res$plots$hb
res$plots$hb_cat


# the plots alone
plots <- summarizeSCI(d, vars = c("hb", "hb_cat"), group = "Arm",
                      type = "plot")$plots

## Not run:
save_plots(res, file_name = "fig", file_path = "figures")

## End(Not run)
```

---

summaryByVisit *Creates publication-ready summary tables for data grouped, by visit*

Description

Creates publication-ready summary tables for data grouped, by visit

Usage

```
summaryByVisit(
  data,
  vars = NULL,
  group = NULL,
  labels = NULL,
  stat_cont = "median_range",
  stat_cat = "n_percent",
  visit = "visit",
  order = NULL,
  visitgroup = NULL,
  digits_cont = 1,
  digits_cat = 0,
  missing_percent = TRUE,
  missing = TRUE,
  missing_text = "Missing",
  add_n = TRUE,
  overall = FALSE,
```

```
  continuous_as_categorical = NULL,
  as_flex_table = TRUE,
  border = TRUE,
  word_output = FALSE,
  file_path = paste0("SummaryByVisit_", format(Sys.Date(), "%Y%m%d"), ".docx")
)
```

Arguments

data	A data frame or tibble containing the data to be summarized.
vars	Variables to include in the summary table. Need to be specified with quotes, e.g. "age" or c("age", "response"). Default to all variables present in the data except group.
group	A single column from data. Need to be specified with quotes, e.g. "treatment". Summary statistics will be stratified according to this variable. Default to NULL. A maximum of 3 groups are currently supported. Character variables and a character group are shown in natural order (e.g. "2" before "10"); factors keep their level order.
labels	A list containing the labels that should be used for the variables in the table, either named by variable (e.g. list(age = "Age (years)")) or unnamed in the order of vars. If NULL, labels are automatically taken from the dataset. If no label present, the variable name is taken.
stat_cont	Summary statistic to display for continuous variables. Options include "median_IQR", "median_range" (default), "mean_sd", "mean_se" and "geomMean_sd".
stat_cat	Summary statistic to display for categorical variables. Options include "n", "n_N" and "n_percent" (default).
visit	Name of the stratum for which summary statistics are displayed by line. Typically, this would be "visit".
order	A numerical variable defining the visit order. If NULL (default), visits are sorted naturally (e.g. "Visit 2" before "Visit 10"); if visit is a factor, its level order is used instead.
visitgroup	A grouping variable for the stratum for which summary statistics are displayed by line. Must be an ordered factor. Typically, this would be a visit group such as e.g., baseline, follow-up etc.
digits_cont	Digits for summary statistics and CI of continuous variables. Default to 1.
digits_cat	Digits for summary statistics and CI of categorical variables. Default to 0.
missing_percent	Indicates whether percentages for missings are shown (TRUE, default) or not (FALSE) for categorical variables. If "both", then both options are displayed next to each other.
missing	Logical. If TRUE (default), the missing values are shown (only for categorical variables).
missing_text	String indicating text shown on missing row. Default to "Missing".
add_n	Logical. If TRUE, an additional column with the total number of non-missing observations for each variable is added. Default to TRUE.

overall	Logical. If TRUE, an additional column with the total is added to the table. Ignored, if no groups are defined. Default to FALSE.
continuous_as_categorical	A subset (vector) of continuous vars to be treated as categrical. Default is NULL.
as_flex_table	Logical. If TRUE (default) the gtsummary object is converted to a flextable object. Useful when rendering to Word.
border	Logical. If TRUE, a border will be drawn around the table. Only available if flex_table = TRUE. Default is TRUE.
word_output	Logical. If TRUE, the table is also saved in a word document.
file_path	Character string. Specify the path of the Word document containing the table. Only used when word_output is TRUE. Needs to end with ".docx". Default to "SummaryByVisit_YYYYMMDD.docx" (today's date) in the working directory.

Value

A table of class "flextable" or c("tbl_strata_nested_stack", "tbl_stack", "gtsummary"). Optionally returns a .docx file in the specified folder.

Examples

```
data("hemoglobin_data")
hb <- subset(hemoglobin_data, visit_nr <= 3)   # first 3 cycles, to keep it short

# Hemoglobin by visit and arm, with an overall column
summaryByVisit(data = hb, vars = "hb", visit = "visit",
               group = "Arm", overall = TRUE,
               labels = list(hb = "Hemoglobin (g/L)"))
```

---

summaryByVisitCategorical

Creates publication-ready summary tables for categorical data grouped, by visit

---

Description

Creates publication-ready summary tables for categorical data grouped, by visit

Usage

