Package: hairline (via r-universe)

October 29, 2026

Title Clean, Predictable Figures for Medical and Data Science

Version 0.2.1

Description A design system for 'ggplot2' figures: one frame for titles, legends, panels and captions, a fixed spacing and type scale, colour-vision checked palettes, brand files in the 'brand.yml' format, and print-ready export. Produces output matching the 'hairline' Python package.

License MIT + file LICENSE

URL https://github.com/schradern/hairline

BugReports https://github.com/schradern/hairline/issues

Copyright The bundled 'Hairline Sans' font is a modified Public Sans, Copyright 2015 The Public Sans Project Authors, under the SIL Open Font License 1.1; see inst/COPYRIGHTS.

Encoding UTF-8

Roxygen list(markdown = TRUE)

Depends R (>= 4.3)

Imports S7, ggplot2 (>= 4.0.0), grid, grDevices, jsonlite, ragg, rlang, systemfonts (>= 1.1.0), textshaping, tools, utils, yaml

Suggests cowplot, metafor, patchwork, png, rsvg, survival, svglite, testthat (>= 3.0.0), withr

Config/testthat/edition 3

Config/roxygen2/version 8.1.1

Config/pak/sysreqs libfontconfig1-dev libfreetype6-dev libfribidi-dev libharfbuzz-dev libicu-dev libjpeg-dev libpng-dev libtiff-dev libwebp-dev

Repository https://schradern.r-universe.dev

Date/Publication 2026-10-22 19:23:49 UTC

RemoteUrl https://github.com/schradern/hairline

RemoteRef HEAD

RemoteSha 4f8b16470bea78c26171ab4e30d61037a8f75560

RemoteSubdir r

Contents

hairline_colours . . . 3
hl_arcs . . . 4
hl_bars . . . 5
hl_brand_report . . . 7
hl_check_palette . . . 7
hl_chord . . . 8
hl_color . . . 9
hl_compare . . . 10
hl_correlation . . . 10
hl_describe . . . 11
hl_distribution . . . 12
hl_donut . . . 13
hl_dotplot . . . 14
hl_effects . . . 15
hl_find_brand . . . 16
hl_forest . . . 17
hl_format_number . . . 18
hl_funnel . . . 19
hl_highlight . . . 20
hl_install_font . . . 20
hl_kaplan_meier . . . 21
hl_layout_report . . . 22
hl_legend . . . 23
hl_load_brand . . . 23
hl_over_time . . . 24
hl_palette . . . 25
hl_palettes . . . 26
hl_panel_header . . . 26
hl_partition . . . 27
hl_prisma . . . 28
hl_raincloud . . . 29
hl_ramp . . . 30
hl_reference . . . 31
hl_ridgeline . . . 32
hl_risk_of_bias . . . 33
hl_sankey . . . 34
hl_save . . . 35
hl_scatter . . . 36
hl_settings . . . 37
hl_spread . . . 37
hl_study . . . 38
hl_survival . . . 40
hl_swatches . . . 40
hl_timeseries . . . 41
hl_use . . . 43
hl_venn . . . 44

scale_colour_hairline 45
scale_y_hairline 46
theme_hairline 47

Index **48**

`hairline_colours` *Colours in hairline*

Description

hairline picks colours for you: every chart function colours its groups, amounts and highlights on its own. This page shows which colour sets there are, how to choose another one, and how to keep a group's colour the same across a report.

Which colours for which job

- Groups (arms, units, diagnoses): a categorical palette, e.g. `hl_use(palette = "calm")`.
- Two groups (intervention, control): a pair, `hl_use(palette = "pair")`.
- A printout in grey: `"ink"`, or the palette `"deep"`.
- Amounts (counts, rates, densities): a sequential ramp, e.g. `scale_fill_hairline_c("blues")`.
- Values above and below a midpoint: a diverging ramp, `scale_fill_hairline_c("blue-coral")`.
- One group against the rest: `+ hl_highlight("Unit B")`, or `highlight =` in `hl_timeseries()`.
- A single series: the accent, `hl_use(accent = "#0b6e4f")`.
- Risk of bias: the `"judgement"` set, applied by `hl_risk_of_bias()`.

See what there is

`hl_palettes()` lists every colour set with its kind, number of colours, use, and which one is active; `hl_swatches()` draws them. The six categorical palettes have eight colours each. Every set opens with its own blue, then its own red, so two groups always read blue against red: `"daylight"` (vivid; the default), `"calm"` (softer and lighter, for long reports), `"deep"` (stays apart when printed in grey), `"lagoon"` (cool), `"citrus"` (warm) and `"bold"` (saturated, for slides). The purpose sets are `"pair"`, `"pair-teal"` and `"pair-print"` for two groups, `"ink"` for greyscale print and `"judgement"` for low, some concerns and high. The ramps are `"blues"`, `"teals"`, `"oranges"`, `"berries"` and `"greys"` (sequential) and `"blue-coral"` and `"teal-berry"` (diverging).

Colour by count

The number of groups decides the colours, the same in every chart: two groups take the first two colours, three groups the first three. The first two, three and four colours of each palette are its most readable sets of that size, for colour-blind readers too, and because the sets are nested a group keeps its colour when more groups are shown. A ninth group is an error, not a repeated colour: fold small groups into "Other" or split the figure into panels.

Colour follows the name

Within a session a group keeps the colour it got first, in every later chart. Pin colours to names with `hl_use(colors = c("Unit B" = "#e79d01"))` and read them back with `hl_color()` and `hl_colors()`. For colours that stay the same across a report whatever order the charts are drawn in, put them in the brand file under `defaults: hairline: colors:`; `groups:` gives the units of a section tints of the section's colour.

Brand and your own colours

A `_brand.yml` file holds an institution's palette (`defaults: hairline: categorical:`), found from the working directory upwards or given with `hl_use(brand = )`; several files layer from institution to project, and `hl_brand_report()` shows what was taken and which layer each colour came from. While a brand is active, its ramps `"brand-sequential"` and `"brand-diverging"` (from its `sequential:` and `diverging:` colours, else its first one and two colours) are the default for `hl_ramp()`, the continuous scales and the correlation heatmap. Any vector of hex colours can be a palette; `hl_check_palette()` measures how far apart the colours are for colour-blind readers and everyone else, and their contrast with the background (`pairs = "all"` for scatter plots and small multiples).

Light colours and dark backgrounds

Colours below 3:1 contrast on white stay as they are for bars and areas; lines in them are drawn a little wider and points are filled with a darker shade of the same hue. With `hl_use(scheme = "dark")` the background, text and grid greys turn dark; data colours stay, and thin marks are shaded lighter where needed. Text never takes a data colour.

See Also

`hl_palettes()`, `hl_swatches()`, `hl_palette()`, `hl_ramp()`, `hl_use()`, `hl_check_palette()`.

Examples

```
hl_palettes("categorical")
hl_palette("pair")
hl_swatches(c("daylight", "calm", "pair", "blues"))
```

---

hl_arcs *Arc diagram*

Description

Nodes on a baseline and an arc above for each pair that exchanges something, its width following the value; without `value` each row counts once. Labels turn upright when they would collide.

Usage

```
hl_arcs(
  data,
  source,
  target,
  value = NULL,
  title = NULL,
  subtitle = NULL,
  caption = NULL,
  order = NULL
)
```

Arguments

`data`	A data frame, one row per flow (or per pair with a value).
`source`, `target`	Columns naming the two ends.
`value`	Column with the amounts; `NULL` counts rows.
`title`, `subtitle`, `caption`	Frame texts; the title defaults to the columns.
`order`	Node order along the baseline.

Value

A ggplot with the hairline frame.

Examples

```
moves <- data.frame(origin = c("ICU", "ICU", "Ward", "Theatre", "ICU", "Ward"),
                    destination = c("Ward", "Ward", "Home", "ICU", "Home", "ICU"))
hl_arcs(moves, "origin", "destination", order = c("Theatre", "ICU", "Ward", "Home"))
```

---

hl_bars *Bar chart*

Description

Bars per category; without `value` the rows are counted. Long or many labels turn the bars horizontal; a single series is sorted by value unless `order` or `sort = "given"` says otherwise. Values sit at the bar ends.

