Package: stemtools (via r-universe)

November 10, 2026

Title Tools and helpful wrappers for analysts at Stem (https://www.stem.cz)

Version 0.2.2

Description Color palettes, ploting functions and more for working at Stem.

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

URL https://github.com/stem-cz/stemtools, https://stem-cz.github.io/stemtools/

BugReports https://github.com/stem-cz/stemtools/issues

Imports ggplot2 (>= 4.0.0), dplyr, tidyr, rlang, srvyr, tibble, scales, forcats, utils, S7, spreadview, mschart, officer, stats

Remotes alesvomacka/spreadview

Suggests knitr, zip, rmarkdown, testthat (>= 3.0.0), withr

Depends R (>= 4.1.0)

LazyData true

VignetteBuilder knitr

Config/roxygen2/version 8.1.0

Config/testthat/edition 3

Config/pak/sysreqs libfontconfig1-dev libfreetype6-dev libfribidi-dev make libharfbuzz-dev libicu-dev libjpeg-dev libpng-dev libtiff-dev libwebp-dev libxml2-dev libssl-dev zlib1g-dev

Repository https://stem-cz.r-universe.dev

Date/Publication 2026-11-10 21:51:12 UTC

RemoteUrl https://github.com/stem-cz/stemtools

RemoteRef v0.2.2

RemoteSha 95b7704d5032f2707cfc502320da6aa4549ced98

Contents

collapse_cats . . . 2
create_project . . . 3
scale_fill_stem . . . 4
se_mean . . . 5
se_median . . . 5
se_prop . . . 6
select_diverging . . . 6
stem_add_chart . . . 7
stem_as_mschart . . . 8
stem_barplot . . . 10
stem_battery . . . 12
stem_export_docx . . . 13
stem_export_pptx . . . 15
stem_inline . . . 16
stem_multiselect . . . 18
stem_palette . . . 19
stem_palette_gen . . . 20
stem_palettes_all . . . 21
stem_summarise . . . 21
stem_summarise_cat . . . 23
stem_summarise_num . . . 24
theme_stem . . . 25
trust . . . 26

Index **28**

`collapse_cats` *Collapses factor levels and returns a new data frame*

Description

This function is a wrapper around `forcats::fct_collapse()`. It takes a dataframe, a factor variable, and a list of factor levels to collapse.

Usage

```
collapse_cats(data, item, collapse_cats)
```

Arguments

`data`	dataset
`item`	factor variable
`collapse_cats`	list of factor levels to collapse in shape of `list(new_level = old_level)`

Value

new data frame with collapsed factor levels

Examples

```
collapse_cats(trust, police, list(Agree = c("Definitely Agree", "Rather Agree")))
```

`create_project` *Scaffold a new Stem analysis project*

Description

Creates a standard Stem project directory structure, a minimal `README.md` and, optionally, a `.gitignore` file. This function backs the RStudio "New Project" template defined in `inst/rstudio/templates/project/create_proje` but can also be called directly.

Usage

```
create_project(path, createGitignore = TRUE, ...)
```

Arguments

`path`	Path to the project directory to create.
`createGitignore`	If `TRUE` (default), writes a `.gitignore` with sensible defaults for R projects. Also supplied by the RStudio template wizard.
`...`	Additional named arguments collected from the RStudio template wizard. Currently ignored.

Value

Invisibly returns `path`. Called for its side effect of creating the project directory structure on disk.

Examples

```
## Not run:
create_project(file.path(tempdir(), "my-analysis"))

## End(Not run)
```

scale_fill_stem *stem discrete colour and fill scales for ggplot2*

Description

Discrete ggplot2 scales that apply a stem colour palette to the `colour` or `fill` aesthetic. Three aliases are provided:

- `scale_colour_stem()` — maps to the `colour` aesthetic (British spelling).
- `scale_color_stem()` — alias for `scale_colour_stem()` (American spelling).
- `scale_fill_stem()` — maps to the `fill` aesthetic.

All functions delegate to `ggplot2::discrete_scale()` via `stem_palette_gen()`, so any argument accepted by `discrete_scale()` (e.g. `name`, `breaks`, `labels`, `na.value`, `drop`) can be passed through `...`.

Usage

```
scale_fill_stem(palette = "div1", direction = 1, ...)

scale_colour_stem(palette = "div1", direction = 1, ...)

scale_color_stem(palette = "div1", direction = 1, ...)
```

Arguments

`palette`	Name of the palette. See `stem_palette()` for the full list of valid names. Defaults to `"div1"`.
`direction`	`1` (default) uses colours in their natural order; `-1` reverses the order.
`...`	Additional arguments passed to `ggplot2::discrete_scale()`.

