Package: mirrorslope (via r-universe)

October 28, 2026

Title Mirror Slope Plots for Two-Arm Clinical Trial Data

Version 0.2.1

Description Creates mirrored slope plots comparing two treatment arms over a single observation interval. Both arms share a common x-axis and converge at a central endpoint, making the between-arm gap the primary visual quantity. Returns standard 'ggplot2' objects for full extensibility. The display was designed by Kati Maharry and is described in Maharry and Sullivan (2026, manuscript in preparation); it was first used in the supplementary appendix of McDonald and colleagues (2026) <doi:10.1016/S0140-6736(26)01385-1>.

License MIT + file LICENSE

URL https://github.com/Sullivanstatistics/mirrorslope

BugReports https://github.com/Sullivanstatistics/mirrorslope/issues

Encoding UTF-8

Language en-GB

Roxygen list(markdown = TRUE)

Depends R (>= 4.0.0)

Imports ggplot2 (>= 3.4.0), rlang (>= 1.0.0), grDevices, stats, utils

Suggests testthat (>= 3.0.0), knitr, rmarkdown, patchwork

VignetteBuilder knitr

Config/roxygen2/version 8.0.0

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

Date/Publication 2026-10-19 15:44:51 UTC

RemoteUrl https://github.com/sullivanstatistics/mirrorslope

RemoteRef HEAD

RemoteSha 991729b0519ffb1190f23235685aa01e2b0d9533

Contents

add_difference 2
add_precision 3
add_summary 3
as.data.frame.mirror_data 4
mirror_palette 5
mirror_slope_plot 5
prep_mirror_data 9
print.mirror_data 10
summary.mirror_data 11
theme_mirror 11

Index **13**

`add_difference` *Attach the between-arm difference to a mirror_data object*

Description

Optional pipeline step. Stores the comparator-minus-reference difference and its optional precision and p-value on the comparator `SUMM` row, per the specification.

Usage

```
add_difference(
  mirror_data,
  estimate,
  se = NULL,
  ci_lower = NULL,
  ci_upper = NULL,
  ci_level = 0.95,
  p_value = NULL
)
```

Arguments

`mirror_data`	A `mirror_data` object that already has summary rows.
`estimate`	Numeric scalar. Difference, comparator minus reference, on the endpoint scale.
`se`	Numeric scalar or `NULL`. Standard error of the difference.
`ci_lower`, `ci_upper`	Numeric scalars or `NULL`. Difference CI bounds.
`ci_level`	Numeric. Confidence level for labelling. Default `0.95`.
`p_value`	Numeric scalar or `NULL`. P-value for the difference.

Value

The `mirror_data` object with difference fields populated on the comparator `SUMM` row.

See Also

add_summary, mirror_slope_plot

---

add_precision — *Attach per-arm precision (SE or CI) to a mirror_data object*

Description

Optional pipeline step. Adds standard errors or confidence intervals for each arm's estimate to the existing SUMM rows. Either se or ci may be supplied; if both are given, ci is used for display.

Usage

```
add_precision(mirror_data, se = NULL, ci = NULL, ci_level = 0.95)
```

Arguments

mirror_data	A mirror_data object that already has summary rows.
se	Named numeric vector of length 2 (one per arm), or NULL.
ci	A named list of length 2; each element a numeric c(lower, upper); or NULL.
ci_level	Numeric confidence level for labelling, default 0.95.

Value

The mirror_data object with precision columns populated on the SUMM rows.

See Also

add_summary

---

add_summary — *Attach group-level summary estimates to a mirror_data object*

Description

Second step of the pipeline. Adds one SUMM row per arm carrying the endpoint summary estimate, the baseline value the mean line is drawn from, and a free-text statistic label.

Usage

```
add_summary(mirror_data, estimates, stat_label = "Mean", baseline = NULL)
```

Arguments

`mirror_data`	A `mirror_data` object from `prep_mirror_data`.
`estimates`	Named numeric vector of length 2 giving the endpoint summary estimate per arm. Names must match the reference and comparator arm labels.
`stat_label`	Character. Free-text label for the estimate type, e.g. `"Mean"`, `"LS-Mean"`, `"Median"`. Default `"Mean"`.
`baseline`	Named numeric vector of length 2 giving the baseline value per arm from which each mean line is drawn. Defaults to `0` for both arms (appropriate for change-from-baseline outcomes). For absolute-scale outcomes, supply the group baseline estimates.

