

Package ‘tsahr’

October 8, 2026

Type Package

Title Trial Sequential Analysis for Meta-Analyses of Hazard Ratios

Version 0.2.8.17

Description Performs Trial Sequential Analysis (TSA) for meta-analyses of time-to-event outcomes reported as hazard ratios. Implements the Schoenfeld required-events sample-size formula generalised for unequal allocation, the Diversity (D-squared) heterogeneity adjustment of Wetterslev et al. (2009), and O'Brien-Fleming-type alpha- and beta-spending trial sequential monitoring boundaries computed at the inverse-variance information based on observed study-level log-HR standard errors (not merely event counts), via a compiled (C++) recursive numerical integration engine ported from the R package 'RTSA' (Soerensen, Olsen, Lange and Gluud; an R implementation of the Trial Sequential Analysis software of the Copenhagen Trial Unit, <<https://ctu.dk/tools>>), with no fixed software-imposed limit on the number of looks (subject to available computational resources). Produces a cumulative Z-curve plot with efficacy, futility, and conventional significance boundaries. Methodology follows Miladinovic et al. (2013) <[doi:10.1016/j.jclinepi.2012.11.007](https://doi.org/10.1016/j.jclinepi.2012.11.007)> and Wetterslev et al. (2009) <[doi:10.1186/1471-2288-9-86](https://doi.org/10.1186/1471-2288-9-86)>.

License GPL (>= 2)

Copyright inst/COPYRIGHTS

Encoding UTF-8

Depends R (>= 4.0.0)

Imports metafor, readxl, ggplot2 (>= 3.4.0), stats, utils, Rcpp (>= 1.0.0)

LinkingTo Rcpp

Suggests testthat (>= 3.0.0), RTSA

Config/testthat/edition 3

URL <https://github.com/tarak-dhaouadi/tsahr>

BugReports <https://github.com/tarak-dhaouadi/tsahr/issues>

NeedsCompilation yes

Maintainer Tarak Dhaouadi <dhaouaditarak@yahoo.fr>

Config/roxygen2/version 8.1.0

Author Tarak Dhaouadi [aut, cre],
Anne Lyngholm Soerensen [ctb, cph] (author of the RTSA package, from
which the boundary engine is ported),
Markus Harboe Olsen [ctb, cph] (author of the RTSA package, from which
the boundary engine is ported),
Theis Lange [ctb, cph] (author of the RTSA package, from which the
boundary engine is ported),
Christian Gluud [ctb, cph] (author of the RTSA package, from which the
boundary engine is ported)

Repository CRAN

Date/Publication 2026-10-08 09:50:02 UTC

Contents

tsahr-package	2
plot.tsa_hr	3
print.tsa_hr	6
summary.tsa_hr	7
tsahr_example_data	7
tsa_hr	8
Index	18

tsahr-package	<i>tsahr: Trial Sequential Analysis for Meta-Analyses of Hazard Ratios</i>
---------------	--

Description

Performs Trial Sequential Analysis (TSA) for meta-analyses of time-to-event outcomes reported as hazard ratios. Implements the Schoenfeld required-events sample-size formula generalised for unequal allocation, the Diversity (D-squared) heterogeneity adjustment of Wetterslev et al. (2009), and O’Brien-Fleming-type alpha- and beta-spending trial sequential monitoring boundaries computed at the inverse-variance information based on observed study-level log-HR standard errors (not merely event counts), via a compiled (C++) recursive numerical integration engine ported from the R package ‘RTSA’ (Soerensen, Olsen, Lange and Gluud; an R implementation of the Trial Sequential Analysis software of the Copenhagen Trial Unit, <https://ctu.dk/tools>), with no fixed software-imposed limit on the number of looks (subject to available computational resources). Produces a cumulative Z-curve plot with efficacy, futility, and conventional significance boundaries. Methodology follows Miladinovic et al. (2013) [doi:10.1016/j.jclinepi.2012.11.007](https://doi.org/10.1016/j.jclinepi.2012.11.007) and Wetterslev et al. (2009) [doi:10.1186/14712288986](https://doi.org/10.1186/14712288986).

Author(s)

Maintainer: Tarak Dhaouadi <dhaouaditarak@yahoo.fr>

Authors:

- Tarak Dhaouadi <dhaouaditarak@yahoo.fr>

Other contributors:

- Anne Lyngholm Soerensen (author of the RTSA package, from which the boundary engine is ported) [contributor, copyright holder]
- Markus Harboe Olsen (author of the RTSA package, from which the boundary engine is ported) [contributor, copyright holder]
- Theis Lange (author of the RTSA package, from which the boundary engine is ported) [contributor, copyright holder]
- Christian Gluud (author of the RTSA package, from which the boundary engine is ported) [contributor, copyright holder]

See Also

Useful links:

- <https://github.com/tarak-dhaouadi/tsahr>
- Report bugs at <https://github.com/tarak-dhaouadi/tsahr/issues>

plot.tsa_hr

Plot a tsa_hr object

Description

Produces the standard Trial Sequential Analysis chart: cumulative Z-curve, O'Brien-Fleming-type alpha (efficacy) and beta (futility) spending boundaries, the conventional (naive) significance boundary, the theoretical Diversity-Adjusted Required Information Size (DARIS) event-equivalent reference line, and – whenever DARIS has actually been reached (see ?tsa_hr, "DARIS reached" criterion) – a second reference line marking the estimated cumulative-events point at which the observed accrued statistical information reached DARIS. The two lines are shown and labelled separately, since they are different quantities that need not coincide (see ?tsa_hr, Section 7b).