```
summaryByVisitCategorical(
  data,
  vars = NULL,
  group = NULL,
  labels = NULL,
```

```
  stat_cat = "n_percent",
  visit = "visit",
  order = NULL,
  visitgroup = NULL,
  digits_cat = 0,
  missing_percent = TRUE,
  missing = TRUE,
  missing_text = "Missing",
  add_n = TRUE,
  overall = FALSE,
  as_flex_table = FALSE,
  border = TRUE,
  word_output = FALSE,
  file_path = paste0("SummaryByVisit_", format(Sys.Date(), "%Y%m%d"), ".docx")
)
```

Arguments

data	A data frame or tibble containing the data to be summarized.
vars	Continuous variables to include in the summary table. Need to be specified with quotes, e.g. "grade" or c("grade", "stage"). Default to all variables present in the data except group.
group	A single column from data. Need to be specified with quotes, e.g. "treatment". Summary statistics will be stratified according to this variable. Default to NULL. A maximum of 3 groups are currently supported.
labels	A list containing the labels that should be used for the variables in the table. If NULL, labels are automatically taken from the dataset. If no label present, the variable name is taken.
stat_cat	Summary statistic to display for categorical variables. Options include "n", "n_N" and "n_percent" (default).
visit	Name of the stratum for which summary statistics are displayed by line. Typically, this would be "visit".
order	A numerical variable defining the visit order.
visitgroup	A grouping variable for the stratum for which summary statistics are displayed by line. Must be an ordered factor. Typically, this would be a visit group such as e.g., baseline, follow-up etc.
digits_cat	Digits for summary statistics of categorical variables. Default to 0.
missing_percent	Indicates whether percentages for missings are shown (TRUE, default) or not (FALSE) for categorical variables. If "both", then both options are displayed next to each other.
missing	Logical. If TRUE (default), the missing values are shown.
missing_text	String indicating text shown on missing row. Default to "Missing".
add_n	Logical. If TRUE, an additional column with the total number of non-missing observations for each variable is added. Default to TRUE.

overall	Logical. If TRUE, an additional column with the total is added to the table. Ignored, if no groups are defined. Default to FALSE.
as_flex_table	Logical. If TRUE, the gtsummary object is converted to a flextable object. Useful when rendering to Word. Default to FALSE.
border	Logical. If TRUE, a border will be drawn around the table. Only available if flex_table = TRUE. Default is TRUE.
word_output	Logical. If TRUE, the table is also saved in a word document.
file_path	Character string. Specify the path of the Word document containing the table. Only used when word_output is TRUE. Needs to end with ".docx".

Value

A table of class "flextable" or c("tbl_strata_nested_stack", "tbl_stack", "gtsummary"). Optionally returns a .docx file in the specified folder.

---

summaryByVisitContinuous

Creates publication-ready summary tables for continuous data grouped, by visit

---

Description

Creates publication-ready summary tables for continuous data grouped, by visit

Usage

```
summaryByVisitContinuous(
  data,
  vars = NULL,
  group = NULL,
  labels = NULL,
  stat_cont = "median_range",
  visit = "visit",
  order = NULL,
  visitgroup = NULL,
  digits_cont = 1,
  add_n = TRUE,
  overall = FALSE,
  as_flex_table = FALSE,
  border = TRUE,
  word_output = FALSE,
  file_path = paste0("SummaryByVisit_", format(Sys.Date(), "%Y%m%d"), ".docx")
)
```

Arguments

data	A data frame or tibble containing the data to be summarized.
vars	Continuous variables to include in the summary table. Need to be specified with quotes, e.g. "age" or c("age", "response"). Default to all variables present in the data except group.
group	A single column from data. Need to be specified with quotes, e.g. "treatment". Summary statistics will be stratified according to this variable. Default to NULL. A maximum of 3 groups are currently supported.
labels	A list containing the labels that should be used for the variables in the table. If NULL, labels are automatically taken from the dataset. If no label present, the variable name is taken.
stat_cont	Summary statistic to display for continuous variables. Options include "median_IQR", "median_range" (default), "mean_sd", "mean_se" and "geomMean_sd". "geomMean_sd" shows the geometric mean with the geometric standard deviation, exp(sd(log(x))), both NA if a value is zero or negative. The geometric SD is a factor close to 1 and is shown with at least 2 decimals.
visit	Name of the stratum for which summary statistics are displayed by line. Typically, this would be "visit".
order	A numerical variable defining the visit order.
visitgroup	A grouping variable for the stratum for which summary statistics are displayed by line. Must be an ordered factor. Typically, this would be a visit group such as e.g., baseline, follow-up etc.
digits_cont	Digits for summary statistics and CI of continuous variables. Default to 1.
add_n	Logical. If TRUE, an additional column with the total number of non-missing observations for each variable is added. Default to TRUE.
overall	Logical. If TRUE, an additional column with the total is added to the table. Ignored, if no groups are defined. Default to FALSE.
as_flex_table	Logical. If TRUE, the gtsummary object is converted to a flextable object. Useful when rendering to Word. Default to FALSE.
border	Logical. If TRUE, a border will be drawn around the table. Only available if flex_table = TRUE. Default is TRUE.
word_output	Logical. If TRUE, the table is also saved in a word document.
file_path	Character string. Specify the path of the Word document containing the table. Only used when word_output is TRUE. Needs to end with ".docx".

Value

A table of class "flextable" or c("tbl_strata_nested_stack", "tbl_stack", "gtsummary"). Optionally returns a .docx file in the specified folder.

summaryLevels *summaryLevels*

Description

Collapses factor levels from multiple columns into one and creates summary table.

Usage