Details

This function accepts pre-computed estimates and is deliberately model-agnostic. A future release may add a method to populate estimates directly from a model object (e.g. an `emmGrid`); the named-vector interface here is forward-compatible with that addition.

Value

The `mirror_data` object with two `SUMM` rows added.

See Also

`prep_mirror_data`, `add_difference`

`as.data.frame.mirror_data` *Coerce a mirror_data object to a plain data frame*

Description

Strips the `mirror_data` class and metadata, returning the underlying tabular structure as an ordinary data frame.

Usage

```
## S3 method for class 'mirror_data'
as.data.frame(x, ...)
```

Arguments

`x`	A `mirror_data` object.
`...`	Unused.

Value

A plain data frame.

mirror_palette — *Color palettes for mirror slope plots*

Description

Returns a length-2 character vector of hex colors for the reference and comparator arms, in that order. Pair the result with `trt_levels` by naming it, or pass it directly to `mirror_slope_plot`, which assigns the first color to the reference arm and the second to the comparator arm.

Usage

```
mirror_palette(name = c("default", "colorblind", "capricor"), reversed = FALSE)
```

Arguments

`name`	Character. One of: `"default"` Neutral gray reference, strong blue comparator. `"colorblind"` Okabe-Ito colorblind-safe pair (gray, blue). `"capricor"` House palette (gray reference, Capricor blue).
`reversed`	Logical. If `TRUE`, swap the two colors so the stronger color is applied to the reference arm. Default `FALSE`.

Value

An unnamed length-2 character vector of hex color codes: `c(reference_color, comparator_color)`.

Examples

```
mirror_palette("default")
mirror_palette("colorblind")
# Name it to bind explicitly to arms:
stats::setNames(mirror_palette("capricor"), c("Placebo", "Active"))
```

mirror_slope_plot — *Render a mirror slope plot*

Description

Produces a publication-quality mirror slope plot from a `mirror_data` object built with `prep_mirror_data` and `add_summary` (and optionally `add_precision` and `add_difference`).

Usage

```
mirror_slope_plot(
  mirror_data,
  show_spaghetti = FALSE,
  spaghetti_alpha = 0.15,
  spaghetti_size = 0.4,
  show_difference = FALSE,
  show_slowing = FALSE,
  show_precision = FALSE,
  show_n = TRUE,
  direction = NULL,
  outcome_label = NULL,
  benefit_label = NULL,
  x_center = 12,
  x_offset = NULL,
  x_min = NULL,
  x_max = NULL,
  y_lim = NULL,
  trt_colors = mirror_palette("default"),
  band_color = NULL,
  band_alpha = 0.12,
  line_size = 1,
  point_shape = 18,
  point_size = 3.2,
  base_size = 11,
  y_percent = TRUE,
  jitter_points = FALSE,
  title = NULL,
  subtitle = NULL,
  y_label = NULL,
  caption = NULL,
  footnote = NULL,
  seed = NULL,
  return_data = FALSE,
  save_path = NULL,
  width = 8,
  height = 6,
  dpi = 300
)
```

Arguments

`mirror_data` A `mirror_data` object with summary rows attached.

`show_spaghetti` Logical. Draw individual subject trajectories beneath the mean lines. Default `FALSE`.

`spaghetti_alpha, spaghetti_size` Numeric. Transparency and line width of the individual trajectory lines.

show_difference
: Logical. Show the difference value (and the difference band). Requires a difference attached via add_difference. Default FALSE.

show_slowing
: Logical. Show the relative benefit percentage (and the difference band). Default FALSE.

show_precision
: Logical. Draw per-arm precision bars at the endpoint markers, if precision is attached. Default FALSE.

show_n
: Logical. Show the analyzed N per arm in the strip header. Default TRUE.

direction
: Character or NULL. "decrease" or "increase"; if NULL, derived from the data (see section above).

outcome_label, benefit_label
: Character or NULL. Override the auto-generated terminology in the benefit line of the annotation zone.

x_center
: Numeric. The x position at which both arm lines converge. Default 12. A positional coordinate only; not a unit of time.

x_offset
: Numeric or NULL. Distance from x_center to each baseline; auto if NULL.

x_min, x_max
: Numeric or NULL. Plot boundaries; auto if NULL. x_max auto-expands to fit the annotation zone.

y_lim
: Numeric length-2 or NULL. Y-axis limits; auto if NULL.

trt_colors
: Length-2 character vector of arm colors c(reference, comparator), or a named vector matching the arms. Default mirror_palette("default").