Usage

```
## S3 method for class 'tsa_hr'
plot(
  x,
  legend = TRUE,
  caption = TRUE,
  caption_size = 8,
  caption_face = "italic",
```

```

show_theoretical_daris = TRUE,
daris_label_size = 3.2,
daris_label_x = NULL,
daris_label_y = NULL,
info_threshold_label_size = 3.2,
info_threshold_label_x = NULL,
info_threshold_label_y = NULL,
events_label_size = 3.2,
events_label_x = NULL,
events_label_y = NULL,
endpoint_label_size = NULL,
endpoint_label_x = NULL,
endpoint_label_y = NULL,
show_historical_daris = TRUE,
historical_label_size = NULL,
historical_label_x = NULL,
historical_label_y = NULL,
xmax_mult = 1.15,
alpha_col = "firebrick",
beta_col = "blue",
naive_col = "darkgreen",
z_col = "black",
...
)

```

Arguments

x	An object of class "tsa_hr".
legend	Logical; show the boundary-type legend at the bottom of the plot. Default TRUE.
caption	Logical; show the methods caption below the plot. Default TRUE. When tsa_hr() was run with a non-standard re_inference ("hksj"/"knha" or "Hksj_adhoc"), the caption gains a line right under the first "Methods" line naming that inference option; it is absent for the default "standard". When the route's own endpoint (DARIS for boundary_route = "design"; the analysis-route endpoint for "analysis") has not yet been reached, the caption gains final lines with the projection: "Theoretical additional events to DARIS (Schoenfeld)", "Estimated additional events to DARIS (historical rate)" and "Estimated additional studies required" (for the analysis route, "DARIS" reads "analysis-route endpoint"). A further line states how many studies were excluded from the events projection (zero events), if any – see ?tsa_hr, "Estimated additional studies/events".
caption_size	Font size for the methods caption text. Default 8.
caption_face	Font face for the methods caption text: one of "italic" (default, matching the previous fixed styling) or "plain". Also accepts any other value ggplot2::element_text() understands for face (e.g. "bold", "bold.italic").
show_theoretical_daris	Logical; show the theoretical DARIS event-equivalent reference line and its label (see Details). Default TRUE. Set to FALSE to hide it – e.g. when it would

clutter the plot, or when only the observed-information "DARIS information reached" marker is of interest. Has no effect on the underlying DARIS calculation or on the "DARIS reached" verdict, only on what is drawn.

`daris_label_size`

Font size for the theoretical "DARIS event-equivalent" label. Default 3.2.

`daris_label_x, daris_label_y`

Position (in data coordinates: x = cumulative events, y = Z-score) for the theoretical DARIS event-equivalent label. Default NULL uses the built-in position (just right of its vertical line, near the top of the plot).

`info_threshold_label_size`

Font size for the "DARIS information reached" label (only shown when DARIS has actually been reached). Default 3.2.

`info_threshold_label_x, info_threshold_label_y`

Position (in data coordinates) for the "DARIS information reached" label. Default NULL uses the built-in position.

`events_label_size`

Font size for the "Events accrued" label. Default 3.2.

`events_label_x, events_label_y`

Position (in data coordinates) for the "Events accrued" label. Default NULL uses the built-in position (bottom right, above the last data point).

`endpoint_label_size`

Font size for the "Analysis-route endpoint (Design_R x DARIS) reached" label (only shown when `boundary_route` = "analysis"; from 0.2.8.3 it is also drawn, worded "not yet reached; theoretical ~ N events", at the theoretical position when the endpoint has not been reached). Default NULL uses the same size as `info_threshold_label_size` (3.2 unless changed), which is what this label followed before 0.2.8.

`endpoint_label_x, endpoint_label_y`

Position (in data coordinates: x = cumulative events, y = Z-score) for the "Analysis-route endpoint (Design_R x DARIS) reached" label. Default NULL uses the built-in position (just right of its vertical line, below the "DARIS information reached" label).

`show_historical_daris`

Logical; when the route's own target has not been reached, draw an extra reference line at the target position projected from the historical information-per-event rate (accrued events plus the estimated additional events, `projection$target_events_historical`). For `boundary_route` = "design" the target is DARIS and the line ("DARIS (historical rate)") sits next to the theoretical (Schoenfeld) DARIS line; for "analysis" the target is the analysis-route endpoint (`design_R` x DARIS) and the line ("Historical information/event-rate projection") sits next to the theoretical endpoint line; it is a projection of where that same endpoint would be reached, not a second definition of the endpoint. Default TRUE. Only affects what is drawn.

`historical_label_size`

Font size for the historical-rate label. Default NULL uses `daris_label_size`.

