

Description
Remarks and examples

Quick start
Stored results

Menu
Methods and formulas

Syntax
References

Options
Also see

Description

`gsdesign logrank` computes stopping boundaries and sample sizes for interim analyses of group sequential clinical trials performing survival analysis by using a log-rank test to compare survivor functions. Stopping can be for efficacy, futility, or both. For stopping boundary calculations without sample sizes, see [ADAPT] [gsbounds](#). For sample-size calculations for a fixed-sample study using a log-rank test, see [PSS-2] [power logrank](#).

Quick start

Sample size and stopping boundaries for the log-rank test of $H_0 : \text{hazard ratio } \Delta = 1$ versus two-sided alternative $H_a : \Delta \neq 1$, with default familywise significance level $\alpha = 0.05$ and power of 0.8 to detect a hazard ratio of $\Delta_a = 0.737$ without censoring, using default group sequential specifications of O'Brien–Fleming efficacy boundaries with two analyses (one interim, one final)

```
gsdesign logrank, hratio(0.737)
```

Same as above, specified as a log hazard-ratio of -0.305

```
gsdesign logrank, lnhratio(-0.305)
```

Same as above, but use Schoenfeld's sample-size calculation instead of the default Freedman method

```
gsdesign logrank, lnhratio(-0.305) schoenfeld
```

Sample size for censored design with survival probabilities $surv_1 = 0.5$ and $surv_2 = 0.6$, using a Kim–DeMets efficacy boundary with parameter $\rho_e = 3$ and analyses at 50%, 75%, and 100% of the total data

```
gsdesign logrank 0.5 0.6, efficacy(kdemets(3)) information(50 75 100)
```

Same as above, specified as $surv_1 = 0.5$ and hazard ratio of 0.737

```
gsdesign logrank 0.5, hratio(0.737) efficacy(kdemets(3)) ///
information(50 75 100)
```

Same as above, but add a binding O'Brien–Fleming-style futility bound

```
gsdesign logrank 0.5, hratio(0.737) efficacy(kdemets(3)) ///
futility(errob Fleming, binding) information(50 75 100)
```

Same as above, but report fractional sample sizes and graph the boundaries

```
gsdesign logrank 0.5, hratio(0.737) nfractional efficacy(kdemets(3)) ///
futility(errob Fleming, binding) information(50 75 100) ///
graphbounds
```

Menu

Statistics > Power, precision, and sample size

Syntax

```
gsdesign logrank [surv1 [surv2]] [ , logrankopts boundopts ]
```

where $surv_1$ is the survival probability in the control (reference) group at the end of the study, and $surv_2$ is the survival probability in the experimental (treatment) group at the end of the study.

logrankopts	Description
Main	
<code>alpha(#)</code>	overall significance level for all tests; default is <code>alpha(0.05)</code>
<code>power(#)</code>	overall power for all tests; default is <code>power(0.8)</code>
<code>beta(#)</code>	overall probability of type II error for all tests; default is <code>beta(0.2)</code>
<code>onesided</code>	request a one-sided test; default is two-sided
<code>nfractional</code>	report fractional sample size
<code>nratio(#)</code>	ratio of sample sizes of experimental to control groups; default is <code>nratio(1)</code> , meaning equal group sizes
<code>hratio(#)</code>	hazard ratio of the experimental to the control group; default is <code>hratio(0.5)</code> ; may not be combined with <code>lnhratio()</code>
<code>lnhratio(#)</code>	log hazard-ratio of the experimental to the control group; may not be combined with <code>hratio()</code>
<code>schoenfeld</code>	use the formula based on the log hazard-ratio in calculations; default is to use the formula based on the hazard ratio
<code>effect(effect)</code>	type of effect to display; default is <code>effect(hratio)</code> unless option <code>schoenfeld</code> is specified, in which case it is <code>effect(lnhratio)</code>
Censoring	
<code>simpson(## # matname)</code>	survival probabilities in the control group at three specific time points to compute the probability of an event (failure), using Simpson's rule under uniform accrual
<code>st1(varname_s varname_t)</code>	variables <code>varname_s</code> , containing survival probabilities in the control group, and <code>varname_t</code> , containing respective time points, to compute the probability of an event (failure), using numerical integration under uniform accrual
<code>wdprob(#)</code>	proportion of subjects anticipated to withdraw from the study; default is <code>wdprob(0)</code>
<code>force</code>	allow calculation with unsupported <code>power logrank</code> options

collect is allowed; see [U] 11.1.10 Prefix commands.
force does not appear in the dialog box.