Value

A ggplot2 `Scale` object to be added to a plot with `+`.

See Also

`stem_palette()`, `stem_palette_gen()`, `ggplot2::discrete_scale()`

Examples

```
ggplot2::ggplot(data = mtcars,
                mapping = ggplot2::aes(x = mpg, y = hp, color = as.factor(am))) +
  ggplot2::geom_point() +
  scale_colour_stem()
```

se_mean *Computes parametric standard error for mean*

Description

Computes SE for the mean using the old-school $\sqrt{\frac{var_x}{n}}$ formula.

Usage

```
se_mean(x)
```

Arguments

x	numeric vector

Value

numeric value representing SE for means

Examples

```
se_mean(trust$age)
```

se_median *Computes parametric standard error for median*

Description

Computes SE for the median using the old-school $\sqrt{1.2533 \cdot \frac{var_x}{n}}$ formula. This assumes normal distribution.

Usage

```
se_median(x)
```

Arguments

x	numeric vector

Value

numeric value representing SE for medians

Examples

```
se_median(trust$age)
```

se_prop *Computes parametric standard error for proportions*

Description

Computes SE for proportions using the old-school $\sqrt{\frac{p(1-p)}{n}}$ formula.

Usage

```
se_prop(p, n)
```

Arguments

p	proportion
n	sample size

Value

numeric value representing SE for proportions

Examples

```
p = 0.3
n = 200
se_prop(p = p, n = n)
```

select_diverging *Select n colours from a diverging palette*

Description

Selects n colours from a diverging palette while preserving its symmetric, center-anchored structure. Rather than simply taking the first n colours, this function samples symmetrically from both ends of the palette:

- **Odd** n: the palette midpoint is always included, with an equal number of colours drawn from each side.
- **Even** n: colours are drawn evenly from the left and right halves.

The selected colours are always returned in their original palette order.

Usage

```
select_diverging(palette, n)
```

Arguments

palette	Character vector of hex colour codes representing the full diverging palette.
n	Number of colours to select. Must be between 1 and length(palette).

Value

Character vector of n hex colour codes in original palette order.

See Also

stem_palette_gen(), stem_palette()

Examples

```
pal <- stem_palette("modern")

# Odd n: midpoint included, balanced sides
select_diverging(pal, n = 3)

# Even n: two from each end
select_diverging(pal, n = 4)
```

---

stem_add_chart *Add a Stem plot to a PowerPoint or Word document as a native chart*

Description

Converts plot with stem_as_mschart() and adds it to an existing officer document. For a PowerPoint document the chart is placed on the current slide, so add a slide with officer::add_slide() first.

Usage

```
stem_add_chart(
  x,
  plot,
  width = NULL,
  height = NULL,
  left = NULL,
  top = NULL,
  units = "cm",
  location = NULL,
  ...
)
```

Arguments

x	An officer::read_pptx() or officer::read_docx() document.
plot	A ggplot2 object created by a Stem plotting function, or an ms_barchart already produced by stem_as_mschart().
width, height	Size of the chart, in units. In Word both default to a readable 15 x 10 cm. In PowerPoint both default to NULL, which uses the body placeholder of the slide layout; give them to size the chart by hand.
left, top	PowerPoint only: position of the chart's top-left corner, in units. Only used when width and height are given; both default to 0.
units	Unit of width, height, left and top: "cm" (default), "mm" or "in".
location	PowerPoint only: an officer::ph_location() giving where to place the chart. Takes precedence over width/height/left/top.
...	Passed to stem_as_mschart().

Value

The document x, updated.

See Also

stem_as_mschart()

Examples

```
## Not run:
doc <- officer::read_pptx()
doc <- officer::add_slide(doc, "Title and Content", "Office Theme")
doc <- stem_add_chart(doc, stem_barplot(trust, government))
print(doc, target = "chart.pptx")

## End(Not run)
```

---

stem_as_mschart *Convert a Stem plot into a native Microsoft chart*

Description

Takes a ggplot2 object created by one of the Stem plotting functions and returns an mschart::ms_barchart() that renders as closely as possible to the ggplot original, but is a *native* Office chart: once placed in a PowerPoint or Word file it carries its own embedded worksheet, so the source data can be opened and edited in Excel.