`historical_label_x, historical_label_y`

Position (in data coordinates) for the historical-rate label. Default NULL uses the built-in position (just right of its vertical line; design route below the theoretical DARIS label, analysis route below the analysis-route endpoint label).

xmax_mult	Positive number; multiplier applied to the largest x value that must fit in the plot (accrued events, theoretical DARIS, DARIS information marker, historical-rate projection, analysis-route endpoint, and the last x of the formal boundaries) to obtain the upper limit of the x-axis. Default 1.15, i.e. 15% of free space to the right. Use a larger value (e.g. 1.4) to leave more room for labels, or 1 to end the axis exactly at the largest element. Values below 1 crop the right-hand part of the plot.
alpha_col	Color for the alpha (efficacy) boundary line. Default "firebrick".
beta_col	Color for the beta (futility) boundary line. Default "blue".
naive_col	Color for the naive/conventional significance boundary line. Default "darkgreen".
z_col	Color for the cumulative Z-score line/points. Default "black".
...	Currently unused.

Details

The subtitle has two lines: the model and design summary (random-effects model, Diversity D^2, anticipated HR, allocation psi, alpha and power), and, from 0.2.8, the pooled random-effects HR with its 95% CI, the p-value of the pooled effect, tau^2 and I^2 (the same values print() and summary() report).

Value

A ggplot object (invisibly), also drawn on the current graphics device / returned for further customisation, e.g. ggplot2::ggsave().

print.tsa_hr	<i>Print a tsa_hr object</i>
--------------	------------------------------

Description

Print a tsa_hr object

Usage

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

Arguments

x	An object of class "tsa_hr".
...	Currently unused.

Value

An object of class "tsa_hr", namely the same object x supplied to the method, returned invisibly. The method prints a concise summary of the Trial Sequential Analysis results and returns the original object unchanged.

summary.tsa_hr	<i>Summarise a tsa_hr object</i>
----------------	----------------------------------

Description

Prints object\$summary_table (Parameter/Value); since 0.2.8.12 Parameter uses short abbreviations (e.g. "DARIS", "AR endpoint", "RE") to keep the table readable, and an "Abbreviations:" line spelling them out is printed directly below the table (from attr(object\$summary_table, "abbreviations")).

Usage

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

Arguments

object	An object of class "tsa_hr".
...	Currently unused.

Value

The underlying summary data.frame (invisibly printed).

tsahr_example_data	<i>Path to a bundled example hazard-ratio meta-analysis dataset</i>
--------------------	---

Description

Returns the file path to one of the two example datasets bundled with the package, suitable for trying out [tsa_hr](#). Both are .xlsx sheets with one row per study and the columns Study, Ethnicity, Age, HR, lower, upper, log_HR, Std_Error, Events_Treatment, N_treatment, Events_controls and N_controls (the last seven are the ones tsa_hr() uses).

Usage

```
tsahr_example_data(dataset = c("HR_meta", "HR_meta_2"))
```

Arguments

dataset	Which example to return: "HR_meta" (default; 20 studies) or "HR_meta_2" (40 studies).
---------	---

Details

Before 0.2.8 the package bundled a single 10-study example (`HR_meta_example.xlsx`); it was replaced by the two datasets above. The package's own test suite keeps a frozen copy of the old dataset (its numerical results are pinned by tests), so this change affects only what `tsahr_example_data()` returns.

Value

A character string giving the path to the example .xlsx file.

Examples

```
path <- tsahr_example_data()           # 20 studies
d <- readxl::read_excel(path)
head(d)

path2 <- tsahr_example_data("HR_meta_2") # 40 studies
nrow(readxl::read_excel(path2))
```

tsa_hr

Trial Sequential Analysis for a meta-analysis of Hazard Ratios

Description

Performs a Trial Sequential Analysis (TSA) for a meta-analysis of time-to-event outcomes reported as hazard ratios (HR). Adapts the classical Wetterslev/Thorlund/CTU TSA framework to time-to-event data using the Schoenfeld required-events formula (generalised for unequal allocation), the Diversity (D-squared) heterogeneity adjustment of Wetterslev et al. (2009), and O'Brien-Fleming-type alpha- and beta-spending monitoring boundaries computed at the observed accrued statistical information.

Usage

```
tsa_hr(
  data,
  alpha_two_sided = 0.05,
  power = 0.8,
  allocation_source = c("data", "manual"),
  allocation_p = 0.5,
  target_HR = NA_real_,
  method = "DL",
  order_by = NULL,
  verbose = TRUE,
  boundary_route = c("design", "analysis"),
  legacy_fallback = TRUE,
  projection_stat = c("median", "mean"),
  info_per_event_basis = c("per_study", "pooled"),
  re_inference = "standard"
)
```