<i>effect</i>	Description
<u>h</u> ratio	hazard ratio
<u>ln</u> hratio	log hazard-ratio

<i>boundopts</i>	Description
Bounds	
<u>e</u> fficacy(<i>boundary</i>)	boundary for efficacy stopping; if neither efficacy() nor futility() is specified, the default is efficacy(obfleming)
<u>f</u> utility(<i>boundary</i> [, <u>b</u> inding])	boundary for futility stopping; use binding to request binding futility bounds (default is nonbinding)
<u>n</u> looks(<u>#</u> [, equal])	total number of analyses (nlooks() – 1 interim analyses and one final analysis); use equal to enforce equal information increments; if neither nlooks() nor information() is specified, the default is nlooks(2)
<u>i</u> nformation(<i>numlist</i>)	sequence of information levels for analyses; default is evenly spaced
<u>n</u> opvalues	suppress <i>p</i> -values
Graph	
<u>g</u> raphbounds[(<i>graphopts</i>)]	graph boundaries
<u>m</u> atlistopts(<i>general_options</i>)	control the display of boundaries and sample size; seldom used
<i>optimopts</i>	optimization options for boundary calculations; seldom used

matlistopts() and *optimopts* do not appear in the dialog box.

<i>boundary</i>	Description
<u>o</u> b Fleming	classical O’Brien–Fleming bound
<u>p</u> ocock	classical Pocock bound
<u>w</u> tsiatis(<u>#</u>)	classical Wang–Tsiatis bound with specified parameter value
<u>e</u> rrpocock	error-spending Pocock-style bound
<u>e</u> rr o b Fleming	error-spending O’Brien–Fleming-style bound
<u>k</u> demets(<u>#</u>)	error-spending Kim–DeMets bound with specified parameter value
<u>h</u> sdecani(<u>#</u>)	error-spending Hwang–Shih–de Cani bound with specified parameter value

<i>graphopts</i>	Description
<u>xdimsampsize</u>	label the x axis with the sample size collected (default)
<u>xdiminformation</u>	label the x axis with the information fraction; use <code>information()</code> specified
<u>xdimlooks</u>	label the x axis with the number of each look
<u>noshade</u>	do not shade the rejection, acceptance, and continuation regions
<u>rejectopts</u> (<i>area_options</i>)	change the appearance of the rejection region
<u>acceptopts</u> (<i>area_options</i>)	change the appearance of the acceptance region
<u>continueopts</u> (<i>area_options</i>)	change the appearance of the continuation region
<u>efficacyopts</u> (<i>connected_options</i>)	change the appearance of the efficacy bound
<u>futilityopts</u> (<i>connected_options</i>)	change the appearance of the futility bound
<u>nolooklines</u>	do not draw vertical reference lines at each look
<u>looklinesopts</u> (<i>added_line_suboptions</i>)	change the appearance of the reference lines marking each look
<u>nofixed</u>	do not label critical values from a fixed study design
<u>fixedopts</u> (<i>marker_options</i>)	change the appearance of the fixed-study critical values
<u>twoway_options</u>	any options other than <code>by()</code> documented in [G-3] <i>twoway_options</i>

<i>optimopts</i>	Description
<u>intpointsscale</u> (#)	scaling factor for number of quadrature points; default is <code>intpointsscale(20)</code>
<u>initinfo</u> (<i>initinfo_spec</i>)	initial value(s) for maximum information
<u>initscale</u> (#)	initial value for scaling factor C of classical bounds
<u>infotolerance</u> (#)	tolerance for bisection search for maximum information of error-spending bounds with futility stopping; default is <code>infotol(1e-6)</code>
<u>marquardt</u>	use the Marquardt stepping algorithm in nonconcave regions; default is to use a mixture of steepest descent and Newton
<u>technique</u> (<i>algorithm_spec</i>)	maximization technique
<u>iterate</u> (#)	perform maximum of # iterations; default is <code>iterate(300)</code>
<u>[no]log</u>	display an iteration log; default is <code>nolog</code>
<u>trace</u>	display current parameter vector in iteration log
<u>gradient</u>	display current gradient vector in iteration log
<u>showstep</u>	report steps within an iteration in iteration log
<u>hessian</u>	display current negative Hessian matrix in iteration log
<u>showtolerance</u>	report the calculated result that is compared with the effective convergence criterion
<u>tolerance</u> (#)	tolerance for the parameter being optimized; default is <code>tolerance(1e-12)</code>
<u>ftolerance</u> (#)	tolerance for the objective function; default is <code>ftolerance(1e-10)</code>
<u>nrtolerance</u> (#)	tolerance for the scaled gradient; default is <code>nrtolerance(1e-16)</code>
<u>nonrtolerance</u>	ignore the <code>nrtolerance()</code> option

Options

Main

`alpha(#)` sets the overall significance level, which is the familywise type I error rate for all analyses (interim and final). `alpha()` must be in $(0, 0.5)$. The default is `alpha(0.05)`.

`power(#)` sets the overall power for all analyses. `power()` must be in $(0.5, 1)$. The default is `power(0.8)`. If `beta()` is specified, `power()` is set to be $1 - \text{beta}()$. Only one of `power()` or `beta()` may be specified.