```
summaryLevels(
  data,
  vars = NULL,
  group = NULL,
  label = NULL,
  levels = NULL,
  stat_cat = "n_percent",
  test = FALSE,
  test_cat = "fisher.test",
  ci = FALSE,
  ci_cat = "clopper.pearson",
  conf_level = 0.95,
  digits_cat = 0,
  overall = FALSE,
  add_n = TRUE,
  as_flex_table = TRUE,
  border = TRUE,
  word_output = FALSE,
  file_path = paste0("SummaryLevels_", format(Sys.Date(), "%Y%m%d"), ".docx")
)
```

Arguments

data	A data frame or tibble containing the data to be summarized.
vars	Variables to include in the summary table. Need to be specified with quotes, e.g. "score" or c("score", "age_cat"). Default to all variables present in the data except group.
group	A single column from data. Need to be specified with quotes, e.g. "treatment". Summary statistics will be stratified according to this variable. Any number of groups (at least two) is possible. Rows with a missing value in group are removed. Default to NULL.
label	A label for the new variable to be created. If no label present, the variable name is taken.
levels	= A vector containing the values indicating presence of the factor level. Included by default are "1", "yes", "Yes". All variables in vars must be binary (e.g. 0/1, yes/no, TRUE/FALSE), unless levels is given; otherwise an error is returned.

stat_cat	Summary statistic to display for categorical variables. Options include "n_percent" (default) and "n", and "n_N".
test	Logical. Indicates whether p-values are displayed (TRUE) or not (FALSE). Default to FALSE
test_cat	Test type used to calculated the p-value for categorical variables. Only used if test = TRUE. Options include "fisher.test" (default), "chisq.test", "chisq.test.no.correct". If NULL, the function decides itself: "chisq.test.no.correct" for categorical variables with all expected cell counts >=5, and "fisher.test" for categorical variables with any expected cell count <5.
ci	Logical. Indicates whether CI are displayed (TRUE) or not (FALSE). Default to FALSE.
ci_cat	Confidence interval method for categorical variables. Options include "clopper.pearson" (default), "wilson", "wilson.no.correct", "wald", "wald.no.correct", "agresti.coull" and "jeffreys". If NULL, no CI will be displayed.
conf_level	Numeric. Confidence level. Default to 0.95.
digits_cat	Numeric. Digits for summary statistics and CI of categorical variables. Default to 0.
overall	Logical. If TRUE, an additional column with the total is added at the end of the table. Default to FALSE.
add_n	Logical. If TRUE (default), a column with the number of patients is added before each group column (and before the Overall column). As missing values are treated as absent, N is the number of patients in each group.
as_flex_table	Logical. If TRUE (default) the gtsummary object is converted to a flextable object. Useful when rendering to Word.
border	Logical. If TRUE, a border will be drawn around the table. Only available if flex_table = TRUE. Default is TRUE.
word_output	Logical. If TRUE, the table is also saved in a word document.
file_path	Character string. Specify the path of the Word document containing the table. Only used when word_output is TRUE. Needs to end with ".docx".

Value

A table of class "flextable" or c("tbl_stack", "gtsummary"). Optionally returns a .docx file in the specified folder.

Examples