Usage

```
stem_as_mschart(
  plot,
  title = NULL,
  labels = NULL,
  legend = NULL,
  num_fmt = NULL,
  axis_num_fmt = "0\"%\"",
  axis_show = TRUE,
  axis_major_unit = NULL,
  gap_width = 30,
  value_name = "%"
)
```

Arguments

plot	A ggplot2 object created by stem_barplot(), stem_inline(), stem_battery() or stem_multiselect().
title	Chart title. Defaults to the ggplot title (set with title_show = TRUE), or no title when the plot has none. Use NA to drop the title.
labels	If TRUE, prints a data label on every segment. Defaults to whatever the ggplot does.
legend	If TRUE, shows a legend. Defaults to TRUE for plots with a fill variable and FALSE otherwise, matching the ggplot.
num_fmt	Excel number format code for the data labels, e.g. "0", "0.0" or "0 \"%\"". By default the chart prints the plot's own label text instead of formatting the worksheet values, so the accuracy (label_accuracy) and the blanks left by label_hide carry over exactly. Give a format to label the values from the worksheet instead, which keeps the labels live when the data are edited in Excel.
axis_num_fmt	Excel number format code for the value axis. Defaults to "0\"%\"", matching the percentage axis of the Stem plots.
axis_show	If TRUE (default), draws the value axis.
axis_major_unit	Spacing of the value axis ticks, in percentage points. Defaults to 25 for stacked charts and 10 otherwise.
gap_width	Gap between bars, as a percentage of the bar width.
value_name	Name of the value column in the embedded worksheet; it is also the series name of charts without a fill variable.

Details

The aggregated data, the category order, the palette, the label formatting and the fonts are all read back off the ggplot object, so the native chart needs no recomputation and stays in sync with the ggplot version.

A few ggplot features have no Office equivalent and are dropped, with a message: confidence interval error bars (errorbar = TRUE) and the background bars of stem_multiselect().

Value

An object of class `ms_barchart`, ready for `officer::ph_with()` / `mschart::body_add_chart()` or for the Stem helpers `stem_add_chart()`, `stem_export_pptx()` and `stem_export_docx()`.

See Also

`stem_add_chart()`, `stem_export_pptx()`, `stem_export_docx()`

Examples

```
## Not run:
chart <- stem_as_mschart(stem_barplot(trust, government))

stem_export_pptx(
  stem_barplot(trust, government),
  stem_battery(trust, items = c(police, eu, government, army)),
  path = "charts.pptx"
)

## End(Not run)
```

`stem_barplot` *Horizontal bar plot*

Description

Plots the (possibly weighted) distribution of a categorical variable as horizontal bars. Without a `group`, one bar is drawn per item category. Supplying a `group` variable instead draws one stacked horizontal bar per group category, with the item mapped to fill; proportions are then computed within each group, so every bar sums to 100%.

Usage

```
stem_barplot(
  data,
  item,
  group = NULL,
  weight = NULL,
  collapse_item = NULL,
  collapse_group = NULL,
  palette = "div1",
  direction = 1,
  labels = TRUE,
  label_accuracy = 1,
  label_suffix = "",
  label_hide = NULL,
  label_color = "black",
  label_bicolor = TRUE,
```

```
  errorbar = FALSE,
  title_show = FALSE,
  title_quote = FALSE,
  title_wrap = 80
)
```

Arguments

data	Data frame holding the item (and group) variables.
item	Categorical variable to plot.
group	Optional grouping variable. When supplied, one stacked horizontal bar is drawn per group category (with the item mapped to fill) and proportions are computed within each group.
weight	Optional survey weights.
collapse_item	Optional named list passed to stem_summarise_cat() to collapse (or rename) item categories.
collapse_group	Optional named list to collapse (or rename) group categories.
palette	Name of a Stem palette. See stem_palette().
direction	Palette direction. Use -1 to reverse the colours.
labels	If TRUE, prints a percentage label on each bar.
label_accuracy	Rounding accuracy of labels. 1 gives whole numbers, 0.1 one decimal place.
label_suffix	Suffix appended to labels. Defaults to "".
label_hide	Proportions below this threshold are left unlabelled. Defaults to 0.05 when a group is supplied (to keep small stacked segments unlabelled) and 0 otherwise.
label_color	Colour of the segment labels in grouped (stacked) plots. Defaults to "black".
label_bicolor	If TRUE (default), in grouped (stacked) plots each label's colour follows the final fill of its segment: white on dark fills, label_color on light ones. It follows the fill even when the fill scale is replaced later (e.g. + scale_fill_manual()). Set to FALSE to colour every label with label_color.
errorbar	If TRUE, adds 95% confidence interval error bars (ungrouped plots only).
title_show	If TRUE, adds a plot title taken from the item's "label" attribute, falling back to the variable name when no label is present. The title is styled (e.g. drawn in bold) by theme_stem(). Defaults to FALSE.
title_quote	If TRUE, wraps the title in low/high double quotation marks („ and "). Defaults to FALSE.
title_wrap	Maximum number of characters per title line; longer titles are wrapped onto several lines so they do not overflow the plot. Use NULL or Inf to disable wrapping. Defaults to 80.