`beta(#)` sets the overall probability of a type II error. `beta()` must be in $(0, 0.5)$. The default is `beta(0.2)`. If `power()` is specified, `beta()` is set to be $1 - \text{power}()$. Only one of `beta()` or `power()` may be specified.

`onesided` requests a study design for a one-sided test. The direction of the test is inferred from the effect size.

`nfractional` specifies that fractional sample sizes be reported.

`nratio(#)` specifies the sample-size ratio of the experimental group relative to the control group, $N2/N1$. The default is `nratio(1)`, meaning equal allocation between the two groups.

`hratio(#)` specifies the hazard ratio (effect size) of the experimental group to the control group. The default is `hratio(0.5)`. This value typically defines the clinically significant improvement of the experimental procedure over the control procedure desired to be detected by the log-rank test with a certain power.

You can specify an effect size either as a hazard ratio in `hratio()` or as a log hazard-ratio in `lnhratio()`. The default is `hratio(0.5)`. If both arguments surv_1 and surv_2 are specified, `hratio()` is not allowed and the hazard ratio is instead computed as $\ln(\text{surv}_2) / \ln(\text{surv}_1)$.

This option may not be combined with `lnhratio()`.

`lnhratio(#)` specifies the log hazard-ratio (effect size) of the experimental group to the control group. This value typically defines the clinically significant improvement of the experimental procedure over the control procedure desired to be detected by the log-rank test with a certain power.

You can specify an effect size either as a hazard ratio in `hratio()` or as a log hazard-ratio in `lnhratio()`. The default is `hratio(0.5)`. If both arguments surv_1 and surv_2 are specified, `lnhratio()` is not allowed and the log hazard-ratio is computed as $\ln\{\ln(\text{surv}_2) / \ln(\text{surv}_1)\}$.

This option may not be combined with `hratio()`.

`schoenfeld` requests calculations using the formula based on the log hazard-ratio, according to [Schoenfeld \(1981\)](#). The default is to use the formula based on the hazard ratio, according to [Freedman \(1982\)](#). See the [technical note](#) in [\[PSS-2\] power logrank](#) for a comparison of the two formulas.

`effect(effect)` specifies the type of effect size to be reported in the output as `delta`. `effect` is one of `hratio` or `lnhratio`. By default, the effect size `delta` is a hazard ratio, `effect(hratio)`, for a hazard-ratio test and a log hazard-ratio, `effect(lnhratio)`, for a log hazard-ratio test (when `schoenfeld` is specified).

Censoring

`simpson(###|matname)` specifies survival probabilities in the control group at three specific time points to compute the probability of an event (failure) using Simpson's rule under the assumption of uniform accrual. Either the actual values or a 1×3 matrix, `matname`, containing these values can

be specified. By default, the probability of an event is approximated as an average of the failure probabilities $1-s_1$ and $1-s_2$; see *Methods and formulas* in [PSS-2] **power logrank**. `simpson()` may not be combined with `st1()` and may not be used if command argument `surv1` or `surv2` is specified.

`st1(varnames varnamet)` specifies variables `varnames`, containing survival probabilities in the control group, and `varnamet`, containing respective time points, to compute the probability of an event (failure) using numerical integration under the assumption of uniform accrual; see [R] **dydx**. The minimum and the maximum values of `varnamet` must be the length of the follow-up period and the duration of the study, respectively. By default, the probability of an event is approximated as an average of the failure probabilities $1-s_1$ and $1-s_2$; see *Methods and formulas* in [PSS-2] **power logrank**. `st1()` may not be combined with `simpson()` and may not be used if command argument `surv1` or `surv2` is specified.

`wdprob(#)` specifies the proportion of subjects anticipated to withdraw from the study. The default is `wdprob(0)`.

Bounds

`efficacy(boundary)` specifies the boundary for efficacy stopping. If neither `efficacy()` nor `futility()` is specified, the default is `efficacy(obfleming)`.

`futility(boundary[, binding])` specifies the boundary for futility stopping.

`binding` specifies binding futility bounds. With binding futility bounds, if the result of an interim analysis crosses the futility boundary and lies in the acceptance region, the trial must end or risk overrunning the specified type I error. With nonbinding futility bounds, the trial does not need to stop if the result of an interim analysis crosses the futility boundary; the familywise type I error rate is controlled even if the trial continues. By default, futility bounds are nonbinding.

`nlooks([#, equal])` specifies the total number of analyses to be performed (`nlooks()` – 1 interim analyses and one final analysis). If neither `nlooks()` nor `information()` is specified, the default is `nlooks(2)`.

`equal` indicates that equal information increments be enforced, which is to say that the same number of new observations will be collected at each look. The default behavior is to start by dividing information evenly among looks, then proceed by rounding up to a whole number of observations at each look. This can cause slight differences in the information collected at each look.