Usage

```
hl_bars(
  data,
  category,
  value = NULL,
  by = NULL,
  title = NULL,
  subtitle = NULL,
  caption = NULL,
  stacked = FALSE,
  percent = FALSE,
  horizontal = NULL,
  sort = NULL,
  order = NULL,
  labels = TRUE,
  colors = NULL,
  size = NULL
)
```

Arguments

`data`	A data frame.
`category`	Column with the categories.
`value`	Column with the amounts; `NULL` counts rows.
`by`	Grouping column; groups side by side, or stacked with `stacked = TRUE`.
`title, subtitle, caption`	Frame texts; the title defaults to the columns.
`stacked`	Stack the groups.
`percent`	Show shares of each category instead of amounts.
`horizontal`	Force horizontal (`TRUE`) or vertical (`FALSE`) bars. By default many or long labels lie down, and so do labels too wide for upright bars at the width of `size`.
`sort`	`"value"` or `"given"`; default by value, except for groups side by side.
`order`	Category order.
`labels`	Print the values: at the bar ends, or inside stacked segments with the totals at the ends.
`colors`	Named colours for this chart only, e.g. `c(Male = "#0a6ddd")`; they override the colour set for the number of groups and are not remembered for other charts.
`size`	Size preset the bars are laid out for, as in `hl_save()` (`NULL`: `"onehalf"`): its width decides between upright and lying bars, and `hl_save()` uses it unless given another size.

Value

A ggplot with the hairline frame.

Examples

```
admissions <- data.frame(
  diagnosis = c("Sepsis", "Sepsis", "Trauma", "Cardiac", "Cardiac", "Cardiac"),
  sex = c("Female", "Male", "Male", "Female", "Male", "Male"))
hl_bars(admissions, "diagnosis")
hl_bars(admissions, "diagnosis", by = "sex", stacked = TRUE)
```

---

hl_brand_report *Brand report*

Description

Brand report

Usage

```
hl_brand_report(brand = NULL)
```

Arguments

brand	A brand from hl_load_brand(), or NULL for the active one.

Value

A character string: the layers by number with their paths, then per slot the name, colour as given and as used, light slot and the layer that set it, the checks, and the pinned or grouped colours with their layers.

Examples

```
path <- file.path(tempdir(), "_brand.yml")
writeLines(c("meta:", "  name: Example Hospital", "color:", "  palette:",
           "    harbor: \"#0b3d6e\"", "    poppy: \"#f2545b\"", "defaults:", "  hairline:",
             "    categorical: [harbor, poppy]"), path)
hl_brand_report(hl_load_brand(path))
```

---

hl_check_palette *Check a palette for readability*

Description

Runs hairline's palette checks: lightness band, chroma floor, separation under simulated protanopia and deuteranopia, normal-vision separation, and contrast against the surface.

Usage

```
hl_check_palette(colors, surface = "#ffffff", pairs = c("adjacent", "all"))
```

Arguments

colors	Character vector of hex colours, in slot order.
surface	Background colour.
pairs	"adjacent" for bars, stacks and lines; "all" for scatter plots and small multiples.

Value

A list with passed, worst_cvd, worst_normal, head_all_pairs, light_slots, warnings, failures.

Examples

```
hl_check_palette(hl_palette(n = 4))
hl_check_palette(c("#1f77b4", "#2b83c4"), pairs = "all")
```

---

hl_chord *Chord diagram*

Description

Groups as arcs on a ring, each sized by its total, and ribbons through the centre joining each pair (both directions combined), as wide as the pair's value at both ends; without value each row counts once.

Usage

```
hl_chord(
  data,
  source,
  target,
  value = NULL,
  title = NULL,
  subtitle = NULL,
  caption = NULL,
  order = NULL,
  colors = NULL
)
```

Arguments

data	A data frame, one row per exchange (or per pair with a value).
source, target	Columns naming the two groups.
value	Column with the amounts; NULL counts rows.
title, subtitle, caption	Frame texts; the title defaults to the columns.
order	Group order around the ring, clockwise from the top.
colors	Named colours for this chart only; they override the colour set for the number of groups.

Value

A ggplot with the hairline frame.

Examples

```
transfers <- data.frame(origin = c("ICU", "Ward", "ICU", "Step-down", "Ward"),
                        destination = c("Ward", "ICU", "Step-down", "Ward", "Step-down"))
hl_chord(transfers, "origin", "destination")
```

---

hl_color *Entity colours*

Description

`hl_color()` returns an entity's colour, assigning the first free palette slot the first time an entity is seen; the colour stays for the session. `hl_colors()` returns every colour fixed or assigned so far.

Usage

```
hl_color(name)

hl_colors()
```

Arguments

`name`	Entity name, e.g. a ward or treatment arm.

Value

A hex colour, or a named character vector.

Examples

```
hl_use(colors = c("Unit A" = "#0a6ddd", "Unit B" = "#e79d01"))
hl_color("Unit B")
hl_colors()
hl_use()
```

hl_compare *Compare one variable between groups*

Description

Numeric variables: median (IQR) per group with the Wilcoxon rank-sum test (two groups) or Kruskal–Wallis (more), shown as a raincloud. Categorical variables: n (%) per group with the chi-squared test (Fisher's exact test for a 2 x 2 table with an expected count below 5), shown as 100 % bars.

Usage

```
hl_compare(study, variable)
```

Arguments

study	A study from hl_study() with a group.
variable	The variable to compare.

Value

A hairline_result.

Examples

```
set.seed(1)
cohort <- data.frame(arm = rep(c("Standard care", "Early mobilisation"), each = 40),
                     age = round(stats::rnorm(80, 64, 12)),
                     sex = sample(c("Female", "Male"), 80, TRUE),
                months = round(stats::rexp(80, 0.05), 1), died = stats::rbinom(80, 1, 0.4))
study <- hl_study(cohort, group = "arm", time = "months", event = "died",
                  units = c(age = "years", months = "months"))
hl_compare(study, "age")$text
hl_compare(study, "sex")$table
```

hl_correlation *Correlation heatmap*

Description

Correlations between numeric columns (all of them by default) as the lower triangle on a diverging ramp, values printed in the cells, each variable named on the empty diagonal; rows with gaps are left out.

Usage

```
hl_correlation(
  data,
  columns = NULL,
  method = "pearson",
  title = NULL,
  subtitle = NULL,
  caption = NULL
)
```

Arguments

`data`	A data frame.
`columns`	Columns to correlate; `NULL` takes every numeric column.
`method`	`"pearson"` or `"spearman"`.
`title, subtitle, caption`	Frame texts; the title defaults to the method.

Value

A ggplot with the hairline frame.

Examples

```
hl_correlation(datasets::mtcars, c("mpg", "disp", "hp", "wt"))
```

`hl_describe` *Table 1: the cohort by group*

Description

One row per statistic or category, one column per group plus Overall, with an overview figure of the numeric variables (median and IQR).

Usage

```
hl_describe(study, stats = "median_iqr")
```

Arguments

`study`	A study from `hl_study()`.
`stats`	Statistics for numeric variables: `"median_iqr"`, `"mean_sd"`, `"range"`, `"n_missing"`.

Value

A `hairline_result`: `figure`, `table`, `values`, `text` and the unrounded `numbers` they are formatted from (per group n, rows missing a value or a group, and the statistics).

Examples

```
set.seed(1)
cohort <- data.frame(arm = rep(c("Standard care", "Early mobilisation"), each = 40),
                     age = round(stats::rnorm(80, 64, 12)),
                     sex = sample(c("Female", "Male"), 80, TRUE),
                  months = round(stats::rexp(80, 0.05), 1), died = stats::rbinom(80, 1, 0.4))
study <- hl_study(cohort, group = "arm", time = "months", event = "died",
                  units = c(age = "years", months = "months"))
table_one <- hl_describe(study)
table_one$table
table_one$figure
```

`hl_distribution` *Distribution of one numeric variable by group*

Description

Distribution of one numeric variable by group

Usage

```
hl_distribution(study, variable, stats = "median_iqr")
```

Arguments

`study`	A study from `hl_study()`.
`variable`	A numeric variable.
`stats`	Statistics as in `hl_describe()`.

Value

A `hairline_result`; the figure is a raincloud, or a ridgeline from five groups.