Value

A ggplot2 object.

Examples

```
## Not run:
stem_barplot(trust, government)
stem_barplot(trust, police, group = eu_index, weight = W)

## End(Not run)
```

`stem_battery` *Stacked bar plot for a battery of like items*

Description

Plots a set of categorical variables that share the same response categories (e.g. a battery of Likert items) as a single chart, drawing one stacked horizontal bar per item. Internally the items are reshaped to long format and handed to the same machinery as `stem_barplot()`, with the item taking the role of the group.

Usage

```
stem_battery(
  data,
  items,
  weight = NULL,
  order_by = NULL,
  item_label = TRUE,
  palette = "div1",
  direction = 1,
  labels = TRUE,
  label_accuracy = 1,
  label_suffix = "",
  label_hide = 0.05,
  label_color = "black",
  label_bicolor = TRUE
)
```

Arguments

`data`	Data frame holding the item (and group) variables.
`items`	Items to plot. Supports tidyselect helpers (e.g. `dplyr::starts_with()`). The response categories (and their order) are taken from the first item; if another item has answers outside them, `stem_battery()` stops with an error listing those answers. If the items have different numbers of valid (non-missing) answers, their bars rest on different bases and `stem_battery()` warns, listing them.
`weight`	Optional survey weights.

order_by	Optional vector of response categories used to order the items, e.g. c("Definitely Agree", "Rather Agree") orders items by their combined share of those two categories. Every value must match a category of the items exactly; otherwise stem_battery() stops with an error listing the values not found and the categories.
item_label	If TRUE (default), item labels (the "label" attribute) are used instead of variable names. Falls back to names, with a warning, if any item lacks a label. Items must have distinct labels; otherwise stem_battery() stops with an error naming them (use item_label = FALSE or give them distinct labels).
palette	Name of a Stem palette. See stem_palette().
direction	Palette direction. Use -1 to reverse the colours.
labels	If TRUE, prints a percentage label on each bar.
label_accuracy	Rounding accuracy of labels. 1 gives whole numbers, 0.1 one decimal place.
label_suffix	Suffix appended to labels. Defaults to "".
label_hide	Proportions below this threshold are left unlabelled. Defaults to 0.05 when a group is supplied (to keep small stacked segments unlabelled) and 0 otherwise.
label_color	Colour of the segment labels in grouped (stacked) plots. Defaults to "black".
label_bicolor	If TRUE (default), in grouped (stacked) plots each label's colour follows the final fill of its segment: white on dark fills, label_color on light ones. It follows the fill even when the fill scale is replaced later (e.g. + scale_fill_manual()). Set to FALSE to colour every label with label_color.

Value

A ggplot2 object.

Examples

```
## Not run:
stem_battery(trust, items = c(police, eu, government, army))
stem_battery(trust,
             items = c(police, eu, government, army),
             weight = W,
             order_by = c("Definitely Agree", "Rather Agree"))

## End(Not run)
```

---

stem_export_docx *Export Stem plots to Word as native charts*

Description

Writes a .docx file holding one native Office chart per plot, each with an embedded worksheet that can be edited in Excel.

Usage

```
stem_export_docx(
  ...,
  path,
  template = NULL,
  width = 15,
  height = 10,
  units = "cm",
  chart_args = list()
)
```

Arguments

`...`	One or more ggplot2 objects created by the Stem plotting functions (or `ms_barchart` objects from `stem_as_mschart()`). A single list of plots is also accepted.
`path`	Path of the `.docx` file to write.
`template`	Optional path to a `.docx` template to build on.
`width, height`	Size of each chart, in `units`. Defaults to 15 x 10 cm, which fits the text width of a portrait A4 page.
`units`	Unit of `width`, `height`, `left` and `top`: `"cm"` (default), `"mm"` or `"in"`.
`chart_args`	A named list of arguments passed on to `stem_as_mschart()`.

Value

The path, invisibly.

See Also

`stem_as_mschart()`, `stem_export_pptx()`

Examples

```
## Not run:
stem_export_docx(stem_barplot(trust, government), path = "trust.docx")

# Half-page charts.
stem_export_docx(
  stem_barplot(trust, government),
  path = "trust.docx",
  width = 16, height = 8
)

## End(Not run)
```

`stem_export_pptx` *Export Stem plots to PowerPoint as native charts*

Description

Writes one slide per plot, each holding a native Office chart whose source data can be edited in Excel.