`information(numlist)` specifies a sequence of information levels for interim and final analyses. This must be a sequence of increasing positive numbers, but the scale is unimportant because the information sequence will be automatically rescaled to ensure the **maximum information** is reached at the final look. By default, analyses are evenly spaced.

`nopvalues` suppresses the *p*-values from being reported in the table of boundaries for each look.

Graph

`graphbounds` and `graphbounds(graphopts)` produce graphical output showing the stopping boundaries.

graphopts are the following:

`xdimsampsize` labels the *x* axis with the sample size collected (the default).

`xdiminformation` labels the *x* axis with the information fraction unless `information()` is specified, in which case information levels will be used.

`xdimlooks` labels the *x* axis with the number of each look.

`noshade` suppresses shading of the rejection, acceptance, and continuation regions of the graph.

`rejectopts`(*area_options*) affects the rendition of the rejection region. See [G-3] *area_options*.

`acceptopts`(*area_options*) affects the rendition of the acceptance region. See [G-3] *area_options*.

`continueopts`(*area_options*) affects the rendition of the continuation region. See [G-3] *area_options*.

`efficacyopts`(*connected_options*) affects the rendition of the efficacy bound. See [G-3] *cline_options* and [G-3] *marker_options*.

`futilityopts`(*connected_options*) affects the rendition of the futility bound. See [G-3] *cline_options* and [G-3] *marker_options*.

`nolooklines` suppresses the vertical reference lines drawn at each look.

`looklinesopts`(*added_line_suboptions*) affects the rendition of reference lines marking each look. See *suboptions* in [G-3] *added_line_options*.

`nofixed` suppresses the fixed-study critical values in the plot.

`fixedopts`(*marker_options*) affects the rendition of the fixed-study critical values. See [G-3] *marker_options*.

twoway_options are any of the options documented in [G-3] *twoway_options*, excluding by(). These include options for titling the graph (see [G-3] *title_options*) and for saving the graph to disk (see [G-3] *saving_option*).

The following options are available with `gsdesign logrank` but are not shown in the dialog box:

`force` indicates that `gsdesign logrank` should allow unsupported *power logrank* options, such as options specifying a cluster randomized design. Even with option `force`, the *power logrank* options specified must be compatible with sample-size determination, not effect size or power calculation. In addition, *numlists* are not supported in options or in arguments as they are with *power*, even when `force` is specified.

`matlistopts`(*general_options*) affects the display of the matrix of boundaries and sample sizes. *general_options* are `title()`, `tindent()`, `rowtitle()`, `showcoleq()`, `coleqonly`, `colorcoleq()`, `aligncolnames()`, and `linesize()`; see *general_options* in [P] *matlist*. This option is seldom used.

optmopts control the iterative algorithm used to calculate stopping boundaries:

`intpointsscale`(#) specifies the scaling factor for the number of quadrature points used during the numerical evaluation of stopping probabilities at each look. The default is `intpointsscale(20)`. See *Methods and formulas* in [ADAPT] *gsbounds*.

`initinfo`(*initinfo_spec*) specifies either one or two initial values to be used in the iterative calculation of the *maximum information*.

The syntax `initinfo`(#) is applicable when using classical group sequential boundaries (Pocock bounds, O'Brien–Fleming bounds, and Wang–Tsatis bounds), as well as with efficacy-only stopping when using error-spending boundaries (error-spending Pocock-style efficacy bounds, error-spending O'Brien–Fleming-style efficacy bounds, Kim–DeMets efficacy bounds, and Hwang–Shih–de Cani efficacy bounds). The default is to use the information from a fixed study design; see *Methods and formulas* in [ADAPT] *gsbounds*.

The syntax `initinfo(# #)` is applicable when using error-spending group sequential boundaries with futility stopping (error-spending Pocock-style bounds, error-spending O’Brien–Fleming-style bounds, Kim–DeMets bounds, and Hwang–Shih–de Cani bounds). With this syntax, the first and second numbers specify the lower and upper starting values, respectively, for the bisection algorithm estimating the maximum information. The default is to use the information from a fixed study design for the lower initial value and the information corresponding to a Bonferroni correction for the upper initial value; see *Methods and formulas* in [ADAPT] **gsbounds**. To specify just the lower starting value, use `initinfo(# .)`, and to specify just the upper starting value, use `initinfo(. #)`.

`initscale(#)` specifies the initial value to be used during the iterative calculation of *scaling factor* C for classical group sequential boundaries (Pocock bounds, O’Brien–Fleming bounds, and Wang–Tsiatis bounds). The default is to use the z -value corresponding to the specified value of `alpha()`. See *Methods and formulas* in [ADAPT] **gsbounds**.