Examples

```
set.seed(1)
cohort <- data.frame(arm = rep(c("Standard care", "Early mobilisation"), each = 40),
                     age = round(stats::rnorm(80, 64, 12)),
                     sex = sample(c("Female", "Male"), 80, TRUE),
                  months = round(stats::rexp(80, 0.05), 1), died = stats::rbinom(80, 1, 0.4))
study <- hl_study(cohort, group = "arm", time = "months", event = "died",
                  units = c(age = "years", months = "months"))
hl_distribution(study, "age", stats = c("median_iqr", "range"))$values
```

`hl_donut` *Donut and nested donut*

Description

One ring with labels outside and leader lines, or up to three hierarchy levels with a hierarchical key; `labels = "ring"` labels the arcs instead. More than seven slices of a single ring fold into "Other". Children use tints of their parent's colour.

Usage

```
hl_donut(
  data,
  label = NULL,
  value = NULL,
  levels = NULL,
  title = NULL,
  subtitle = NULL,
  caption = NULL,
  labels = NULL,
  total = TRUE,
  total_label = NULL,
  colors = NULL,
  order = NULL
)
```

Arguments

`data`	A data frame.
`label`	Column naming the slices of a single ring.
`value`	Column with the amounts; `NULL` counts rows.
`levels`	Up to three hierarchy columns, inner ring first; replaces `label`.
`title`, `subtitle`, `caption`	Frame texts.
`labels`	`"outside"` (one ring), `"key"` or `"ring"`; `NULL` picks outside for one ring, key otherwise.
`total`	Show the total in the centre.
`total_label`	Text under the total, e.g. `"patients"`.
`colors`	Named colours for this chart only, e.g. `c(Male = "#0a6ddd")`; they override the colour set for the number of groups and are not remembered for other charts.
`order`	Top-level slices in this order; any others follow in their data order.

Value

A ggplot with the hairline frame.

Examples

```
discharge <- data.frame(route = c("Home", "Rehabilitation", "Ward", "Died"),
                        patients = c(412, 168, 97, 23))
hl_donut(discharge, "route", "patients", total_label = "patients")
```

`hl_dotplot` *Dot plot with intervals*

Description

One dot per row, largest at the top. With `low` and `high` each dot gets its interval and the numbers on the right; `reference` adds a vertical line.

Usage

```
hl_dotplot(
  data,
  label,
  estimate,
  low = NULL,
  high = NULL,
  title = NULL,
  subtitle = NULL,
  caption = NULL,
  sort = TRUE,
  ratio = FALSE,
  reference = NULL,
  digits = NULL
)
```

Arguments

`data`	A data frame.
`label`, `estimate`	Columns: row label and value.
`low`, `high`	Interval columns, optional.
`title`, `subtitle`, `caption`	Frame texts; the title defaults to the columns.
`sort`	Sort by value, largest at the top.
`ratio`	Values are ratios (log axis).
`reference`	Value for a vertical reference line.
`digits`	Decimals of the numbers; `NULL` gives two, or more for small or close values.

Value

A ggplot with the hairline frame.

Examples

```
rates <- data.frame(unit = c("Medical", "Surgical", "Cardiac"), mortality = c(0.13, 0.08, 0.06),
                    low = c(0.11, 0.07, 0.04), high = c(0.15, 0.10, 0.08))
hl_dotplot(rates, "unit", "mortality", "low", "high")
```

`hl_effects` *Effects with 95 % intervals*

Description

From a fitted `lm`, `glm` or `coxph` model, or a table with estimate and interval columns; terms that are study columns take their labels.

Usage

```
hl_effects(
  study,
  model_or_table,
  measure = NULL,
  study_column = "study",
  estimate = "estimate",
  low = "low",
  high = "high",
  exponentiate = NULL,
  digits = 2,
  ...
)
```

Arguments

`study`	A study from `hl_study()`.
`model_or_table`	A fitted model or a data frame.
`measure`	Prefix for the values and text, e.g. `"OR"`.
`study_column, estimate, low, high`	Columns of a table.
`exponentiate`	Exponentiate model terms; default for `glm` and `coxph`.
`digits`	Decimals of the estimates.
`...`	Passed to `hl_forest()`.

Value

A `hairline_result`.

Examples

```
set.seed(1)
cohort <- data.frame(arm = rep(c("Standard care", "Early mobilisation"), each = 40),
                     age = round(stats::rnorm(80, 64, 12)),
                     sex = sample(c("Female", "Male"), 80, TRUE),
                months = round(stats::rexp(80, 0.05), 1), died = stats::rbinom(80, 1, 0.4))
study <- hl_study(cohort, group = "arm", time = "months", event = "died",
                  units = c(age = "years", months = "months"))
model <- stats::glm(died ~ age + sex, family = stats::binomial, data = cohort)
hl_effects(study, model, "OR")$values
```

`hl_find_brand` *Find a brand file*

Description

Looks for `_brand.yml` from `start` upwards and stops at the project root (a folder containing `.git`, `_quarto.yml`, `pyproject.toml` or `DESCRIPTION`).

Usage

```
hl_find_brand(start = getwd())
```

Arguments

`start`	Folder to start from.

Value

The path, or `NULL`.

Examples

```
path <- file.path(tempdir(), "_brand.yml")
writeLines(c("meta:", "  name: Example Hospital", "color:", "  palette:",
           "    harbor: \"#0b3d6e\"", "    poppy: \"#f2545b\"", "defaults:", "  hairline:",
             "    categorical: [harbor, poppy]"), path)
hl_find_brand(tempdir())
```

`hl_forest` *Forest plot*

Description

A results table (study, extra columns, effect with 95 % interval, weight) beside the intervals: squares sized by weight, a pooled diamond, a solid null line, a dashed pooled line, a log axis for ratios and "favours" labels under it. Heterogeneity goes into the caption. On a figure too narrow for the columns beside the intervals, they sit on a line under each study.

Usage

```
hl_forest(
  data,
  study = "study",
  estimate = "estimate",
  low = "low",
  high = "high",
  weight = NULL,
  columns = character(),
  pooled = NULL,
  ratio = TRUE,
  favours = NULL,
  heterogeneity = NULL,
  title = NULL,
  subtitle = NULL,
  caption = NULL,
  exponentiate = NULL,
  ...
)
```

Arguments

`data`	A data frame of results, a fitted `lm`, `glm` or `survival::coxph` model, or a `metafor::rma` object.
`study`, `estimate`, `low`, `high`	Columns: study label, effect and its interval limits.
`weight`	Column of weights (any scale); squares get areas proportional to it.
`columns`	Named character vector of extra text columns, `c("Events / n" = "events")`.
`pooled`	`list(estimate = , low = , high = , label = )` for the pooled row.
`ratio`	Effects are ratios (log axis, null line at 1); `FALSE` for differences.
`favours`	Two labels for the left and right of the null line.
`heterogeneity`	Text put at the start of the caption.
`title`, `subtitle`, `caption`	Frame texts.

exponentiate	For fitted models: report exp(coefficient) (odds, rate or hazard ratios); NULL decides from the model (logit and log links, Cox models).
...	study_heading, effect_heading, weight_heading, digits.

Value

A ggplot with the hairline frame.

Examples

```
trials <- data.frame(study = c("Trial A", "Trial B", "Trial C"), or = c(0.62, 0.81, 0.47),
              low = c(0.41, 0.55, 0.25), high = c(0.94, 1.19, 0.88), weight = c(40, 35, 25))
hl_forest(trials, "study", "or", "low", "high", weight = "weight",
          pooled = list(estimate = 0.66, low = 0.52, high = 0.84, label = "Pooled"))
```

---

hl_format_number *Format numbers, percentages and intervals*

Description

Fixed decimals, locale separators ("en" or "de") and a true minus sign.

Usage

```
hl_format_number(value, digits = 1, locale = NULL)

hl_format_percent(share, digits = 1, locale = NULL)

hl_format_interval(estimate, low, high, digits = 2, locale = NULL)
```

Arguments

value, share, estimate, low, high	Numbers; share is a proportion (0.614 for 61.4 %).
digits	Decimal places.
locale	"en", "de", or NULL for the active setting.

Value

A character string.

Examples

```
hl_format_number(1234.5, 1)
hl_format_number(-0.25, 2, locale = "de")
hl_format_percent(0.125, 1)
hl_format_interval(0.62, 0.41, 0.94)
```

`hl_funnel` *Funnel plot*

Description

Effect against standard error (running down from 0) with pseudo 95 % limits around the pooled estimate, a dashed pooled line and a solid null line; the pooled estimate and Egger's test become the key numbers.