Usage

```
stem_export_pptx(
  ...,
  path,
  template = NULL,
  layout = "Title and Content",
  master = "Office Theme",
  width = NULL,
  height = NULL,
  left = NULL,
  top = NULL,
  units = "cm",
  location = NULL,
  chart_args = list()
)
```

Arguments

`...`	One or more ggplot2 objects created by the Stem plotting functions (or `ms_barchart` objects from `stem_as_mschart()`). A single list of plots is also accepted.
`path`	Path of the `.pptx` file to write.
`template`	Optional path to a `.pptx` template to build on. Defaults to officer's blank presentation.
`layout, master`	Name of the slide layout and of the master to use for the new slides.
`width, height`	Size of each chart, in `units`. Both default to `NULL`, which uses the body placeholder of `layout`, so a template's own geometry is respected. Give them to size the charts by hand.
`left, top`	Position of each chart's top-left corner, in `units`. Only used when `width` and `height` are given; both default to `0`.
`units`	Unit of `width`, `height`, `left` and `top`: `"cm"` (default), `"mm"` or `"in"`.
`location`	An `officer::ph_location()` giving where to place the charts. Takes precedence over `width`/`height`/`left`/`top`.
`chart_args`	A named list of arguments passed on to `stem_as_mschart()`.

Value

The path, invisibly.

See Also

stem_as_mschart(), stem_export_docx()

Examples

```
## Not run:
stem_export_pptx(
  stem_barplot(trust, government, title_show = TRUE),
  stem_inline(trust, police),
  path = "trust.pptx"
)

# A 24 x 12 cm chart, 2 cm from the left and 4 cm from the top of the slide.
stem_export_pptx(
  stem_barplot(trust, government),
  path = "trust.pptx",
  width = 24, height = 12, left = 2, top = 4
)

## End(Not run)
```

---

stem_inline *Inline stacked bar plot*

Description

Plots the (possibly weighted) distribution of a single categorical variable as one stacked horizontal bar. Useful as a compact, inline summary of a single item.

Usage

```
stem_inline(
  data,
  item,
  weight = NULL,
  collapse_item = NULL,
  palette = "div1",
  direction = 1,
  labels = TRUE,
  label_accuracy = 1,
  label_suffix = "",
  label_hide = 0.05,
  label_color = "black",
  label_bicolor = TRUE,
  title_show = FALSE,
  title_quote = FALSE,
  title_wrap = 80
)
```

Arguments

data	Data frame holding the item (and group) variables.
item	Categorical variable to plot.
weight	Optional survey weights.
collapse_item	Optional named list passed to stem_summarise_cat() to collapse (or rename) item categories.
palette	Name of a Stem palette. See stem_palette().
direction	Palette direction. Use -1 to reverse the colours.
labels	If TRUE, prints a percentage label on each bar.
label_accuracy	Rounding accuracy of labels. 1 gives whole numbers, 0.1 one decimal place.
label_suffix	Suffix appended to labels. Defaults to "".
label_hide	Proportions below this threshold are left unlabelled. Defaults to 0.05 when a group is supplied (to keep small stacked segments unlabelled) and 0 otherwise.
label_color	Colour of the segment labels in grouped (stacked) plots. Defaults to "black".
label_bicolor	If TRUE (default), in grouped (stacked) plots each label's colour follows the final fill of its segment: white on dark fills, label_color on light ones. It follows the fill even when the fill scale is replaced later (e.g. + scale_fill_manual()). Set to FALSE to colour every label with label_color.
title_show	If TRUE, adds a plot title taken from the item's "label" attribute, falling back to the variable name when no label is present. The title is styled (e.g. drawn in bold) by theme_stem(). Defaults to FALSE.
title_quote	If TRUE, wraps the title in low/high double quotation marks („ and “). Defaults to FALSE.
title_wrap	Maximum number of characters per title line; longer titles are wrapped onto several lines so they do not overflow the plot. Use NULL or Inf to disable wrapping. Defaults to 80.

Value

A ggplot2 object.

Examples

```
## Not run:
stem_inline(trust, police)
stem_inline(trust, police, weight = W)

## End(Not run)
```

`stem_multiselect` *Bar plot for a set of multiple-choice items*

Description

Plots a set of "select all that apply" items, showing the share of respondents that picked each option. The items are reshaped to long format and their (possibly weighted) frequencies are rescaled by the number of items so that the bars express percentages of respondents rather than of responses.