`infotolerance(#)` specifies the tolerance for the bisection algorithm used in the iterative calculation of the maximum information of error-spending group sequential boundaries with futility stopping (error-spending Pocock-style bounds, error-spending O’Brien–Fleming-style bounds, Kim–DeMets bounds, and Hwang–Shih–de Cani bounds). The default is `infotolerance(1e-6)`. See *Methods and formulas* in [ADAPT] **gsbounds**.

`marquardt` specifies that the optimizer should use the modified Marquardt algorithm when, at an iteration step, it finds that H is singular. The default is to use a mixture of steepest descent and Newton, which is equivalent to the `difficult` option in [R] **ml**.

`technique(algorithm_spec)` specifies how the objective function is to be maximized. The following algorithms are allowed. For details, see Pitblado, Poi, and Gould (2024).

`technique(bfgs)` specifies the Broyden–Fletcher–Goldfarb–Shanno (BFGS) algorithm.

`technique(nr)` specifies Stata’s modified Newton–Raphson (NR) algorithm.

`technique(dfp)` specifies the Davidon–Fletcher–Powell (DFP) algorithm.

The default is `technique(bfgs)` when using classical group sequential boundaries (Pocock bounds, O’Brien–Fleming bounds, and Wang–Tsiatis bounds) and also for the second optimization step used to estimate the maximum information with efficacy-only stopping when using error-spending boundaries (error-spending Pocock-style efficacy bounds, error-spending O’Brien–Fleming-style efficacy bounds, Kim–DeMets efficacy bounds, and Hwang–Shih–de Cani efficacy bounds). The default is `technique(nr)` for the sequential optimization steps used to estimate critical values for error-spending boundaries. You can also switch between two algorithms by specifying the technique name followed by the number of iterations. For example, specifying `technique(nr 10 bfgs 20)` requests 10 iterations with the NR algorithm followed by 20 iterations with the BFGS algorithm, and then back to NR for 10 iterations, and so on. The process continues until convergence or until the maximum number of iterations is reached.

`iterate(#)` specifies the maximum number of iterations. If convergence is not declared by the time the number of iterations equals `iterate()`, an error message is issued. The default value of `iterate(#)` is the number set using `set maxiter`, which is 300 by default.

`[no]log` requests an iteration log showing the progress of the optimization. The default is `nolog`.

`trace` adds to the iteration log a display of the current parameter vector.

`gradient` adds to the iteration log a display of the current gradient vector.

`showstep` adds to the iteration log a report on the steps within an iteration. This option was added so that developers at StataCorp could view the stepping when they were improving the `ml` optimizer code. At this point, it mainly provides entertainment.

`hessian` adds to the iteration log a display of the current negative Hessian matrix.

`showtolerance` adds to the iteration log the calculated value that is compared with the effective convergence criterion at the end of each iteration. Until convergence is achieved, the smallest calculated value is reported. `shownrtolerance` is a synonym of `showtolerance`.

Below, we describe the three convergence tolerances. Convergence is declared when the `nrtolerance()` criterion is met and either the `tolerance()` or the `ftolerance()` criterion is also met.

`tolerance(#)` specifies the tolerance for the parameter vector. When the relative change in the parameter vector from one iteration to the next is less than or equal to `tolerance()`, the `tolerance()` convergence criterion is satisfied. The default is `tolerance(1e-12)`.

`ftolerance(#)` specifies the tolerance for the objective function. When the relative change in the objective function from one iteration to the next is less than or equal to `ftolerance()`, the `ftolerance()` convergence is satisfied. The default is `ftolerance(1e-10)`.

`nrtolerance(#)` specifies the tolerance for the scaled gradient. Convergence is declared when $\mathbf{gH}^{-1}\mathbf{g}' < \text{nrtolerance}()$. The default is `nrtolerance(1e-16)`.

`nonnrtolerance` specifies that the default `nrtolerance()` criterion be turned off.

boundary

`obfleming` specifies a classical O'Brien–Fleming design for efficacy or futility bounds (O'Brien and Fleming 1979). O'Brien–Fleming efficacy bounds are characterized by being extremely conservative at early looks. The O'Brien–Fleming design is a member of the Wang–Tsiatis family and is equivalent to specifying a *boundary* of `wtisiatis(0)`.

`pocock` specifies a classical Pocock design for efficacy or futility bounds (Pocock 1977). Pocock efficacy bounds are characterized by using the same critical value at all looks. The Pocock design is a member of the Wang–Tsiatis family and is equivalent to specifying a *boundary* of `wtisiatis(0.5)`.

`wtisiatis(#)` specifies a classical Wang–Tsiatis design for efficacy or futility bounds (Wang and Tsiatis 1987). The shape of Wang–Tsiatis bounds is determined by parameter $\Delta \in [-10, 0.7]$, where smaller values of Δ yield bounds that are more conservative at early looks.

`errpocock` specifies an error-spending Pocock-style design for efficacy or futility bounds (Lan and DeMets 1983). The critical values from error-spending Pocock-style bounds are very similar to those of classic Pocock bounds, but they are obtained using an error-spending function.