Arguments

data	A data.frame, or a path to an .xlsx file, containing one row per study with (at least) the columns: Study, log_HR, Std_Error, Events_Treatment, N_treatment, Events_controls, N_controls. Rows are treated as being in chronological (publication) order; reorder your data before calling this function if the row order in your file is not chronological.
alpha_two_sided	Overall two-sided type I error for the TSA monitoring boundaries. Default 0.05.
power	Desired power (1 - beta) for the required information size calculation. Default 0.80.
allocation_source	Either "data" (default; the allocation ratio psi is computed automatically as $\text{sum}(N_{\text{treatment}}) / \text{sum}(N_{\text{treatment}} + N_{\text{controls}})$ across the included studies) or "manual" (psi is fixed to allocation_p, e.g. for planning a future trial with a pre-specified ratio).
allocation_p	Allocation proportion to use when allocation_source = "manual". Must be strictly between 0 and 1. Ignored when allocation_source = "data".
target_HR	Anticipated hazard ratio used for the required information size calculation. Default NA, which uses the *observed* pooled HR from the random-effects meta-analysis – see Details for an important caution about this default. Set to a pre-specified clinically-anticipated value (e.g. 0.80) for a standard, non-circular, protocol-driven TSA.
method	Character string specifying the heterogeneity-variance (τ^2) estimator used for the <i>random-effects</i> meta-analysis and cumulative (sequential) TSA model, passed to method in metafor::rma() after validation and, for the aliases "CO"/"VC", normalisation to "HE". One of "DL" (DerSimonian-Laird, the default – kept as the default here for backward compatibility with earlier tsahr versions, which always used DL; note this differs from metafor::rma()'s own default of "REML"), "HE", "HS", "HSk", "SJ", "ML", "REML", "EB", "PM", or "PMM". "CO" and "VC" are also accepted and normalised to "HE" (see below). See ?metafor::rma for the definition of each estimator. This only changes the <i>random-effects</i> model (res_re); the equal-effects model used internally for the Diversity (D^2) heterogeneity adjustment is always fitted with method = "FE" and is unaffected by this argument. Alpha/beta-spending boundary calculations are unaffected by this argument. "CO" and "VC" are accepted as aliases for "HE" and are normalised to "HE" before being passed to metafor::rma(). metafor's documentation notes that the Hedges estimator is also known as the Cochran ("CO") or variance-component ("VC") estimator, and that those strings may be used to select it – but that alias is not accepted by every metafor version (older releases reject a bare "CO" with "Unknown 'method' specified"). Normalising here makes tsa_hr() behave identically across metafor versions rather than inheriting that version skew, and avoids pinning a minimum metafor version purely for an alias. All three strings denote the same estimator, so this has no numerical consequence. The returned object records both parameters\$method (the normalised string actually used, i.e. "HE") and parameters\$method_requested (what the caller passed).

Two method strings accepted by `metafor::rma()` are deliberately NOT supported here: "GENQ" and "GENQM" require the caller to also supply a weights argument to `metafor::rma()`, which `tsa_hr()` does not currently collect or pass through, so passing them here raises an explicit error explaining why rather than silently forwarding to `metafor::rma()` and surfacing its own unrelated error.

<code>order_by</code>	Optional name of a column in data to sort by (ascending) before the cumulative analysis, e.g. a publication-year column. TSA is order-dependent, so getting the chronological order right matters. Default NULL, which uses the row order already present in data and assumes it is chronological (with no way for the package to verify this). Column names in data have spaces replaced with underscores on load (so an Excel header "Std Error" becomes <code>Std_Error</code>); <code>order_by</code> is normalised the same way, so either "Publication Year" or "Publication_Year" will match that column. If two distinct headers would collide once spaces become underscores, <code>tsa_hr()</code> stops rather than silently using whichever column came first.
<code>verbose</code>	Logical; print analysis details to the console as the function runs (mirrors the diagnostic output of the original script). Default TRUE.
<code>boundary_route</code>	Character string, one of "design" (default) or "analysis", selecting which of RTSA's two retrospective boundary-computation routes to use. See "Retrospective boundary timeline" under Details for what each computes and, for "analysis", how it changes the formal endpoint, the "DARIS reached" verdict, and every decision field in <code>results</code> – this is more than swapping out the futility numbers.
<code>legacy_fallback</code>	Logical, default TRUE. Governs what happens if the compiled RTSA-derived boundary engine fails to produce a result for the requested design (e.g. no root bracket exists for an unusual information-fraction schedule). When TRUE (the default), <code>tsa_hr()</code> falls back to the legacy, pre-0.2.7.11 R-only approximate engine, with an immediate warning and a visible banner in <code>print()/summary()</code> output, and marks the result (<code>beta_engine\$engine == "legacy_r_fallback"</code>) so the fallback is never silent – but it IS a fallback: a caller who wraps the call in <code>suppressWarnings()</code> will not see it, and the returned boundaries are not RTSA-comparable when this happens. Set <code>legacy_fallback = FALSE</code> for strict fail-closed behaviour: an engine failure then stops <code>tsa_hr()</code> with an error instead of silently substituting the approximate engine, appropriate when the result will be reported as RTSA-equivalent and an unnoticed fallback would be worse than a hard stop.
<code>projection_stat</code>	Character string, one of "median" (default) or "mean", selecting the summary statistic used to turn the OBSERVED per-study contributions into "typical" values for the retrospective projection described under "Estimated additional studies/events" in Details: the information per study ($1/\text{Std_Error}^2$, giving the additional-studies estimate) and the information per event ($1/\text{Std_Error}^2$ divided by the study's total events, giving the additional-events estimate). "median" is the default because it is more robust to a single unusually large or small study; "mean" is offered as an alternative when that robustness is not wanted (e.g. a

deliberately evenly-sized set of studies). This has no effect on any other quantity returned by `tsa_hr()` – it only feeds the projection element of the return value and the corresponding printed/plotted text.

info_per_event_basis

Character string, one of "per_study" (default) or "pooled", selecting how the "historical rate" of information per event used for the additional-EVENTS projection is obtained. "per_study": the projection_stat (median by default, or mean) of each study's own information per event ($1/\text{Std_Error}^2$ divided by its total events). "pooled": the ratio of sums (total information / total events), which weights larger studies more and equals the slope of observed cumulative information against cumulative events. Studies with zero events (or another non-finite or non-positive ratio) are technically accepted but are excluded from either variant, and the number excluded is stated explicitly in the printed output. It does not affect the additional-STUDIES estimate, which always uses projection_stat. Both variants are always returned in projection as a sensitivity check.

re_inference

Character string selecting how inference on the pooled *random-effects* effect, and on every cumulative look, is carried out; matching is case-insensitive. The heterogeneity-variance estimator (method) and the pooled point estimate are the same under every option – only the standard error, test statistic, p-value and 95% CI change. One of:

"standard" (default; exactly the behaviour of earlier tsahr versions) the usual Wald-type inference: standard normal reference distribution, variance $1 / \sum(w)$ with random-effects weights $w = 1 / (\text{Std_Error}^2 + \tau^2)$.