band_color
: Character or NULL. Difference band fill; defaults to the comparator color.

band_alpha
: Numeric. Difference band transparency. Default 0.12.

line_size
: Numeric. Mean line width. Default 1.2.

point_shape
: Integer. Endpoint marker shape. Default 18.

point_size
: Numeric. Endpoint marker size. Default 4.

base_size
: Numeric. Base font size passed to theme_mirror. Default 11.

y_percent
: Logical. If TRUE (default), y-axis tick labels and the header statistics are formatted with a percent sign, appropriate for change-from-baseline outcomes. Set FALSE for absolute-scale outcomes.

jitter_points
: Logical. If TRUE, individual endpoints are given a small horizontal jitter (reference arm toward the left, comparator toward the right) so overlapping subjects are visible. Default FALSE. Use seed for reproducibility.

title, subtitle, caption
: Character or NULL. Standard labels.

y_label
: Character or NULL. Y-axis label; defaults to the value label stored in the object.

footnote
: Character or NULL. A figure footnote rendered in a smaller font below the caption.

seed
: Integer or NULL. Seed for reproducibility of any jittered elements. Default NULL. The plot is otherwise deterministic.

return_data
: Logical. If TRUE, return a list with $plot and $data. Default FALSE.

save_path	Character or NULL. If given, save via ggplot2::ggsave(). The format is taken from the file extension; use .pdf or .svg for a vector figure suitable for journal submission, or .png/.tiff at dpi 300 or higher for raster.
width, height	Numeric. Output size in inches when saving.
dpi	Numeric. Output resolution when saving. Default 300.

Details

Both arms share one x-axis and converge at a central endpoint position, so the between-arm gap at the endpoint is the primary visual quantity. All optional elements are off by default: the object stores everything available, and this function exposes only what is requested.

Value

A ggplot object, or a list of $plot and $data when return_data = TRUE. The plot can be extended with standard ggplot2 syntax.

Direction and labels

direction describes the direction of numeric movement of the value, derived by comparing the reference arm endpoint estimate to its baseline ("decrease" if the value moved down, "increase" if up). The default benefit label is phrased accordingly (e.g. "percent less decline"). For outcomes where the favourable direction is unusual (for example a score where a lower number is better), supply outcome_label and benefit_label explicitly rather than relying on the defaults.

See Also

prep_mirror_data, add_summary, add_difference, add_precision, theme_mirror, mirror_palette

Examples

```
df <- data.frame(
  USUBJID = rep(paste0("S", 1:20), each = 2),
  TRTP    = rep(c("Placebo", "Active"), each = 20),
  AVISIT  = rep(c("Baseline", "Month 12"), times = 20),
  AVAL    = c(rbind(rep(0, 20),
                    c(rnorm(10, -10, 3), rnorm(10, -6, 3))))
)
md <- prep_mirror_data(df, trt_levels = c("Placebo", "Active"),
                       time_baseline = "Baseline", time_endpoint = "Month 12")
md <- add_summary(md, c("Placebo" = -10.1, "Active" = -5.8),
                  stat_label = "LS-Mean")
md <- add_difference(md, estimate = 4.3, p_value = 0.01)
mirror_slope_plot(md, show_difference = TRUE, show_slowing = TRUE)
```

prep_mirror_data *Prepare analysis data for a mirror slope plot*

Description

First step of the mirror slope pipeline. Validates and structures a long analysis data frame into a `mirror_data` object containing subject-level (`OBS`) rows, ready to receive group-level summaries via `add_summary`.

Usage

```
prep_mirror_data(
  data,
  subject_var = "USUBJID",
  trt_var = "TRTP",
  trt_levels,
  time_var = "AVISIT",
  time_baseline,
  time_endpoint,
  value_var = "AVAL",
  pop_flags = NULL,
  value_label = "Change from Baseline",
  population_label = NULL
)
```