`errobfleming` specifies an error-spending O’Brien–Fleming-style design for efficacy or futility bounds (Lan and DeMets 1983). The critical values from error-spending O’Brien–Fleming-style bounds are very similar to those of classic O’Brien–Fleming bounds, but they are obtained using an error-spending function.

`kdemets`(#) specifies an error-spending Kim–DeMets design for efficacy or futility bounds (Kim and DeMets 1987). The shape of Kim–DeMets bounds is determined by power parameter $\rho \in (0, 10]$, where larger values of ρ yield bounds that are more conservative at early looks.

`hsdecani`(#) specifies an error-spending Hwang–Shih–de Cani design for efficacy or futility bounds (Hwang, Shih, and de Cani 1990). The shape of Hwang–Shih–de Cani bounds is determined by parameter $\gamma \in [-30, 3]$, where smaller values of γ yield bounds that are more conservative at early looks.

For a design with both efficacy and futility stopping boundaries, if you specify a classical boundary (that is, in the Wang–Tsiatis family) for one, then you must specify a classical boundary for the other. So, you could not specify a boundary in the Wang–Tsiatis family for one boundary and an error-spending boundary for the other. When specifying efficacy and futility boundaries from the same family, the efficacy parameter does not need to be the same as the futility parameter.

Boundaries that are conservative at early looks, such as the O’Brien–Fleming bound, offer little chance of early stopping unless the true effect size is quite large (in the case of efficacy bounds) or quite small (in the case of futility bounds). A trial employing a conservative bound is more likely to continue to the final look, yielding an expected sample size that is not dramatically smaller than the sample size required by an equivalent fixed-sample trial. However, the maximum sample size (that is, the sample size at the final look) of a trial with a conservative bound is generally not much greater than the sample size required by an equivalent fixed trial. Another direct result of specifying conservative bounds is that the critical value at the final look tends to be close to the critical value employed by an equivalent fixed design. In contrast, anticonservative boundaries such as the Pocock bound offer a much better shot at early stopping (often yielding a small expected sample size) at the cost of a larger maximum sample size and final critical values that are considerably larger than the critical value of an equivalent fixed design.

Remarks and examples

Remarks are presented under the following headings:

[Introduction](#)

[Using `gsdesign logrank`](#)

[Background for examples](#)

[Computing sample size and boundaries in the absence of censoring](#)

[Computing sample size and boundaries in the presence of censoring](#)

[Computing sample size and boundaries with uniform accrual](#)

This entry describes the use of the `gsdesign logrank` command for designing a group sequential analysis for a two-sample comparison of survivor functions using the log-rank test. See [\[ADAPT\] GSD intro](#) for a general introduction to group sequential designs (GSDs) for clinical trials; see [\[ADAPT\] gs-bounds](#) for information about group sequential bounds; and see [\[ADAPT\] gsdesign](#) for information about designing group sequential clinical trials with the `gsdesign` command. Also see [\[PSS-2\] Intro \(power\)](#) for a general introduction to power and sample-size analysis, and see [\[PSS-2\] power logrank](#) for details about fixed-sample study designs for a log-rank test of two survivor functions.

Introduction

When analyzing time-to-event data, we frequently use the failure function, the survivor function, and the hazard function, denoted $F(t)$, $S(t)$, and $h(t)$, respectively. The failure function is the probability of experiencing a failure event at or before time t . If we denote the time of failure as T , we can define the failure function as the cumulative distribution function of T , where $F(t) = \Pr(T \leq t)$. The probability density function of T is the derivative of the failure function with respect to time, written as $f(t) = \partial F(t)/\partial t$. The survivor function is defined as the probability of surviving beyond time t , expressed mathematically as $S(t) = \Pr(T > t) = 1 - F(t)$. The hazard function at time t is the instantaneous rate of failure at time t , conditional on survival until time t , written as $h(t) = f(t)/S(t)$.

Consider a survival study comparing survivor functions in two groups using the log-rank test. Let $S_1(t)$ and $S_2(t)$ denote the survivor functions of the control and the experimental groups, respectively. The log-rank test is the most powerful nonparametric test of $S_1(\cdot) = S_2(\cdot)$ if the hazard functions are proportional. That is, $h_2(t) = \Delta h_1(t)$ for all t or, equivalently, $S_2(t) = \{S_1(t)\}^\Delta$, where Δ is the hazard ratio. If $\Delta < 1$, survival in the experimental group is higher than survival in the control group, which means that the experimental treatment is superior to the control treatment. If $\Delta > 1$, then the control treatment is superior to the experimental treatment. Under the proportional-hazards assumption, the test of the equality of the two survivor functions $H_0: S_1(t) = S_2(t)$ versus $H_a: S_1(t) \neq S_2(t)$ is equivalent to testing $H_0: \Delta = 1$ versus $H_a: \Delta \neq 1$ or $H_0: \ln(\Delta) = 0$ versus $H_a: \ln(\Delta) \neq 0$.