Usage

```
stem_multiselect(
  data,
  items,
  group = NULL,
  weight = NULL,
  palette = "div1",
  direction = 1,
  labels = TRUE,
  label_accuracy = 1,
  label_suffix = "",
  label_hide = 0,
  infreq_order = TRUE,
  background = TRUE,
  background_fill = "grey85",
  background_alpha = 1
)
```

Arguments

`data`	Data frame holding the item (and group) variables.
`items`	Items to plot. Supports tidyselect helpers (e.g. `dplyr::starts_with()`).
`group`	Optional grouping variable. When supplied, one stacked horizontal bar is drawn per group category (with the item mapped to fill) and proportions are computed within each group.
`weight`	Optional survey weights.
`palette`	Name of a Stem palette. See `stem_palette()`.
`direction`	Palette direction. Use `-1` to reverse the colours.
`labels`	If `TRUE`, prints a percentage label on each bar.
`label_accuracy`	Rounding accuracy of labels. `1` gives whole numbers, `0.1` one decimal place.
`label_suffix`	Suffix appended to labels. Defaults to `""`.
`label_hide`	Proportions below this threshold are left unlabelled. Defaults to `0.05` when a `group` is supplied (to keep small stacked segments unlabelled) and `0` otherwise.
`infreq_order`	If `TRUE` (default), options are ordered by frequency.

background — If TRUE (default, ungrouped plots only), draws a full-width background bar behind each option.

background_fill — Fill colour of the background bars.

background_alpha — Opacity of the background bars.

Value

A ggplot2 object.

Examples

```
## Not run:
stem_multiselect(trust, items = dplyr::starts_with("biggest_concern"))
stem_multiselect(trust, items = dplyr::starts_with("biggest_concern"), weight = W)

## End(Not run)
```

---

stem_palette *Retrieve a stem colour palette*

Description

Returns a character vector of hex colour codes for the requested palette. The vector carries a type attribute ("nominal", "diverging", or "sequential") used internally by stem_palette_gen().

Usage

```
stem_palette(palette = "div1")
```

Arguments

palette — Name of the palette. One of:

Nominal (categorical, unordered data): "nom1", "nom2".

Sequential (ordered data from low to high): "seq1", "seq2", "seq3", "seq4".

Diverging (data with a meaningful midpoint): "modern", "div1", "div2", "div3".

Value

Character vector of hex colour codes with a type attribute ("nominal", "diverging", or "sequential").

See Also

stem_palette_gen(), scale_colour_stem(), scale_fill_stem()

Examples

```
# Returns a plain character vector
stem_palette("modern")

# Inspect the palette type
attr(stem_palette("modern"), "type")

# Use with scales package for inspection
scales::show_col(stem_palette("nom1"))
```

`stem_palette_gen` *Colour palette generator factory*

Description

Returns a function `function(n)` that selects `n` colours from the named palette. This factory is used internally by `scale_colour_stem()` and `scale_fill_stem()` to satisfy ggplot2's discrete scale interface, but can also be called directly.

Diverging palettes use `select_diverging()` to pick colours that preserve the symmetric structure of the palette. All other palette types return the first `n` colours, optionally reversed.

Usage

```
stem_palette_gen(palette = "div1", direction = 1)
```

Arguments

`palette`	Name of the palette. See `stem_palette()` for the full list of valid names.
`direction`	`1` (default) returns colours in their natural order; `-1` reverses the order.

Value

A `function(n)` that returns a character vector of `n` hex colour codes.

See Also

`stem_palette()`, `select_diverging()`, `scale_colour_stem()`

Examples

```
# Create a generator then call it for 3 colours
gen <- stem_palette_gen("modern")
gen(3)

# Reversed order
stem_palette_gen("div1", direction = -1)(5)
```

stem_palettes_all *Display all available stem palettes*

Description

Plots a visual overview of all available stem colour palettes, grouped by type (nominal, diverging, sequential). Each row shows one palette's colours as tiles, similar to `RColorBrewer::display.brewer.all()`.

Usage

```
stem_palettes_all()
```

Value

A ggplot2 object.

See Also

`stem_palette()`, `scale_colour_stem()`

Examples

```
stem_palettes_all()
```

stem_summarise *Summarise numeric or categorical variable (possibly segmented)*

Description

Summarise either numeric or categerical variable, possible segment by another categorical variable. Actually a wrapper for `stem_summarise_cat()` and `stem_summarise_num()`.

Usage

```
stem_summarise(
  data,
  item,
  group = NULL,
  weight = NULL,
  long = FALSE,
  collapse_item = NULL,
  collapse_group = NULL,
  na.rm = TRUE,
  return_n = FALSE
)
```

Arguments

data	Dataframe including variables to be analyzed
item	Variable to be summarised
group	Optional segmenting variable
weight	Optional survey weights
long	Returns data in long format. Useful if multiple dataframes are to be merged
collapse_item	Named list. Optionally collapes (or renames) categories of the item variable (Only if item is categorical).
collapse_group	Named list. Optionally collapes (or renames) categories of the group variable
na.rm	If TRUE (default), NA values in the item are dropped and observations with an NA group are excluded. If FALSE, NA is kept (as its own category for categorical items, or yielding an NA mean for numeric items) and NA group values form their own group.
return_n	If TRUE, returns absolute group sizes.