"hksj" (**alias** "knha") Hartung-Knapp-Sidik-Jonkman (Knapp-Hartung) adjustment: the variance is multiplied by $q = \sum(w * (\log_HR - \text{pooled})^2) / (k - 1)$ and the test and CI use a t distribution with $k - 1$ degrees of freedom (metafor::`rma(test = "knha")`). q may be below 1, in which case the adjusted CI can be *narrower* than the standard one.

"Hksj_adhoc" (**alias** "knha_adhoc") HKSJ with the ad hoc correction that q is never allowed to be below 1: the variance is multiplied by $\max(1, q)$ (t distribution, $k - 1$ df), so the standard error is never smaller than under "standard". Also known as the modified/truncated HKSJ method.

See "Random-effects inference" under Details for how the option enters the cumulative Z-curve and what it does *not* change. When a non-standard option is used, the plot caption states it.

Details

****Circularity caution:**** using the observed pooled effect (`target_HR = NA`) to determine the required information size is circular – it tends to make the required information size small whenever the pooled effect is large and precise, which can make the TSA boundary collapse to the conventional boundary almost immediately. For a publication-quality TSA, set `target_HR` to a value fixed independently of (and ideally before looking at) the meta-analysis result.

****Random-effects caveat:**** the cumulative Z-curve is estimated from a random-effects model whose between-study variance is re-estimated at every step. The canonical Lan-DeMets/O'Brien-Fleming theory assumes a fixed, canonical information process with independent Brownian-motion

increments. Because tsahr obtains each cumulative Z statistic from a random-effects meta-analysis with tau-squared re-estimated at each look, the resulting Z process does not exactly satisfy that canonical model; the displayed monitoring boundaries should therefore be regarded as an approximation (as in the official Copenhagen Trial Unit TSA software), not an exact result.

Random-effects inference (`re_inference`, **0.2.8.11**). `re_inference` changes how the *random-effects* pooled effect and each cumulative look are tested, not how heterogeneity is estimated: method still chooses the τ^2 estimator, and the pooled point estimate (log-HR) and τ^2 at every look are identical under all three options. For "hksj" and "hksj_adhoc" the standard error, p-value and 95% CI of the pooled HR (`res_re`, printed output, summary table and plot subtitle) and of every row of cumulative are the HKSJ ones, computed look by look from the studies accrued so far.

- **Z-curve scale.** The monitoring boundaries and the conventional boundary are defined on the standard normal scale, whereas an HKSJ statistic follows a t distribution with $k - 1$ df at look k . The cumulative Z is therefore the *normal-equivalent* of the HKSJ t statistic: the value with the same two-sided p-value on the normal scale ($\text{sign}(\text{estimate}) * \text{qnorm}(1 - p / 2)$). Consequently $|Z| \geq \text{qnorm}(1 - \alpha/2)$ exactly when the HKSJ p-value is below α , and the conventional (naive) boundary keeps its usual meaning. The raw t statistic is kept in `cumulative$zval`, the degrees of freedom in `cumulative$re_df` and the variance multiplier in `cumulative$re_scale` (these two columns exist only for non-standard options). The t distribution is not what the alpha spending boundaries were derived for, so this is a further approximation on top of the one described under "Random-effects caveat".
- **Early looks.** HKSJ needs at least two studies: the HKSJ statistic is undefined at $k = 1$ (a t distribution on 0 df has no defined quantile), so `cumulative$Z` is NA at the first look for "hksj"/"hksj_adhoc" – shown as NA in the printed cumulative tables and simply not drawn on the plotted Z-curve (`cumulative$se`, `$pval` etc. still hold the standard, z-based values at that look; only Z is withheld). A look at which the scale factor is zero or not finite (e.g. all estimates identical) likewise keeps the standard z-based Z; both cases have `re_df = Inf` and `re_scale = 1`. At the second look $k - 1 = 1$ degree of freedom, so early HKSJ looks (however defined) are very conservative and erratic; a `warning()` says so (with `call. = FALSE`, the same way as the `target_HR-near-1` caveat below) whenever a non-standard `re_inference` is actually used, regardless of `verbose`. Early cumulative HKSJ values should not be interpreted as directly comparable in magnitude with conventional normal Z-statistics.
- **What is not changed.** The Diversity D^2 , the adjustment factor, DARIS, the information fractions and hence the alpha/beta boundaries are always computed from the standard random-effects and equal-effects variances and do not depend on `re_inference`, as do I^2 , τ^2 and Q. Only the Z-curve (and the decisions that compare it with the boundaries) and the reported pooled inference change. With `target_HR = NA` the anticipated HR is the pooled point estimate, which is also unaffected.
- **Degenerate data.** If the HKSJ scale factor of the full data set is zero or not finite, `tsa_hr()` warns and falls back to "standard"; `parameters$re_inference` then reads "standard" while `parameters$re_inference_requested` keeps what was asked for.