The methods implemented in `gsdesign logrank` for boundary and sample-size calculations relate the power of the log-rank test directly to the number of events observed in the study. The required sample size is equal to the required number of events if a failure event is observed for every participant in the trial. Often, the time of failure is not known for some participants, a phenomenon known as censoring. [Administrative censoring](#) occurs when a trial ends before all participants have experienced a failure event. Nonadministrative censoring occurs when participants withdraw from the study or are lost to follow-up. If censoring occurs in the study, the required number of participants will be greater than the required number of events. In the presence of administrative censoring or withdrawal, `gsdesign logrank` requires additional information to estimate the probability that a participant's failure time will be observed.

Using `gsdesign logrank`

`gsdesign logrank` computes stopping boundaries and sample size for a log-rank test comparing the survivor functions in two groups. `gsdesign logrank` can be thought of as a combination of [power logrank](#) for sample-size calculations and [gsbounds](#) for stopping boundary calculations. `gsdesign logrank` supports two methods of estimating the required sample size: the method of [Freedman \(1982\)](#), which uses a formula based on the hazard ratio and is the default, and the method of [Schoenfeld \(1981\)](#), which uses a formula based on the log hazard-ratio.

To determine the required number of events, the investigator must specify the effect size. Effect size is usually expressed as a hazard ratio, Δ_a , by using the `hratio()` option. Alternatively, you may specify the effect size as a log hazard-ratio, $\ln(\Delta_a)$, with the `lnhratio()` option. When administrative censoring is anticipated, the survival probabilities of the two groups, $surv_1$ and $surv_2$, may be specified and the effect size is calculated from the survival probabilities. If the effect size is not specified, a hazard ratio of 0.5 is assumed.

By default, all computations assume no censoring. In the presence of [administrative censoring](#), you must specify a survival probability at the end of the study in the control group as the first command argument, `surv1`. You can also specify a survival probability at the end of the study in the experimental group as the second command argument, `surv2`. Otherwise, it will be computed using the specified hazard ratio or log hazard-ratio and the control-group survival probability. To accommodate an [accrual period](#) under the assumption of uniform accrual, survival information may instead be supplied in the `simpson()` option or the `st1()` option; see [Including information about subject accrual](#) in [PSS-2] **power logrank** for details. To adjust the sample-size calculation for withdrawal from the trial, specify the anticipated proportion of withdrawals in the `wdprob()` option.

By default, `gsdesign logrank` assumes that the control and experimental arms will be the same size. If participants are not allocated equally between the two arms, the `nratio()` option is used to specify the ratio of participants in the experimental arm to the control arm.

Options `alpha()`, `power()`, `beta()`, and `onesided` are used for both sample-size and stopping-boundary calculations. The default significance level, known as the familywise type I error rate, is 0.05 and can be changed by specifying the `alpha()` option. The default power is 0.8, which corresponds to a type II error rate of 0.2. This can be modified either by specifying the power in the `power()` option or by specifying the type II error in the `beta()` option. The default test is two-sided, and the `onesided` option requests a one-sided test, the direction of which is indicated by the effect size.

The group sequential stopping rule is determined by the `efficacy()` and `futility()` options. Stopping can be for efficacy, futility, or both, and if no stopping rule is specified, the default is to use an O’Brien–Fleming efficacy bound. If futility bounds are requested, the default behavior is to treat them as nonbinding. A trial that crosses a nonbinding futility bound can be stopped for futility, but the familywise type I error is controlled even if the trial continues. Binding futility bounds can be requested with `futility()` suboption `binding`. A trial that crosses a binding futility bound must be stopped for futility; if it continues, the familywise type I error will not be controlled at the specified significance level.

The number of looks, or analyses of the trial data, is specified with `nlooks()`. Alternatively, the `information()` option can be used to specify the spacing of the looks as a [numlist](#) of increasing information levels. In this case, values of the `numlist` are automatically rescaled so that the final look has the [maximum information](#) required by the design. If neither `nlooks()` nor `information()` is specified, the default is two looks.

By default, the sample sizes in each arm are rounded up to whole numbers at each look, but the `nfractional` option can be used to report fractional sample sizes. If `nlooks()` is specified, the default behavior is to divide information evenly among each look before rounding. Rounding can cause slight differences in the amount of information collected at each look, and `nlooks()` suboption `equal` can be specified to enforce equal information increments by requiring the same number of new observations per arm at each look.

Background for examples

In the following examples, we consider designing a clinical trial of a treatment for hepatocellular carcinoma, the most common type of primary liver cancer. In 2023, [Peng et al.](#) described the results of the LAUNCH trial, a phase 3 randomized controlled trial comparing lenvatinib monotherapy (the control arm) against lenvatinib plus transarterial chemoembolization (the experimental arm) as a treatment for primary advanced hepatocellular carcinoma. The primary endpoint of the trial was overall survival, the time from randomization to death from any cause.