Details

Apart from either point estimates (proportions or means), the function also returns 95% confidence interval bounds. If unweighted, the intervals are computed using the basic `sqrt((p * (1-p)) / n)` formula, where `n` is the size of the (possibly segmented) sample. If the estimated proportions are very high/low, this may lead to interval estimates outside of the (0;1) bounds. If weights are used, the confidence intervals come from `srvyr`/`survey` (Taylor-series linearization).

If `long = TRUE`, new column is added holding name of the item (and group) variable. This is useful if you need to loop through multiple variables and bind the results into a single data frame. See online vignettes for details.

`collapse_item` and `collapse_group` can be used to collapse categories of item and group variable. The named list should include vector of old category names with corresponding to the new category name, e.g. `list(Agree = c("Definitely Agree", "Rather Agree"))`. You can also use the arguments to rename categories (`list(Yes = "Agree")`) or pass other arguments from the `forcats::fct_collapse()` function.

If `return_n = TRUE`, columns holding the absolute frequencies will be added. Column `n` is the number of observations for the specific combination of item and grouping variable, `item_n` is the frequency of the item categories and `group_n` is the frequency of the group categories.

Value

An aggregated data frame with point estimates (either proportions or means) and 95% confidence intervals.

Examples

```
stem_summarise(data = trust, item = government)

stem_summarise(data = trust, item = age, group = eu_index)

stem_summarise(data = trust, item = government,
               collapse_item = list(Agree = c("Definitely Agree", "Rather Agree")))
```

`stem_summarise_cat` *Summarise categorical variable (possibly segmented)*

Description

Takes one categorical variable and returns a tidy dataframe, including relative frequencies and absolute size of each group. The variable can be further segmented by another categorical variable. Can also take survey weights. Returns 95% confidence intervals.

Usage

```
stem_summarise_cat(
  data,
  item,
  group = NULL,
  weight = NULL,
  long = FALSE,
  collapse_item = NULL,
  collapse_group = NULL,
  na.rm = TRUE,
  return_n = FALSE
)
```

Arguments

`data`	Dataframe including variables to be analyzed
`item`	Variable to be summarised
`group`	Optional segmenting variable
`weight`	Optional survey weights
`long`	Returns data in long format. Useful if multiple dataframes are to be merged
`collapse_item`	Named list. Optionally collapes (or renames) categories of the item variable
`collapse_group`	Named list. Optionally collapes (or renames) categories of the group variable
`na.rm`	If `TRUE` (default), `NA` values in the item are dropped and observations with an `NA` group are excluded. If `FALSE`, `NA` is kept as its own item category (inflating the denominator so proportions still sum to 1) and `NA` group values form their own group.
`return_n`	If `TRUE`, returns absolute group sizes.

Value

Tidy dataframe

Examples

```
trust |> stem_summarise_cat(item = police,
                            group = eu,
                            weight = W,
                      collapse_item = list(Agree = c("Definitely Agree", "Rather Agree")))
```

`stem_summarise_num` *Summarise numeric variable using arithmetic mean (possibly segmented)*

Description

Takes one numeric variable and returns a tidy dataframe, including relative mean and absolute size of each group. The variable can be further segmented by another categorical variable. Can also take survey weights. Returns 95% confidence intervals.

Usage

```
stem_summarise_num(
  data,
  item,
  group = NULL,
  weight = NULL,
  long = FALSE,
  collapse_group = NULL,
  na.rm = TRUE,
  return_n = FALSE
)
```

Arguments

`data`	Dataframe including data to be analyzed
`item`	Variable to be summarised
`group`	Optional segmenting variable
`weight`	Optional survey weights
`long`	Returns data in long format. Useful if multiple dataframes are to be merged
`collapse_group`	Named list. Optionally collapses (or renames) categories of the group variable
`na.rm`	If `TRUE` (default), `NA` values in the item are dropped from the calculations and observations with an `NA` group are excluded. If `FALSE`, `NA` item values are kept (so the mean is `NA`) and `NA` group values form their own group.
`return_n`	If `TRUE`, returns absolute group sizes.