The plot caption names the option whenever it is not "standard"; `print()` and `summary()` do the same.

****Retrospective boundary timeline:**** the observed cumulative Z-curve continues through every included study, but the formal alpha and beta boundaries follow the RTSA retrospective convention: observed looks are retained only while `info_fraction < 1`, followed by one synthetic final-

analysis point at $t = 1$ (HARIS/DARIS). Boundaries are not continued through studies occurring after DARIS. The synthetic point is stored in `boundary_timeline`; the observed-study rows in cumulative have boundary values set to NA at and after DARIS.

Formal crossing/futility decisions are evaluated only through the first observed look reaching the route endpoint (DARIS for `boundary_route = "design"`; `design_R * DARIS` for "analysis"), using the definitive boundary at that endpoint.

Decision fields: "at any formal look" versus "at the definitive look" (0.2.7.14). results reports two families of fields that answer different questions:

At ANY formal look `crossed_tsa` and `entered_futility_region` are TRUE if the cumulative Z-curve crossed the efficacy boundary (respectively lay inside the futility region) at any look up to and including the definitive one. A trial that crossed efficacy at an interim look keeps `crossed_tsa = TRUE` even if the definitive look then fell back below the boundary.

At the DEFINITIVE look `final_crossed_efficiency`, `final_non_efficiency` ($= !\text{final_crossed_efficiency}$) and `final_entered_futility_region` refer ONLY to the first look reaching the route endpoint (`results$final_tsa_look`) and are NA when that endpoint has not been reached. Version 0.2.7.13 defined `final_non_efficiency` as `!crossed_tsa`, which means "never crossed at any look" rather than "the definitive look did not cross"; this was corrected in 0.2.7.14.

At the definitive look the futility boundary equals the final efficacy boundary (as in RTSA's design pass, where the two are calibrated to meet there; e.g. about 2.1-2.2 rather than 1.959964 for a two-sided alpha of 0.05 with many looks), so `final_entered_futility_region` is the complement of `final_crossed_efficiency` (both are TRUE only at $|Z|$ exactly equal to the boundary). It merely says that the Z-curve did not reach the final efficacy boundary; it is not a formal interim futility stop, and neither family of fields is a recommendation to stop a trial. (Versions 0.2.6.x-0.2.7.11 used $\min(qnorm(1 - \alpha/2), <\text{final efficacy boundary}>)$ as the final futility boundary.)

Route endpoint versus DARIS (0.2.7.13/0.2.7.14). `boundary_route` picks between direct ports of RTSA's two retrospective boundary computations:

"design" (default) `RTSA::boundaries(type = "design")`: alpha and beta boundaries are computed directly on the observed information-fraction timeline (looks below $t = 1$ plus one synthetic $t = 1$ point), with NO further inflation. The formal endpoint is DARIS itself (`info_fraction = 1`), matching this package's stated RIS/DARIS definition and the values validated against RTSA's design pass.

"analysis" `RTSA::RTSA(type = "analysis", design = NULL)`: a design pass first solves an inflation factor (`design_R`) so that the design's own futility and efficacy boundaries meet at its final look; the boundaries returned are then recomputed on the timeline scaled by `design_R`, and the efficacy boundaries change as well as the futility ones (alpha is rescaled on $t / \text{design_R}$). The formal endpoint becomes `design_R * DARIS`, not DARIS. This is RTSA's own inflated-sequential-design convention, not the no-inflation convention of the Copenhagen Trial Unit's TSA software.

DARIS and the route endpoint are reported separately and never conflated: `results$daris_reached` and `information_size$DARIS_info_threshold_events` always refer to DARIS itself, whereas `results$final_reached`, `information_size$route_endpoint_info` and `information_size$route_endpoint_events` refer to the route endpoint (identical to DARIS for "design"). For "analysis" the printed output, summary table and plot call the endpoint "analysis-route endpoint (x.xxx x DARIS)" and never "DARIS". Both routes share the same compiled recursion (`src/rtsa_core.h`); only the orchestration differs.

Fallbacks (0.2.7.14). `settings$route_used("design", "analysis" or "legacy")`, `settings$fallback_used`, `settings$fallback_route("none", "design" or "legacy")` and `settings$fallback_reason` record programmatically what actually produced the boundaries. "design": `boundary_route = "analysis"` failed and the RTSA-derived design-route result is returned; "legacy": the RTSA-derived engine failed and the legacy, approximate R-only engine was used. Both are announced by warnings and, for "legacy", a banner in `print()/summary()`. For confirmatory or RTSA-parity work use `legacy_fallback = FALSE`, which makes either failure an error.

Numerical diagnostics. The compiled engine warns when a boundary search converged only within a loose tolerance, when an integration grid collapsed to a degenerate interval, and – with a separate, more alarming warning – when an integration interval was REVERSED (lower wall above the upper wall), a state RTSA's own code would have stopped on and which invalidates the boundaries from that look onward. Diagnostics are reported for the converged passes, not for the transient candidate information scales tried inside the root searches.