```
# Site of progression: one column per site, more than one site possible
progression <- data.frame(
  arm   = rep(c("A", "B"), each = 5),
  liver = c("absent", "present", "absent", "present", "absent",
            "absent", "present", "absent", "present", "absent"),
  lung  = c("absent", "absent", "present", "absent", "absent",
            "absent", "absent", "absent", "absent", "absent"),
  brain = c("present", "absent", "present", "present", "present",
            "present", "present", "present", "present", "present")
```

```
)

summaryLevels(data = progression,
              vars = c("liver", "lung", "brain"),
              label = "Site of progression",
              levels = "present")

# By arm, with an overall column
summaryLevels(data = progression,
              vars = c("liver", "lung", "brain"),
              group = "arm",
              overall = TRUE,
              label = "Site of progression",
              levels = "present")
```

---

summaryTable *Creates publication-ready summary tables*

Description

Creates publication-ready summary tables based on the gtsummary package.

Usage

```
summaryTable(
  data,
  vars = NULL,
  group = NULL,
  labels = NULL,
  stat_cont = "median_range",
  stat_cat = "n_percent",
  continuous_as = "continuous",
  dichotomous_as = "dichotomous",
  continuous_as_categorical = NULL,
  dichotomous_as_categorical = NULL,
  ref_level = NULL,
  test = FALSE,
  test_cont = NULL,
  test_cat = "fisher.test",
  ci = FALSE,
  ci_cont = "wilcox.test",
  ci_cat = "clopper.pearson",
  conf_level = 0.95,
  digits_cont = 1,
  digits_cat = 0,
  missing = TRUE,
  missing_percent = TRUE,
```

```
  missing_text = "Missing",
  overall = FALSE,
  add_n = TRUE,
  as_flex_table = TRUE,
  border = TRUE,
  word_output = FALSE,
  file_path = paste0("SummaryTable_", format(Sys.Date(), "%Y%m%d"), ".docx")
)
```

Arguments

data	A data frame or tibble containing the data to be summarized.
vars	Variables to include in the summary table. Need to be specified with quotes, e.g. "age" or c("age", "response"). Default to all variables present in the data except group.
group	A single column from data. Need to be specified with quotes, e.g. "treatment". Summary statistics will be stratified according to this variable. Default to NULL.
labels	A list containing the labels that should be used for the variables in the table. If NULL, labels are automatically taken from the dataset. If no label present, the variable name is taken.
stat_cont	Summary statistic to display for continuous variables. Options include "median_IQR", "median_range" (default), "mean_sd", "mean_se" and "geomMean_sd". "geomMean_sd" shows the geometric mean with the geometric standard deviation, exp(sd(log(x))), both NA if a value is zero or negative. The geometric SD is a factor close to 1 and is shown with at least 2 decimals.
stat_cat	Summary statistic to display for categorical variables. Options include "n_percent" (default) and "n", and "n_N".
continuous_as	Type for all the continuous variables. Can either be "continuous" (default) or "categorical".
dichotomous_as	Type for the dichotomous variables. Can either be "categorical" (one row per level) or "dichotomous" (only one row with reference level (see argument ref_level), only works if the variable does not contain missing values, or if missing = "FALSE" or missing_percent = FALSE.
continuous_as_categorical	A subset (vector) of continuous vars to be treated as categorical. Default is NULL.
dichotomous_as_categorical	A subset (vector) of dichotomous vars to be treated as categorical. Default is NULL.
ref_level	Specifies the reference level of a variable to display on a single row. Default is the first appearing level. The syntax is as follows: ref_level = list(varname ~ "level to show").
test	Logical. Indicates whether p-values are displayed (TRUE) or not (FALSE). Default to FALSE
test_cont	Test type used to calculate the p-value for continuous variables. Only used if test = TRUE. Options include "t.test", "oneway.test", "kruskal.test" (default for more than two groups), "wilcox.test" (default for two groups).

test_cat	Test type used to calculated the p-value for categorical variables. Only used if test = TRUE. Options include "fisher.test" (default), "chisq.test", "chisq.test.no.correct". For "fisher.test", if the exact algorithm runs out of workspace (FEXACT error, e.g. for variables with many levels), the workspace is increased and, if still insufficient, a Monte Carlo p-value (B = 100,000) is reported instead; this is indicated in the table footnote. It varies slightly between runs unless set.seed() is called before. If NULL, the function decides itself: "chisq.test.no.correct" for categorical variables with all expected cell counts >=5, and "fisher.test" for categorical variables with any expected cell count <5.
ci	Logical. Indicates whether CI are displayed (TRUE) or not (FALSE). Default to FALSE.
ci_cont	Confidence interval method for continuous variables. Only used if ci = TRUE. Options include "t.test" and "wilcox.test" (default).
ci_cat	Confidence interval method for categorical variables. Options include "clopper.pearson" (default) "wilson", "wilson.no.correct", "wald", "wald.no.correct", "agresti.coull" and "jeffreys". If NULL, no CI will be displayed.
conf_level	Numeric. Confidence level. Default to 0.95.
digits_cont	Numeric. Digits for summary statistics and CI of continuous variables. Default to 1.
digits_cat	Numeric. Digits for summary statistics and CI of categorical variables. Default to 0.
missing	Logical. If TRUE (default), the missing values are shown.
missing_percent	Indicates whether percentages for missings are shown (TRUE, default) or not (FALSE) for categorical variables. If "both", then both options are displayed next to each other.
missing_text	String indicating text shown on missing row. Default to "Missing".
overall	Logical. If TRUE, an additional column with the total is added to the table. Default to FALSE.
add_n	Logical. If TRUE (default), an additional column with the total number of non-missing observations for each variable is added.
as_flex_table	Logical. If TRUE (default) the gtsummary object is converted to a flextable object. Useful when rendering to Word.
border	Logical. If TRUE, a border will be drawn around the table. Only available if flex_table = TRUE. Default is TRUE.
word_output	Logical. If TRUE, the table is also saved in a word document.
file_path	Character string. Specify the path of the Word document containing the table. Only used when word_output is TRUE. Needs to end with ".docx".

Details

Character variables (and a character group) are converted to factors with natural level order: values starting with a number are sorted numerically (e.g. "3", "10", "more than 10"), other values alphabetically with embedded numbers compared numerically (e.g. "Arm 2" before "Arm 10"). To control the order fully, supply a factor with the desired levels.

Value

A table of class "flextable" or c("tbl_summary", "gtsummary"). Optionally returns a .docx file in the specified folder.

Examples