Usage

```
hl_funnel(
  data,
  estimate,
  se,
  title = NULL,
  subtitle = NULL,
  caption = NULL,
  ratio = TRUE,
  pooled = NULL,
  x_title = NULL
)
```

Arguments

`data`	A data frame.
`estimate`, `se`	Columns: effect and its standard error (on the log scale for ratios).
`title`, `subtitle`, `caption`	Frame texts; the title defaults to the axis.
`ratio`	Effects are ratios (log axis, null at 1).
`pooled`	Pooled estimate; inverse-variance weighted when `NULL`.
`x_title`	Axis title; defaults to the effect column.

Value

A ggplot with the hairline frame.

Examples

```
trials <- data.frame(odds_ratio = c(0.84, 0.67, 0.91, 0.72, 0.55),
                     se = c(0.12, 0.27, 0.09, 0.2, 0.35))
hl_funnel(trials, "odds_ratio", "se")
```

hl_highlight *Highlight series or panels*

Description

Keeps the named series (legend labels of the colour or fill mapping) or panels (facet label or 1-based index) in their colours and draws every other data series in the context grey.

Usage

```
hl_highlight(series = NULL, panels = NULL)
```

Arguments

series	Legend labels to keep in colour.
panels	Facet labels or 1-based panel numbers to keep in colour.

Value

An object to add to a ggplot with +.

Examples

```
data <- data.frame(day = rep(1:10, 3),
                   share = c(seq(0.6, 0.9, length.out = 10), rep(0.7, 10), rep(0.8, 10)),
                    unit = rep(c("Unit A", "Unit B", "Unit C"), each = 10))
ggplot2::ggplot(data, ggplot2::aes(day, share, colour = unit)) + ggplot2::geom_line() +
  ggplot2::labs(title = "Occupancy") + theme_hairline() + hl_highlight("Unit A")
```

hl_install_font *Install the bundled font for PDF output*

Description

Copies the bundled 'Hairline Sans' files into your user font folder so that every PDF route (cairo, rsvg, Quartz) can embed it. Nothing is installed unless you call this function.

Usage

```
hl_install_font(confirm = interactive())
```

Arguments

confirm	Ask before copying (default in interactive sessions).

Value

The installed file paths, invisibly.

Examples

```
if (interactive()) hl_install_font()
```

---

hl_kaplan_meier *Kaplan–Meier curves*

Description

Step curves per group with 95 % bands (Greenwood, log scale) and censor ticks, a dotted 50 % line, medians and your statistics in the panel header, and the number at risk under the x-axis aligned to its ticks. A figure whose frame would leave the curves less than 8 x the base size high (a narrow figure with a number-at-risk table) grows taller.

Usage

```
hl_kaplan_meier(
  data,
  time = NULL,
  event = NULL,
  by = NULL,
  title = NULL,
  subtitle = NULL,
  caption = NULL,
  x_title = "Time",
  endpoint = "Survival",
  median_line = TRUE,
  stats = character(),
  at_risk = TRUE,
  times = NULL,
  order = NULL,
  test = TRUE,
  colors = NULL
)
```

Arguments

`data`	A data frame, or a `survival::survfit` object.
`time, event`	Columns with follow-up time and event (1 = event, 0 = censored); unused for `survfit`. `time` may also be a formula `Surv(time, event) ~ group`.
`by`	Grouping column; one curve per group.
`title, subtitle, caption`	Frame texts.

x_title	Axis title under the x-axis.
endpoint	Panel header name, e.g. "Overall survival".
median_line	Draw the dotted 50 % line.
stats	Named character vector of key numbers, e.g. c(HR = "0.64 [0.45, 0.91]").
at_risk	Show the number at risk.
times	Times for the number at risk; default the x-axis breaks.
order	Group order; groups left out are neither drawn nor part of the log-rank test.
test	Add the log-rank p to the key numbers when there are groups.
colors	Named colours for this chart only, e.g. c(Male = "#0a6ddd"); they override the colour set for the number of groups and are not remembered for other charts.

Value

A ggplot with the hairline frame.

Examples

```
set.seed(1)
trial <- data.frame(months = round(stats::rexp(60, 0.06), 1), died = stats::rbinom(60, 1, 0.6),
                    arm = rep(c("Standard care", "Early mobilisation"), 30))
hl_kaplan_meier(trial, "months", "died", by = "arm", x_title = "Months since discharge")
```

---

hl_layout_report *Frame positions of a hairline plot*

Description

Lays out the plot at the given size and returns the positions hairline computes (pt from the top-left corner); the same keys as the Python package's layout_report(), used for cross-language parity checks.

Usage

```
hl_layout_report(plot, width_mm = NULL, height_mm = NULL, size = NULL)
```

Arguments

plot	A ggplot with theme_hairline().
width_mm, height_mm	Figure size in mm; override the preset.
size	Size preset, as in hl_save() (NULL for the plot's own); its base size applies unless one was set explicitly.

Value

A named list.

Examples

```
plot <- ggplot2::ggplot(datasets::mtcars, ggplot2::aes(wt, mpg)) + ggplot2::geom_point() +
  ggplot2::labs(title = "Fuel use by weight") + theme_hairline()
str(hl_layout_report(plot, size = "single")$panels)
```

---

hl_legend *Explicit legend items*

Description

Replaces the legend row collected from the plot's colour and fill scales with these items, e.g. to explain line kinds instead of groups.

Usage

```
hl_legend(items)
```

Arguments

items	List of `list(kind, colour, label)`; `kind` is one of `"line"`, `"thin"`, `"dash"`, `"band"`, `"dot"`, `"square"`.

Value

An object to add to a ggplot with `+`.

Examples

```
ggplot2::ggplot(datasets::mtcars, ggplot2::aes(wt, mpg)) + ggplot2::geom_point() +
  ggplot2::labs(title = "Fuel use by weight") + theme_hairline() +
  hl_legend(list(list(kind = "dot", colour = hl_palette(n = 1), label = "Car")))
```

---

hl_load_brand *Read a brand file*

Description

Parses a `_brand.yml` file in the brand.yml format. Errors name the file, the key path and the offending value.

Usage

```
hl_load_brand(path, output = NULL)
```

Arguments

path	Path to _brand.yml, or several paths: layers from institution to document, later layers overriding earlier ones.
output	Export type whose overrides apply: "screen", "pdf", "html" or "docx".

Value

A list with name, path, names, colors_given, colors_used, adapt, font, heading_font, heading_weight, neutral, base_size, sizes, locale, entity_colors, entity_failures, output, sequential_hue, diverging_hues, layers (the file paths), slot_origins and entity_origins (the layer that set each colour).

Examples

```
path <- file.path(tempdir(), "_brand.yml")
writeLines(c("meta:", "  name: Example Hospital", "color:", "  palette:",
           "    harbor: \"#0b3d6e\"", "    poppy: \"#f2545b\"", "defaults:", "  hairline:",
             "    categorical: [harbor, poppy]"), path)
brand <- hl_load_brand(path, output = "pdf")
brand$name
```

---

hl_over_time *One measure over time by group*

Description

Small multiples per group with the raw values and a trailing mean; the table gives n, mean (SD) and median (IQR) over the whole period. ISO date text (2025-01-31) is read as dates.

Usage

```
hl_over_time(study, date, value, window = 7)
```

Arguments

study	A study from hl_study().
date	Column with dates (or numbers on a time axis).
value	Numeric column to follow.
window	Observations in the trailing mean.

Value

A hairline_result.

Examples

```
set.seed(1)
visits <- data.frame(day = 1:60, lactate = round(2 + sin(1:60 / 8) + stats::rnorm(60, 0, 0.3), 1))
study <- hl_study(visits, units = c(day = "days", lactate = "mmol/l"))
hl_over_time(study, "day", "lactate", window = 7)$text
```

`hl_palette` *Palette colours in slot order*

Description

Colours for groups, in the order hairline assigns them: two groups take slots 1-2, three groups slots 1-3, and a group keeps its colour when more are shown. There are six categorical palettes of eight slots, each opening blue then red so two groups always read blue against red (`"daylight"`, the default, `"calm"`, `"deep"`, `"lagoon"`, `"citrus"` and `"bold"`), and purpose sets for common cases: `"pair"`, `"pair-teal"` and `"pair-print"` for two groups, `"ink"` for greyscale print and `"judgement"` for low, some concerns and high. `hl_palettes()` lists them all with their uses and `hl_swatches()` draws them; make one the default with `hl_use(palette = )`. For amounts, use a ramp (`hl_ramp()`) instead.