Placeholder boundaries: RTSA's tolerance, kept (0.2.7.22). At looks where the cumulative alpha spend is below RTSA's absolute search tolerance of $1e-9$ (very early looks, small information fractions) RTSA does not solve for a boundary but reports the placeholder value 20, and `tsahr` reproduces this deliberately (`TSA_boundary_upper == 20`). It is not a computed boundary: the true boundary there is finite (about 6.7 and 6.2 at information fractions 0.108 and 0.125 in a 37-look example), and a Z-curve above the placeholder is not counted as crossing. The same tolerance also makes the first boundary that IS solved after such looks slightly inaccurate (an error of about $7e-3$ in that example, shrinking at later looks). Both effects are RTSA's own; `tsahr` keeps RTSA's tolerance so that its bounds match RTSA's, and the placeholder stretches the vertical axis of `plot()`.

Final-look-only beta spend (0.2.7.22). When every interim look is suppressed (small information fractions) the final look carries the whole beta spend. RTSA has an exact-float shortcut for that case that fires or not depending on the last bit of beta; `tsahr` computes $\beta = 1 - \text{power}$ and used to hit it at power 0.80, losing the compiled engine to the legacy fallback. The shortcut is no longer ported, so the result no longer depends on the last bit of beta.

Estimated additional studies/events (retrospective projection). Whenever the route's target has NOT been reached in the observed data (`results$daris_reached == FALSE` for `boundary_route = "design"`; `results$final_reached == FALSE` for "analysis"), `tsa_hr()` additionally projects how many more studies, and roughly how many more events, would be needed to reach it – based directly on the OBSERVED study-level information increments already in the data, not on any new assumption about future trial size. This is deliberately a retrospective, data-driven HR-specific counterpart to RTSA's own prospective `minTrial()/ris(..., type = "retrospective")` machinery, not a re-implementation of it.

The target information I_{required} differs by route – computed separately, never conflated:

- "design": $I_{\text{required}} = \text{DARIS}$ (the information target is DARIS itself, i.e. `information_size$DARIS_info`).
- "analysis": $I_{\text{required}} = \text{design_R} \times \text{DARIS}$ (the analysis-route endpoint, i.e. `information_size$route_endpoint`).

The shortfall $I_{\text{required}} - \text{info_accrued_final}$ is translated in two separate ways, which answer different questions:

Additional events (primary) The shortfall divided by a "historical rate" of information per event, chosen with `info_per_event_basis`: "per_study" (default) uses the `projection_stat` (median by default, or mean) of each included study's own $(1/\text{Std_Error}^2) / \text{total_events}$; "pooled" uses the ratio of sums, total information / total events, which weights the larger studies more and equals the slope of observed cumulative information against cumulative events.

The result is continuous and is rounded up only for display. It is returned in `projection$additional_events_estimated`; both variants are always returned (`additional_events_study_level` and `additional_events_pooled`) so the choice can be checked as a sensitivity analysis.

Additional studies (secondary) The shortfall divided by a single "typical future study" information increment – the `projection_stat` of each study's own $1/\text{Std_Error}^2$ – and rounded up to the nearest whole study, so it is always a natural number. It is not affected by `info_per_event_basis`.

The two are deliberately not chained: the events figure is NOT the number of studies times an events-per-study increment, because rounding the studies up would inflate it. That chained quantity (whole typical studies x typical events per study) is still returned, as `projection$events_from_whole_studies`, but it answers "how many events would the minimum whole number of typical studies bring?", not "how many events are needed?".

For both routes three separate quantities are reported when the route's own target has not been reached (printed output, `summary()` and the plot caption); for `boundary_route = "design"` the target is DARIS, for `"analysis"` it is the analysis-route endpoint (`design_R x DARIS`):

Theoretical additional events (Schoenfeld) The direct, deterministic difference between the theoretical event-equivalent of the target (`DARIS_events` for design, `DARIS_events x design_R` for analysis) and `events_accrued` (both already pooled-psi event-equivalents), floored at 0 (`projection$additional_events_theoretical`; for the design route this equals `projection$additional_events`).

Estimated additional events (historical rate) The events projection above, with `I_required` set to the route's target information (`projection$additional_events_estimated`). `plot()` draws it as a second reference line, at accrued events plus these additional events (`projection$target_events_historical`) next to the theoretical line; see `show_historical_daris` in `plot.tsa_hr`. The printed output and `summary()` also give that position in cumulative events, just before the additional-events figure ("DARIS (historical rate)" for the design route; "Historical information/event-rate projection" for the analysis route).

Estimated additional studies required The studies projection above.

These two events figures rest on different assumptions and can disagree – for example, when the studies supply less information per event than the $\psi^*(1-\psi)$ assumption behind the Schoenfeld figure, the historical-rate figure is larger, and it can be positive when the Schoenfeld figure is already 0. Both are approximations.

Zero-event studies. Studies with zero events (or another non-finite or non-positive information-per-event ratio) are technically accepted by `tsa_hr()` but carry information with no events to attach it to, so they are excluded from the information-per-event summaries (both variants) and hence from the events projection. They remain in the additional-studies projection and in every other result. The number excluded is stated explicitly in the printed output, in `summary()`, and in the plot caption, and is returned as `projection$n_zero_event_studies` and `projection$n_excluded_events_projection`.