Value

Tidy dataframe

Examples

```
trust |> stem_summarise_num(age,
                            group = eu_index,
                            weight = W,
         collapse_group = list(`Neutral or no opinion` = c("Neutral", "Doesn't Know")))
```

`theme_stem` *Complete ggplot2 theme for Stem.*

Description

A *complete* ggplot2 theme carrying the Stem look. Because it is complete, the Stem plotting functions do not apply it themselves; instead activate it globally with `ggplot2::theme_set()` so every plot picks it up:

Usage

```
theme_stem(
  ink = "black",
  paper = "white",
  accent = "#35978F",
  family = "Calibri",
  label_size = 14,
  ...
)
```

Arguments

`ink`	Foreground colour used for text, titles and axis lines. Also passed to `ggplot2::element_geom()` as the default geom foreground colour.
`paper`	Background colour used behind the panel, plot, strips and legend keys. Also passed to `ggplot2::element_geom()` as the default geom background colour. Use `NA` for a transparent background (useful when exporting plots to be placed on a coloured surface).
`accent`	Accent colour passed to `ggplot2::element_geom()` as the default geom accent colour (e.g. the fill of `ggplot2::geom_smooth()`). Defaults to the primary Stem brand colour.
`family`	Font family for all text. Defaults to `"Calibri"`, the Stem house font. Pass `""` to use the graphics device's default family (useful on machines where Calibri is not installed).
`label_size`	Point size of the numeric labels drawn inside the plot (the `stem_label` values printed by `stem_barplot()` and friends). Passed to `ggplot2::element_geom()` as `fontsize`, so it is given in points, like the other text sizes of the theme. Defaults to `14`, the size the labels are exported at by `stem_export_pptx()` and `stem_export_docx()`.
`...`	Arguments to be passed to `ggplot2::theme()`, overriding the Stem defaults.

Details

`ggplot2::theme_set(theme_stem())`

It can still be added to an individual plot with `+ theme_stem()` in the usual way. The theme is built on `ggplot2::theme_grey()` with the Stem overrides layered on top.

Value

A complete ggplot2 `ggplot2::theme()` object.

Examples

```
## Not run:
# Activate globally for the whole session.
ggplot2::theme_set(theme_stem())
stem_barplot(trust, government)

# Or add it to a single plot.
stem_barplot(trust, government) + theme_stem()

## End(Not run)
```

---

trust *Simulated data on attitudes towards institutions*

Description

Simulated data to use for examples and function testing.

Usage

```
trust
```

Format

`trust`**:**

A data frame with 1 000 rows and 12 columns:

police Generally speaking, can the following be trusted: Police

government Generally speaking, can the following be trusted: Czech government

eu Generally speaking, can the following be trusted: European Union

army Generally speaking, can the following be trusted: Army

scientists Generally speaking, can the following be trusted: Scientists

eu_index Attitudes towards EU index

nat_index Openness towards society index

age Age

W Sampling weight

biggest_concern1 Biggest concern about society in the last month

biggest_concern2 Second biggest concern about society in the last month
biggest_concern3 Third biggest concern about society in the last month ...

Index

∗ **datasets**
trust, 26

collapse_cats, 2
create_project, 3

forcats::fct_collapse(), *22*

ggplot2::discrete_scale(), *4*
ggplot2::element_geom(), *25*
ggplot2::geom_smooth(), *25*
ggplot2::theme(), *25, 26*
ggplot2::theme_grey(), *26*
ggplot2::theme_set(), *25*

mschart::body_add_chart(), *10*
mschart::ms_barchart(), *8*

officer::add_slide(), *7*
officer::ph_location(), *8*, *15*
officer::ph_with(), *10*
officer::read_docx(), *8*
officer::read_pptx(), *8*

scale_color_stem *(*scale_fill_stem*)*, 4
scale_colour_stem *(*scale_fill_stem*)*, 4
scale_colour_stem(), *19–21*
scale_fill_stem, 4
scale_fill_stem(), *19, 20*
se_mean, 5
se_median, 5
se_prop, 6
select_diverging, 6
select_diverging(), *20*
stem_add_chart, 7
stem_add_chart(), *10*
stem_as_mschart, 8
stem_as_mschart(), *7, 8*, *14–16*
stem_barplot, 10
stem_barplot(), *9*, *12*, *25*
stem_battery, 12
stem_battery(), *9*
stem_export_docx, 13
stem_export_docx(), *10*, *16*, *25*
stem_export_pptx, 15
stem_export_pptx(), *10*, *14*, *25*
stem_inline, 16
stem_inline(), *9*
stem_multiselect, 18
stem_multiselect(), *9*
stem_palette, 19
stem_palette(), *4*, *7*, *11*, *13*, *17, 18*, *20, 21*
stem_palette_gen, 20
stem_palette_gen(), *4*, *7*, *19*
stem_palettes_all, 21
stem_summarise, 21
stem_summarise_cat, 23
stem_summarise_cat(), *11*, *17*, *21*
stem_summarise_num, 24
stem_summarise_num(), *21*

theme_stem, 25
theme_stem(), *11*, *17*
trust, 26