Usage

```
hl_palette(name = NULL, n = NULL)
```

Arguments

`name`	Palette or purpose-set name, a vector of hex colours, or `NULL` for the active palette.
`n`	Number of categories; raises an error rather than repeating colours.

Value

Character vector of hex colours.

See Also

`hl_palettes()`, `hl_swatches()`, `hl_check_palette()`, `hl_use()`.

Examples

```
hl_palette()
hl_palette(n = 3)
hl_palette("pair")
hl_palette("calm", n = 4)
```

hl_palettes *Every colour set, with its use*

Description

The catalogue of hairline's colours: categorical palettes and purpose sets for groups, sequential and diverging ramps for amounts. `active` marks the palette in use; a palette from a brand or `hl_use(palette = c(...))` comes first. Use a name with `hl_palette()`, `hl_use(palette = )` or `hl_ramp()`.

Usage

```
hl_palettes(kind = NULL)
```

Arguments

`kind`	`NULL` for all, or one of `"categorical"`, `"purpose"`, `"sequential"`, `"diverging"`.

Value

A data frame with `name`, `kind`, `slots`, `active`, `use` and `colors` (hex codes separated by spaces).

See Also

`hl_swatches()` draws them.

Examples

```
hl_palettes()
hl_palettes("purpose")
```

hl_panel_header *Panel header with key numbers*

Description

Adds a header line above a panel: the name, an optional qualifier, and key numbers right-aligned. When the numbers do not fit they move to a second line; numbers that still do not fit are dropped with a warning.

Usage

```
hl_panel_header(
  name = NULL,
  qualifier = NULL,
  stats = NULL,
  panel = NULL,
  key = NULL,
  x_title = NULL
)
```

Arguments

name	Header name. For faceted plots, defaults to the facet label of panel.
qualifier	Muted text after the name, e.g. "24 beds".
stats	Named character vector of key numbers, e.g. c(Mean = "61.4%", Median = "61.3%").
panel	Facet label the header belongs to; NULL for a single-panel plot.
key	Name of the one key number to emphasise, e.g. "Days over 90%".
x_title	Title under this panel's own x-axis numbers, e.g. its unit when panels measure different things.

Value

An object to add to a ggplot with +.

Examples

```
data <- data.frame(day = rep(1:10, 2), share = c(seq(0.6, 0.9, length.out = 10), rep(0.7, 10)),
                   unit = rep(c("Unit A", "Unit B"), each = 10))
ggplot2::ggplot(data, ggplot2::aes(day, share)) + ggplot2::geom_line() +
  ggplot2::facet_wrap(~unit) +
  ggplot2::labs(title = "Occupancy") + theme_hairline() +
  hl_panel_header(panel = "Unit A", qualifier = "12 beds", stats = c(Mean = "75 %")) +
  hl_panel_header(panel = "Unit B", qualifier = "8 beds", stats = c(Mean = "70 %"))
```

---

hl_partition *Partition bars*

Description

A hierarchy as stacked rows, one per level, children directly under their parent; the documented alternative to a nested donut. Labels sit inside the bars (name and share, else the share, else none); a legend row names the top level.

Usage

```
hl_partition(
  data,
  levels,
  value = NULL,
  title = NULL,
  subtitle = NULL,
  caption = NULL,
  level_names = NULL,
  colors = NULL,
  order = NULL
)
```

Arguments

`data`	A data frame.
`levels`	Hierarchy columns, top level first (at most three).
`value`	Column with the amounts; `NULL` counts rows.
`title`, `subtitle`, `caption`	Frame texts.
`level_names`	Row headings; default the column names.
`colors`	Named colours for this chart only, e.g. `c(Male = "#0a6ddd")`; they override the colour set for the number of groups and are not remembered for other charts.
`order`	Top-level slices in this order; any others follow in their data order.

Value

A ggplot with the hairline frame.

Examples

```
discharge <- data.frame(route = c("Home", "Home", "Rehabilitation", "Ward"),
                        support = c("Alone", "With support", "Inpatient", "Ward"),
                        patients = c(250, 162, 168, 97))
hl_partition(discharge, c("route", "support"), "patients",
             level_names = c("Destination", "Support"))
```

---

hl_prisma *PRISMA 2020 flow diagram*

Description

Identified, screened, sought, assessed and included records with the exclusions beside them, phases as brackets on the left. Counts that do not add up give a warning.

Usage

```
hl_prisma(counts, title = NULL, subtitle = NULL, caption = NULL)
```

Arguments

counts	Named list: identified (and registers), duplicates, automation, other_removed, screened, excluded, sought, not_retrieved, assessed, excluded_reasons (named numbers), included, reports.
title, subtitle, caption	Frame texts.

Value

A ggplot with the hairline frame.

Examples

```
hl_prisma(list(identified = 1240, duplicates = 260, screened = 980, excluded = 860, sought = 120,
          not_retrieved = 10, assessed = 110, excluded_reasons = c("Wrong population" = 92),
               included = 18))
```

---

hl_raincloud *Raincloud plot*

Description

Per group a half density, the box (interquartile range and median) and the points, with "n = ..." under the group names. Comparison brackets show your test results and are stacked so they never overlap. Missing values are left out, and "n = ..." counts the values shown.

Usage

```
hl_raincloud(
  data,
  group,
  value = NULL,
  title = NULL,
  subtitle = NULL,
  caption = NULL,
  y_title = NULL,
  order = NULL,
  comparisons = list(),
  colors = NULL
)
```

Arguments

data	A data frame.
group, value	Columns: the grouping and the measurement; or a formula value ~ group as group.
title, subtitle, caption	Frame texts.
y_title	Axis title above the y-axis.
order	Group order.
comparisons	"all" or "adjacent" pairs, or a list of c(group, group) pairs (Wilcoxon rank-sum p computed) or c(group, group, text) with your own result.
colors	Named colours for this chart only, e.g. c(Male = "#0a6ddd"); they override the colour set for the number of groups and are not remembered for other charts.

Value

A ggplot with the hairline frame.

Examples

```
set.seed(1)
stays <- data.frame(unit = rep(c("Surgical", "Medical"), each = 30),
                    days = round(stats::rexp(60, 0.2), 1))
hl_raincloud(stays, "unit", "days", y_title = "Days")
```

---

hl_ramp *Sequential or diverging ramp*

Description

Colours for amounts. Sequential ramps run light to dark in one hue ("blues", "teals", "oranges", "berries", "greys"); diverging ramps run from one pole through a neutral midpoint to the other ("blue-coral", "teal-berry"), for values below and above a reference. scale_colour_hairline_c() and scale_fill_hairline_c() use them in a plot. While a brand is active, "brand-sequential" and "brand-diverging" follow its colours (its sequential and diverging entries, else its first one and two colours) and become the default.

Usage

```
hl_ramp(name = NULL, n = NULL)
```

Arguments

name	Ramp name, e.g. "blues" or "blue-coral"; NULL for the default ramp (the brand's while one is active, else "blues").
n	Number of colours; other counts than the stored 7 (or 13) interpolate in OKLab.

Value

Character vector of hex colours.

See Also

hl_palettes(), hl_swatches().

Examples

```
hl_ramp("blues")
hl_ramp("blues", n = 3)
hl_ramp("blue-coral", n = 5)
```

---

hl_reference *Reference line with a label*

Description

A thin dashed line (threshold, target) behind the data in every panel, with an optional label at its end and an optional legend entry.

Usage

```
hl_reference(y = NULL, x = NULL, label = NULL, legend_label = NULL)
```

Arguments

y, x	Position of a horizontal (y) or vertical (x) line; give exactly one.
label	Text at the end of the line, e.g. "90 %".
legend_label	Text for a dashed legend entry.

Value

An object to add to a ggplot with +.