Caveat: fixed-effect information scale. All projections are linear extrapolations on the fixed-effect, study-level inverse-variance information scale ($1/\text{Std_Error}^2$) that `tsa_hr()` compares with DARIS. They deliberately do not model how random-effects weights or the between-study variance (τ^2) would change as further studies are added, nor any change in the size, allocation or baseline risk of future studies; the mathematics are not adjusted for random effects. Read every projected number of events or studies as indicative, not as a required quantity.

This projection is deliberately NOT called "number of studies required" anywhere in `tsa_hr()`'s output: that phrasing reads as deterministic, and it is not one. Figures are labelled "Estimated ...",

and the printed output always carries a note that the projection assumes future studies contribute information at approximately the observed historical rate and is not a formal guarantee. The full detail is returned in projection (see "Value"); when the endpoint HAS already been reached, the projected fields (`n_additional_studies`, `additional_events_estimated` and their variants) are NA (there is nothing left to project), and nothing is printed or plotted for it.

Value

An object of class "tsa_hr": a list containing the fitted random-effects and fixed-effect metafor::rma model objects, heterogeneity statistics (including D2, capped at 99.9 numerical stability in extreme-heterogeneity cases, alongside the uncapped D2_raw and a D2_was_capped logical flag so capping is auditable rather than silent), allocation and required-information-size details (`information_size`, which includes two distinct circularity flags: `circularity_warning` is TRUE whenever `target_HR` was left unspecified, since deriving the required information size from the observed pooled effect is circular regardless of how much information accrued; `circularity_severe` additionally requires accrued events to exceed three times the resulting DARIS, the runaway case in which the boundary collapses to the conventional one almost immediately), the cumulative analysis data frame (`cumulative`), the formal sequential boundary schedule including the synthetic $t=1$ final-analysis point (`boundary_timeline`; see "Retrospective boundary timeline" above), boundary-crossing results (`results`, including `crossed_conventional` – deliberately evaluated over the FULL cumulative Z-curve, including studies added after DARIS, unlike `crossed_tsa`, which is restricted to the formal decision horizon – and `entered_futility_region`, see the caveat under "Retrospective boundary timeline" above; a TRUE value at the DARIS-reaching look reflects a comparison against the definitive $t=1$ futility boundary, not an interim one, and is not itself a formal stopping recommendation; the definitive-look fields `final_crossed_efficacy`, `final_non_efficacy` and `final_entered_futility_region`, plus `daris_reached` and `final_reached` – see "Decision fields" under Details), a settings list recording `boundary_route`, `route_used`, `legacy_fallback`, `fallback_used`, `fallback_route`, `fallback_reason`, the resolved `route_endpoint` (1 for "design", `design_R` for "analysis"), `route_endpoint_info` and `used_legacy_engine`, a projection list (see "Estimated additional studies/events" above) with `method` (`projection_stat` used), `I_required`, `info_accrued`, `info_per_event_basis`, `events_accrued`, `additional_info_required`, `central_info_increment`, `central_event_increment`, `central_info_per_event` (the historical rate used), `study_level_info_per_event`, `pooled_info_per_event`, `n_additional_studies`, `additional_events_estimated` (continuous, information / info per event, on the chosen basis), `additional_events_study_level`, `additional_events_pooled`, `events_from_whole_studies`, `target_events_historical_rate` (accrued plus estimated additional events), `n_studies`, `n_zero_event_studies`, `n_excluded_events_projection` and, design-route only, `additional_events_required_design` (design-route only, equal to `additional_events_theoretical` there) and `additional_events_theoretical` – the projected quantities are NA once the route's endpoint has been reached, and a summary data frame (character Parameter and Value columns; Value is character so logical rows print as TRUE/FALSE/NA rather than 1/0/NA). Use `plot()`, `summary()`, or `print()` on the result.

`parameters` records `re_inference` (the normalised option actually used: "standard", "hksj" or "hksj_adhoc") and `re_inference_requested` (what the caller passed). With a non-standard option, `res_re` is the corresponding metafor::rma fit, cumulative gains the columns `re_scale` and `re_df`, and the summary table gains a "Random-effects inference" row; `cumulative$Z` is NA at the first look (see "Random-effects inference" under Details).

The summary table's Parameter column (0.2.8.12) uses short abbreviations (e.g. "DARIS", "AR endpoint", "RE") to stay readable; the expansions are returned as a named character vector in

`attr(summary_table, "abbreviations")` and are printed by `summary()` underneath the table.

References

Miladinovic B, Mhaskar R, Hozo I, Kumar A, Mahony H, Djulbegovic B. "Optimal information size in trial sequential analysis of time-to-event outcomes reveals potentially inconclusive results because of the risk of random error." *J Clin Epidemiol*. 2013;66(6):654-9.

Wetterslev J, Thorlund K, Brok J, Gluud C. "Estimating required information size by quantifying diversity in random-effects model meta-analyses." *BMC Med Res Methodol*. 2009;9:86.

Examples

```
path <- tsahr_example_data()
res <- tsa_hr(path, target_HR = 0.80)
summary(res)
plot(res)
```

Index

`plot.tsa_hr`, [3](#), [15](#)

`print.tsa_hr`, [6](#)

`summary.tsa_hr`, [7](#)

`tsa_hr`, [7](#), [8](#)

`tsahr` (`tsahr-package`), [2](#)

`tsahr-package`, [2](#)

`tsahr_example_data`, [7](#)