Arguments

`data`	A data frame in long format, one row per subject per visit.
`subject_var`	Character. Column name for the subject identifier. Default `"USUBJID"`.
`trt_var`	Character. Column name for the treatment arm. Default `"TRTP"`.
`trt_levels`	Character vector of length 2 in the form `c(reference, comparator)`. The first element is the reference (control) arm and plots on the left; the second is the comparator (treatment) arm and plots on the right. Values must be present in `trt_var`.
`time_var`	Character. Column name for the visit/time variable. Default `"AVISIT"`.
`time_baseline`	Character. The `time_var` value identifying the baseline visit.
`time_endpoint`	Character. The `time_var` value identifying the endpoint visit.
`value_var`	Character. Column name for the analysis value. Default `"AVAL"`.
`pop_flags`	Optional named list of population flag filters applied with AND logic, e.g. `list(SAFFL = "Y", ANL01FL = "Y")`. Each name is a column and each value the level to retain. Default `NULL` (no filtering).
`value_label`	Character. Human-readable description of `value_var`, used as the default y-axis label. Default `"Change from Baseline"`.
`population_label`	Character. Description of the analysis population for metadata and figure footnotes, e.g. `"ITT"`. Default `NULL`.

Details

The function filters to exactly two treatment arms, applies optional population flags, retains only the baseline and endpoint visits, assigns the `ARMROLE` and `TIMEROLE` role columns, counts the analyzed N per arm, and records provenance metadata. It does not compute any statistics.

Value

A `mirror_data` object containing `OBS` rows and populated metadata, including `schema_version` and provenance.

See Also

`add_summary`, `add_difference`, `add_precision`, `mirror_slope_plot`

Examples

```
df <- data.frame(
  USUBJID = rep(paste0("S", 1:6), each = 2),
  TRTP    = rep(c("Placebo", "Active"), each = 6),
  AVISIT  = rep(c("Baseline", "Week 24"), times = 6),
  AVAL    = c(0, -8, 0, -11, 0, -9, 0, -4, 0, -6, 0, -5),
  SAFFL   = "Y"
)
md <- prep_mirror_data(
  df,
  trt_levels    = c("Placebo", "Active"),
  time_baseline = "Baseline",
  time_endpoint = "Week 24",
  pop_flags     = list(SAFFL = "Y")
)
md
```

`print.mirror_data` *Print a mirror_data object*

Description

Shows a structured summary: arms and roles, timepoints, population, analyzed N per arm, the statistic label, and which optional elements (summary, precision, difference) are attached.

Usage

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

Arguments

`x`	A `mirror_data` object.
`...`	Unused.

Value

`x`, invisibly.

`summary.mirror_data` *Summarize a mirror_data object*

Description

Returns per-arm, per-timepoint N, mean, and range of the observed values.

Usage

```
## S3 method for class 'mirror_data'
summary(object, ...)
```

Arguments

`object`	A `mirror_data` object.
`...`	Unused.

Value

A data frame, invisibly, with one row per arm/timepoint.

`theme_mirror` *A publication theme for mirror slope plots*

Description

A `ggplot2` theme tuned for journal figures: a clean sans typeface, a coherent family of light hairline rules, no x-axis ticks (the x-axis is positional, not numeric), a clear type hierarchy, and proportionally scaled text. Because `mirror_slope_plot` returns a standard `ggplot` object, you can apply this theme and then override any element with `+ theme(...)`.

Usage

```
theme_mirror(base_size = 11, base_family = NULL)
```

Arguments

`base_size`	Numeric. Base font size in points; all text scales from it. Default `11`.
`base_family`	Character. Base font family. Defaults to a clean sans stack resolved from what the graphics device provides (`"Helvetica"`, then `"Arial"`, then `"sans"`). Pass a specific family to override.

Details

The theme hides x-axis ticks and the minor grid because the horizontal axis in a mirror slope plot encodes position (baseline versus endpoint for each arm), not a continuous numeric scale. A single set of very light horizontal guides is retained to aid reading values off the y-axis. Rule weights (axis lines, ticks, gridlines) are deliberately kept to a light, consistent family so that no chart furniture competes with the data.

For submission-quality output, render to a vector format (PDF or SVG) or to a raster at 300 dpi or higher, and ensure the chosen `base_family` is available to and embedded by the graphics device.

Value

A `ggplot2` theme object.

Examples

```
# Applied automatically inside mirror_slope_plot(), or layer it yourself:
# p + theme_mirror(base_size = 13)
```

Index

add_difference, 2, *4*, *5*, *7*, *8*, *10*
add_precision, 3, *5*, *8*, *10*
add_summary, 3, *3*, *5*, *8–10*
as.data.frame.mirror_data, 4

mirror_palette, 5, *8*
mirror_slope_plot, *3*, 5, *5*, *10*, *11*

prep_mirror_data, *4*, *5*, *8*, 9
print.mirror_data, 10

summary.mirror_data, 11

theme_mirror, *7*, *8*, 11