Examples

```
ggplot2::ggplot(datasets::airquality, ggplot2::aes(Day, Temp)) + ggplot2::geom_point() +
  ggplot2::labs(title = "Daily temperature") + theme_hairline() +
  hl_reference(y = 80, label = "80 F")
```

`hl_ridgeline` *Ridgeline plot*

Description

One density ridge per group on a shared x-axis, the first group on top, each overlapping the row above; all ridges share one height scale, and a short line marks each median.

Usage

```
hl_ridgeline(
  data,
  group,
  value = NULL,
  title = NULL,
  subtitle = NULL,
  caption = NULL,
  order = NULL,
  medians = TRUE,
  colors = NULL
)
```

Arguments

`data`	A data frame.
`group`, `value`	Columns: the grouping and the measurement; or a formula `value ~ group` as `group`.
`title`, `subtitle`, `caption`	Frame texts; the title defaults to the columns.
`order`	Group order, top to bottom.
`medians`	Mark the medians.
`colors`	Named colours for this chart only, e.g. `c(Male = "#0a6ddd")`; they override the colour set for the number of groups and are not remembered for other charts.

Value

A ggplot with the hairline frame.

Examples

```
set.seed(1)
diagnoses <- c("Sepsis", "Trauma", "Cardiac", "Respiratory", "Neuro")
ages <- data.frame(diagnosis = rep(diagnoses, each = 40),
                   age = round(stats::rnorm(200, rep(c(68, 45, 70, 62, 55), each = 40), 10)))
hl_ridgeline(ages, "diagnosis", "age")
```

`hl_risk_of_bias` *Risk-of-bias summary*

Description

One 100 % bar per domain with the number of studies judged low, some concerns or high inside, and an overall row (each study's worst judgement) below a rule. Reads a long table (study, domain, judgement) or a wide one with one column per domain.

Usage

```
hl_risk_of_bias(
  data,
  study,
  domain = NULL,
  judgement = NULL,
  domains = NULL,
  overall = "Overall",
  title = NULL,
  subtitle = NULL,
  caption = NULL
)
```

Arguments

`data`	A data frame.
`study`	Study column.
`domain`, `judgement`	Columns of a long table.
`domains`	Domain columns of a wide table, in place of `domain` and `judgement`.
`overall`	Name of the overall row; `NULL` leaves it out.
`title`, `subtitle`, `caption`	Frame texts.

Value

A ggplot with the hairline frame.

Examples

```
trials <- data.frame(study = c("Trial 1", "Trial 2"), Randomisation = c("Low", "Some concerns"),
                     Reporting = c("Low", "High"), check.names = FALSE)
hl_risk_of_bias(trials, "study", domains = c("Randomisation", "Reporting"))
```

hl_sankey *Sankey diagram*

Description

Flows from left to right through two or more columns. Nodes share one scale; later columns are ordered to reduce crossings and links are ordered at both ends, so they never cross inside a node. Links take the colour of their group in `color_by` (default the last column).

Usage

```
hl_sankey(
  data,
  columns,
  value = NULL,
  title = NULL,
  subtitle = NULL,
  caption = NULL,
  color_by = NULL,
  headings = NULL,
  colors = NULL
)
```

Arguments

`data`	A data frame, one row per path with its value.
`columns`	The columns, left to right.
`value`	Column with the amounts; `NULL` counts rows.
`title, subtitle, caption`	Frame texts.
`color_by`	Column whose groups colour the links, or `"none"`.
`headings`	Column headings; default the column names.
`colors`	Named colours for this chart only, e.g. `c(Male = "#0a6ddd")`; they override the colour set for the number of groups and are not remembered for other charts.

Value

A ggplot with the hairline frame.

Examples

```
budget <- data.frame(income = c("Salary", "Salary", "Benefit"),
                     account = c("Joint", "Joint", "Savings"),
                     spending = c("Rent", "Food", "Reserve"), euros = c(1200, 420, 150))
hl_sankey(budget, c("income", "account", "spending"), "euros")
```

`hl_save` *Save print-ready files*

Description

Writes `name.<format>` for each format: PDF (vector), PNG and TIFF (600 dpi by default, TIFF LZW-compressed) and SVG (text kept as text). Every file is named by the plot's title, with the title and subtitle as their description; none shows a timestamp or producer banner. PDF uses cairo where it is available; otherwise svglite + rsvg. Both embed the bundled font once it can be found: cairo finds it on its own outside Windows, otherwise run `hl_install_font()` once. PDFs are byte-identical across runs with cairo before 1.17.6; later versions keep a creation date inside a compressed stream that no PDF reader shows.

Usage

```
hl_save(
  plot,
  name,
  formats = c("pdf", "png"),
  size = NULL,
  width_mm = NULL,
  height_mm = NULL,
  dpi = 600,
  background = NULL
)
```

Arguments

`plot`	A ggplot, ideally with `theme_hairline()`.
`name`	Output path without extension; parent folders are created.
`formats`	Any of `"pdf"`, `"png"`, `"svg"`, `"tiff"`, `"html"` (the SVG with tooltips and series highlighting, self-contained).
`size`	Size preset: `"single"` (85 mm), `"onehalf"` (120), `"double"` (174), `"a4"` (160), `"slide"`. `NULL` uses the plot's own size: `"double"` for `hl_timeseries()` (its panels run in rows across the page, as in the Python package), `"onehalf"` for every other plot.
`width_mm, height_mm`	Explicit size in mm; overrides the preset.
`dpi`	Resolution for PNG and TIFF.
`background`	`NULL` for the plot's background or `"transparent"`.

Value

The written paths, invisibly.

Examples

```
plot <- ggplot2::ggplot(datasets::mtcars, ggplot2::aes(wt, mpg)) + ggplot2::geom_point() +
  ggplot2::labs(title = "Fuel use by weight") + theme_hairline()
hl_save(plot, file.path(tempdir(), "fuel"), formats = "png", size = "single", dpi = 150)
```

`hl_scatter` *Scatter plot with fit*

Description

Points per group in their entity colours and, with `fit = "linear"`, a linear fit over all points with its 95 % band; n, r and slope become the key numbers.

Usage

```
hl_scatter(
  data,
  x,
  y,
  by = NULL,
  title = NULL,
  subtitle = NULL,
  caption = NULL,
  fit = "linear",
  order = NULL,
  colors = NULL
)
```

Arguments

`data`	A data frame.
`x`, `y`	Columns for the axes.
`by`	Grouping column.
`title`, `subtitle`, `caption`	Frame texts; the title defaults to the columns.
`fit`	`"linear"` or `NULL`.
`order`	Group order.
`colors`	Named colours for this chart only, e.g. `c(Male = "#0a6ddd")`; they override the colour set for the number of groups and are not remembered for other charts.

Value

A ggplot with the hairline frame.

Examples

```
hl_scatter(datasets::mtcars, "wt", "mpg", title = "Fuel use by weight")
```

hl_settings *Active hairline settings*

Description

Active hairline settings

Usage

```
hl_settings()
```

Value

The settings list set by the last `hl_use()`, or the defaults.

Examples

```
hl_settings()$font
```

hl_spread *Spread rows: distributions per group on one axis*

Description

Per group, first on top: the range as a thin line, the interquartile range as a band in the group's colour, the median as a dot, an optional reference tick, and the key numbers right-aligned. On a figure too narrow for the numbers beside the plot, they sit on a line under each row.

Usage

```
hl_spread(
  data,
  group,
  value,
  title = NULL,
  subtitle = NULL,
  caption = NULL,
  order = NULL,
  marks = NULL,
  mark_label = NULL,
  stats = c("median", "iqr", "mean"),
  percent = FALSE,
  digits = NULL,
  zero = FALSE,
  axis_title = NULL,
  colors = NULL
)
```

Arguments

`data`	A data frame.
`group, value`	Grouping column and numeric column.
`title, subtitle, caption`	Frame texts; the title defaults to the columns.
`order`	Group order, top to bottom.
`marks`	Named reference values per group, e.g. `c("Unit A" = 20)` (a capacity, a target).
`mark_label`	Name of the reference in the numbers column, e.g. `"beds"`.
`stats`	Numbers to show: any of `"median"`, `"iqr"`, `"mean"`, `"n"`, `"range"`.
`percent`	Values are shares (0-1), shown as percentages.
`digits`	Decimals; default by magnitude (none for whole numbers).
`zero`	Include zero on the axis.
`axis_title`	Title under the axis numbers, e.g. the unit.
`colors`	Named colours for this chart only.