```
library(survival)
data("cancer")
summaryTable(data = cancer,vars = c("inst", "time","age", "ph.ecog"),
             labels = list(inst = "Institution code",
                           time = "Time",
                           age = "Age",
                           ph.ecog = "ECOG score"))
```

---

tidySummaryByVisit *Read the numbers off a summaryByVisit table*

Description

Converts a table produced by summaryByVisit() into a data frame with one row per variable, visit, group and (for categorical variables) level, holding the parsed numbers. This is the parser behind plotByVisit().

Usage

```
tidySummaryByVisit(tbl, stat_cont = NULL)
```

Arguments

tbl	A table returned by summaryByVisit() with as_flex_table = FALSE (the default TRUE returns a flextable, from which the numbers cannot be read back).
stat_cont	The statistic of continuous variables, as passed to summaryByVisit(). Default is NULL, which reads it from the table footnote; if the footnote is not recognised, plotByVisit() stops and asks for stat_cont rather than guess. A stat_cont that does not fit the printed cells (e.g. "mean_sd" for a median (range) table) is an error too.

Value

A data frame with columns variable (the label of the header row), var_type, visitgroup, visit, visit_id (position of the visit within the variable), group, level (NA for continuous variables), stat (the cell as printed), n (the number of non-missing observations shown in the table's N column), and the parsed numbers centre, lower, upper (continuous) and count (categorical).

Index

∗ **datasets**
    hemoglobin_data, 6

boxplotSCI, 2
boxplotSCI(), *19–23*

hemoglobin_data, 6

lineplotSCI, 6
lineplotSCI(), *10, 11*, *20*

plot.sci_plot *(*sci_plot*)*, 14
plotByVisit, 10
plotByVisit(), *21*, *35*
print.sci_plot *(*sci_plot*)*, 14

save_plot, 12
save_plot(), *14*
save_plots, 13
save_plots(), *23*
sci_plot, 14
stackedbarplotSCI, 14
stackedbarplotSCI(), *10, 11*, *20*, *22, 23*
summarizeByVisitSCI, 19
summarizeByVisitSCI(), *13*
summarizeSCI, 22
summarizeSCI(), *12–14*, *21*
summaryByVisit, 24
summaryByVisit(), *5*, *10, 11*, *19*, *21*, *35*
summaryByVisitCategorical, 26
summaryByVisitContinuous, 28
summaryLevels, 30
summaryTable, 32
summaryTable(), *5*, *22, 23*

tidySummaryByVisit, 35
tidySummaryByVisit(), *11*