Value

A ggplot with the hairline frame.

Examples

```
set.seed(1)
beds <- data.frame(unit = rep(c("Unit A", "Unit B", "Unit C"), each = 50),
            occupied = c(stats::rpois(50, 10), stats::rpois(50, 7), stats::rpois(50, 12)))
hl_spread(beds, "unit", "occupied", marks = c("Unit A" = 12, "Unit B" = 8, "Unit C" = 14),
          mark_label = "Beds",
          zero = TRUE, axis_title = "Occupied beds")
```

---

hl_study *Describe a dataset once for the study endpoints*

Description

Names the column that plays each role, runs the checks and returns a study for `hl_describe()`, `hl_distribution()`, `hl_compare()`, `hl_survival()` and `hl_effects()`. Checks report problems (print the study) and never change the data; errors block only the endpoints that need the column.

Usage

```
hl_study(
  data,
  group = NULL,
  time = NULL,
  event = NULL,
  id = NULL,
  variables = NULL,
  labels = NULL,
  types = NULL,
  order = NULL,
  unit = NULL,
  units = NULL
)
```

Arguments

`data`	A data frame, one row per participant.
`group`	Column with the comparison groups.
`time`, `event`	Follow-up time and event (1 = event, 0 = censored).
`id`	Participant id, checked for duplicates; `hl_describe()`, `hl_compare()` and `hl_distribution()` refuse a variable with more than one row per id.
`variables`	Characteristics to describe; default every other column.
`labels`	Named display names, e.g. `c(sofa = "SOFA score")`.
`types`	Named types overriding the inference, e.g. `c(score = "categorical")`.
`order`	Group order; default order of first appearance.
`unit`	What a row is, for the texts: a plural noun (default "participants"; one reads "1 participant") or a singular and plural pair, e.g. `c("mouse", "mice")`.
`units`	Named units of numeric columns, e.g. `c(age = "years", lactate = "mmol/l")`; tables, panel headers and texts name them.

Value

A `hairline_study`.

Examples

```
set.seed(1)
cohort <- data.frame(arm = rep(c("Standard care", "Early mobilisation"), each = 40),
                     age = round(stats::rnorm(80, 64, 12)),
                     sex = sample(c("Female", "Male"), 80, TRUE),
                months = round(stats::rexp(80, 0.05), 1), died = stats::rbinom(80, 1, 0.4))
study <- hl_study(cohort, group = "arm", time = "months", event = "died",
                  units = c(age = "years", months = "months"))
study
```

hl_survival *Survival by group*

Description

Kaplan–Meier curves with the log-rank test; the table gives n, events and the median per group. A fit already made (`survival::survfit`) can be passed as `fit`; it keeps its own intervals, and the log-rank test comes from the study's own time, event and group columns when they are set.

Usage

```
hl_survival(study, fit = NULL)
```

Arguments

`study`	A study from `hl_study()`.
`fit`	Optional `survival::survfit`.

Value

A `hairline_result`.

Examples

```
set.seed(1)
cohort <- data.frame(arm = rep(c("Standard care", "Early mobilisation"), each = 40),
                     age = round(stats::rnorm(80, 64, 12)),
                     sex = sample(c("Female", "Male"), 80, TRUE),
                months = round(stats::rexp(80, 0.05), 1), died = stats::rbinom(80, 1, 0.4))
study <- hl_study(cohort, group = "arm", time = "months", event = "died",
                  units = c(age = "years", months = "months"))
hl_survival(study)$values
```

hl_swatches *Colour sets as swatches*

Description

Draws palettes, purpose sets and ramps as rows of squares under their headings, each with its use, to choose colours by eye. In HTML output each square shows its slot and hex code on hover.

Usage

```
hl_swatches(names = NULL, title = NULL, subtitle = NULL, caption = NULL)
```

Arguments

names	Colour sets to show, as hl_palettes() lists them; NULL shows all, in catalogue order.
title, subtitle, caption	Frame texts; the title defaults to "Colour sets".

Value

A ggplot with the hairline frame.

See Also

hl_palettes(), hl_palette(), hl_ramp().

Examples

```
hl_swatches()
hl_swatches(c("daylight", "calm", "pair", "blues"), title = "Colours for this report")
```

---

hl_timeseries *Small-multiple time series with two levels of detail*

Description

One panel per group: raw values as a thin line under a trailing rolling mean in the group's entity colour, with key numbers in the panel headers, and optional period mean, threshold, highlighted periods and events.

Usage

```
hl_timeseries(
  data,
  x,
  y,
  by = NULL,
  title = NULL,
  subtitle = NULL,
  caption = NULL,
  window = 7,
  raw = TRUE,
  period_mean = FALSE,
  percent = FALSE,
  threshold = NULL,
  threshold_label = NULL,
  threshold_legend = NULL,
  stats = c("mean", "median"),
  share_label = NULL,
```

```
  key = NULL,
  key_panels = NULL,
  highlight = NULL,
  qualifiers = list(),
  bands = list(),
  events = list(),
  order = NULL,
  raw_label = "Raw values",
  mean_label = NULL,
  colors = NULL
)
```

Arguments

`data`	A data frame.
`x`, `y`, `by`	Column names: time, value and the grouping (one panel per group).
`title`, `subtitle`, `caption`	Frame texts.
`window`	Rolling-mean window in rows (trailing, partial windows at the start).
`raw`	Draw the raw values.
`period_mean`	Draw each panel's mean as a dashed line.
`percent`	Values are shares; label axis and key numbers as percentages.
`threshold`, `threshold_label`, `threshold_legend`	Reference line, its label and legend entry.
`stats`	Key numbers: any of `"mean"`, `"median"`, `"range"`, `"share_above"`.
`share_label`	Label of the `"share_above"` key number.
`key`, `key_panels`	The key number to emphasise and the panels to emphasise it in.
`highlight`	Panels to keep in colour; the others become context grey.
`qualifiers`	Named list of qualifier texts per group, e.g. bed counts.
`bands`	List of `list(start, end, label)` highlighted periods.
`events`	List of `list(x, label)` event markers.
`order`	Group order; defaults to the order of appearance.
`raw_label`, `mean_label`	Legend labels of the two lines.
`colors`	Named colours for this chart only, e.g. `c(Male = "#0a6ddd")`; they override the colour set for the number of groups and are not remembered for other charts.

Value

A ggplot with the hairline frame; `hl_save()` writes it at the `"double"` size (174 mm) unless told otherwise, as the Python package does.

Examples

```
set.seed(1)
occupancy <- data.frame(date = rep(as.Date("2025-01-01") + 0:89, 2),
                        unit = rep(c("Unit A", "Unit B"), each = 90),
                        share = pmin(1, c(0.8 + stats::rnorm(90, 0, 0.05),
                                          0.7 + stats::rnorm(90, 0, 0.05))))
hl_timeseries(occupancy, "date", "share", by = "unit", percent = TRUE, threshold = 0.9,
              threshold_label = "90 %")
```

`hl_use` *Set hairline's style for this session*

Description

Precedence: these arguments > the brand (`brand` = path, or a `_brand.yml` found from the working directory up to the project root; `brand` = `FALSE` disables the lookup) > hairline's defaults.

Usage

```
hl_use(
  font = NULL,
  base_size = NULL,
  sizes = NULL,
  palette = NULL,
  locale = NULL,
  background = NULL,
  brand = NULL,
  colors = NULL,
  output = NULL,
  accent = NULL,
  scheme = NULL
)
```

Arguments

`font`	Font family; falls back to the bundled 'Hairline Sans' with a warning when missing.
`base_size`	Body text size in pt; every other size derives from it.
`sizes`	Named list of role sizes in pt (`title`, `subtitle`, `panel`, `label`, `caption`).
`palette`	Palette or purpose-set name (e.g. `"calm"`, `"pair"`; see `hl_palettes()`), or a vector of hex colours.
`locale`	`"en"` or `"de"`.
`background`	`"#ffffff"`-style colour or `"transparent"`.
`brand`	Path to a `_brand.yml`, several paths as layers (institution first), `NULL` to look one up, or `FALSE`.

colors	Named vector of entity colours, e.g. c("Unit B" = "#e79d01"); see hl_color().
output	Export type whose brand overrides apply: "screen", "pdf", "html" or "docx".
accent	Colour of a lone series (charts without groups), made readable against the background.
scheme	"light" (default) or "dark": the dark neutral tokens (a brand's light neutrals do not carry over).

Value

The active settings, invisibly.

Examples

```
hl_use(base_size = 10, locale = "de", accent = "#0b6e4f")
hl_settings()$locale
hl_use()
```

---

hl_venn *Venn diagram*

Description

The overlap of two or three sets with the count of every region inside it and each set's name and size outside. Circles are schematic (equal sizes); with proportional = TRUE, two circles get areas proportional to their sizes and overlap by the shared count. Four or more sets are refused: show the intersections as bars instead.

Usage

```
hl_venn(
  data,
  columns = NULL,
  title = NULL,
  subtitle = NULL,
  caption = NULL,
  colors = NULL,
  proportional = FALSE
)
```

Arguments

data	A named list of member vectors, or a data frame with one row per member.
columns	For a data frame, the membership columns (TRUE/FALSE or 0/1).
title, subtitle, caption	Frame texts; the title defaults to the set names.
colors	Named colours for this chart only; they override the colour set for the number of sets.
proportional	Areas proportional to the counts (two sets only).

Value

A ggplot with the hairline frame.

Examples

```
hl_venn(list(Diabetes = c(1, 2, 3, 5, 8), Hypertension = c(2, 3, 4, 8, 9, 10),
             Obesity = c(3, 5, 9, 11)))
```

---

scale_colour_hairline *Colour and fill scales from hairline palettes and ramps*

Description

Discrete scales give every level its entity colour (see `hl_color()`) when `palette` is `NULL`, or the palette's slots in order otherwise, and refuse more categories than slots; continuous scales use a sequential or diverging ramp. Charts made with hairline's own functions pick their colours already; these scales are for plots built with ggplot2 directly.

Usage

```
scale_colour_hairline(palette = NULL, ...)

scale_fill_hairline(palette = NULL, ...)

scale_colour_hairline_c(ramp = NULL, ...)

scale_fill_hairline_c(ramp = NULL, ...)
```

Arguments

`palette`	Palette or purpose-set name (see `hl_palettes()`), hex vector, or `NULL` for the active palette with entity colours.
`...`	Passed to the underlying ggplot2 scale.
`ramp`	Ramp name for continuous scales: `"blues"`, `"teals"`, `"oranges"`, `"berries"`, `"greys"`, or a diverging `"blue-coral"` or `"teal-berry"` for values on both sides of a midpoint; `NULL` for the default ramp (the brand's while one is active, else `"blues"`).

Value

A ggplot2 scale.

See Also

`hl_palettes()`, `hl_swatches()`.

Examples

```
ggplot2::ggplot(datasets::iris, ggplot2::aes(Sepal.Length, Petal.Length, colour = Species)) +
 ggplot2::geom_point() + ggplot2::labs(title = "Petal by sepal length") + theme_hairline() +
  scale_colour_hairline()
ggplot2::ggplot(ggplot2::faithfuld, ggplot2::aes(waiting, eruptions, fill = density)) +
 ggplot2::geom_raster() + ggplot2::labs(title = "Old Faithful eruptions") + theme_hairline() +
  scale_fill_hairline_c("teals")
```

---

scale_y_hairline *Position scales with nice breaks*

Description

Continuous position scales using hairline's 3–6 'nice' breaks, locale number formatting and limits that end on a break. Applied automatically to hairline plots; add explicitly to change `percent` or `target`.

Usage

```
scale_y_hairline(percent = FALSE, target = NULL, digits = NULL, ...)

scale_x_hairline(percent = FALSE, target = NULL, digits = NULL, ...)
```

Arguments

`percent`	Treat values as shares and label them as percentages (unit on the top label only).
`target`	Preferred number of breaks; `NULL` follows the panel rows (5, 4, then 3).
`digits`	Decimals of the values a reader looks up (tooltips of interactive output), so that 89.6 % never reads as 90 % next to a 90 % target; `NULL` follows the shared digits rule. Axis labels keep their own decimals, set by the step between breaks.
`...`	Passed to `ggplot2::scale_y_continuous()` or `ggplot2::scale_x_continuous()`.

Value

A ggplot2 scale.

Examples

```
shares <- data.frame(unit = c("A", "B", "C"), share = c(0.42, 0.65, 0.81))
ggplot2::ggplot(shares, ggplot2::aes(unit, share)) + ggplot2::geom_col() +
  ggplot2::labs(title = "Share of beds occupied") + theme_hairline() +
  scale_y_hairline(percent = TRUE)
```

`theme_hairline` *hairline theme and frame*

Description

Adds hairline's theme and frame to a ggplot: title, subtitle, legend row, panel headers and caption on one left edge, with the spacing scale applied wherever the plot is printed or saved.

Usage

```
theme_hairline(base_size = NULL, font = NULL, axes = TRUE)
```

Arguments

`base_size`	Body text size in pt; `NULL` uses the active setting.
`font`	Font family; `NULL` uses the active setting.
`axes`	`FALSE` draws panels without axes, ticks, grid or baseline (donut, Sankey and the like); with `coord_fixed()` the plot area keeps its shape, centred in the panel.

Value

An object to add to a ggplot with `+`.

Examples

```
ggplot2::ggplot(datasets::mtcars, ggplot2::aes(wt, mpg)) + ggplot2::geom_point() +
 ggplot2::labs(title = "Fuel use by weight", subtitle = "32 cars, 1974", caption = "Motor Trend.",
               x = "Weight (1000 lb)", y = "Miles per gallon") + theme_hairline()
```

Index

ggplot2::scale_x_continuous(), *46*
ggplot2::scale_y_continuous(), *46*

hairline_colors *(*hairline_colours*)*, 3
hairline_colours, 3
hl_arcs, 4
hl_bars, 5
hl_brand_report, 7
hl_brand_report(), *4*
hl_check_palette, 7
hl_check_palette(), *4*, *25*
hl_chord, 8
hl_color, 9
hl_color(), *4*, *44, 45*
hl_colors *(*hl_color*)*, 9
hl_colors(), *4*
hl_compare, 10
hl_compare(), *38, 39*
hl_correlation, 10
hl_describe, 11
hl_describe(), *12*, *38, 39*
hl_distribution, 12
hl_distribution(), *38, 39*
hl_donut, 13
hl_dotplot, 14
hl_effects, 15
hl_effects(), *38*
hl_find_brand, 16
hl_forest, 17
hl_forest(), *15*
hl_format_interval *(*hl_format_number*)*, 18
hl_format_number, 18
hl_format_percent *(*hl_format_number*)*, 18
hl_funnel, 19
hl_highlight, 20
hl_install_font, 20
hl_install_font(), *35*
hl_kaplan_meier, 21
hl_layout_report, 22
hl_legend, 23
hl_load_brand, 23
hl_load_brand(), *7*
hl_over_time, 24
hl_palette, 25
hl_palette(), *4*, *26*, *41*
hl_palettes, 26
hl_palettes(), *3, 4*, *25*, *31*, *41*, *43*, *45*
hl_panel_header, 26
hl_partition, 27
hl_prisma, 28
hl_raincloud, 29
hl_ramp, 30
hl_ramp(), *4*, *25, 26*, *41*
hl_reference, 31
hl_ridgeline, 32
hl_risk_of_bias, 33
hl_risk_of_bias(), *3*
hl_sankey, 34
hl_save, 35
hl_save(), *6*, *22*, *42*
hl_scatter, 36
hl_settings, 37
hl_spread, 37
hl_study, 38
hl_study(), *10–12*, *15*, *24*, *40*
hl_survival, 40
hl_survival(), *38*
hl_swatches, 40
hl_swatches(), *3, 4*, *25, 26*, *31*, *45*
hl_timeseries, 41
hl_timeseries(), *3*, *35*
hl_use, 43
hl_use(), *4*, *25*, *37*
hl_venn, 44

scale_colour_hairline, 45
scale_colour_hairline_c *(*scale_colour_hairline*)*, 45
scale_colour_hairline_c(), *30*

scale_fill_hairline
(scale_colour_hairline), 45
scale_fill_hairline_c
(scale_colour_hairline), 45
scale_fill_hairline_c(), *30*
scale_x_hairline (scale_y_hairline), 46
scale_y_hairline, 46

theme_hairline, 47
theme_hairline(), *22*, *35*