Lenvatinib is an anti-cancer medication that can be taken orally and is used to treat some thyroid, kidney, and liver cancers, including hepatocellular carcinoma. Transarterial chemoembolization is a procedure where a catheter is inserted in the artery supplying blood to the tumor, and small particles with injectable anti-cancer drugs are introduced directly into the area of the tumor, blocking off the tumor's blood supply and providing a concentrated dose of chemotherapeutic medication.

Computing sample size and boundaries in the absence of censoring

► Example 1: GSD for a log-rank test with O'Brien–Fleming efficacy bounds

Peng et al. (2023) randomized participants to the control and experimental arms in a 1:1 ratio and conducted a log-rank test of $H_0: \Delta = 1$ versus the two-sided alternative $H_a: \Delta \neq 1$ with a familywise significance level of 5%. They required 90% power to detect a hazard ratio of $\Delta_a = 0.67$ and planned a single interim analysis using classical O'Brien–Fleming efficacy bounds once two-thirds of the data had been collected. We use `gsdesign logrank` to design and graph the boundaries of a clinical trial with these parameters.

```
. gsdesign logrank, hratio(0.67) power(0.9) efficacy(obfleming)
> information(0.667 1) graphbounds

Group sequential design for two-sample comparison of survivor functions
Log-rank test, Freedman method
H0: HR = 1 versus Ha: HR != 1
Efficacy: O'Brien-Fleming

Study parameters:
  alpha = 0.0500  (two-sided)
  power = 0.9000
  delta = 0.6700  (hazard ratio)
  hratio = 0.6700

Censoring:
  Pr_E = 1.0000

Expected number of events:
  H0 = 272.71
  Ha = 220.55

Info. ratio = 1.0155
  E fixed = 270
  N fixed = 270
  N max = 274
  N1 max = 137
  N2 max = 137

Fixed-study crit. values = ±1.9600

Critical values, p-values, and sample sizes for a group sequential design
```

Look	Info. frac.	Efficacy		p-value	Events	
		Lower	Upper			E
1	0.67	-2.4524	2.4524	0.0142		183
2	1.00	-2.0028	2.0028	0.0452		274

Note: Critical values are for z statistics; otherwise, use p-value boundaries.

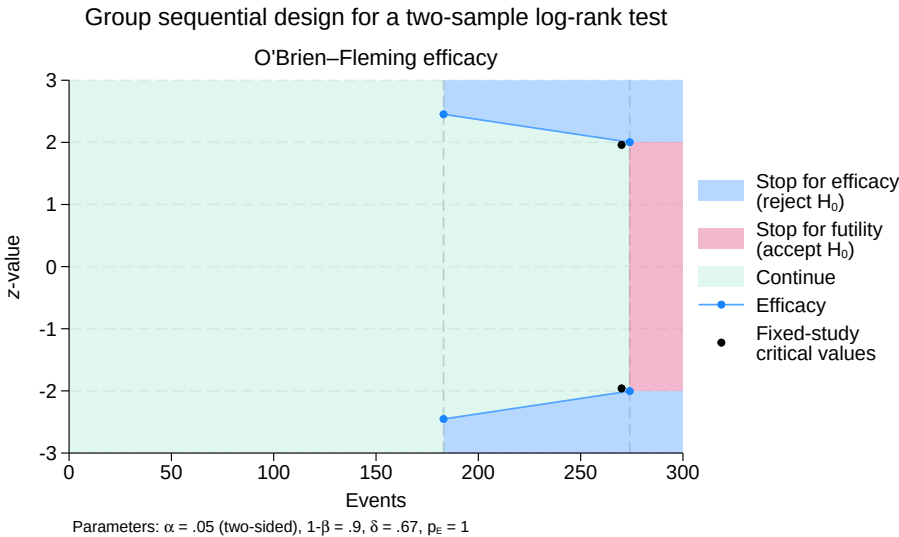


Figure 1. GSD with O'Brien–Fleming efficacy bounds for a two-sided log-rank test

`gsdesign logrank` displays the specified study parameters, including `hratio`, the hazard ratio under the alternative hypothesis, and `Pr_E`, the probability that a participant will die by the end of the study.

The next section of the output displays the expected number of events, which is the average number of events if the group sequential trial were to be repeated many times. The following section reports the information ratio, the sample size for a fixed study with an equivalent significance level and power (`N fixed`), the maximum sample size of the GSD (`N max`), and the maximum sample sizes for each group (`N1 max` and `N2 max`). The information ratio is the ratio of the maximum sample size of the GSD to the fixed-study sample size.

If the null hypothesis of $H_0 : \Delta = 1$ were true, the control and experimental arms of the trial would have equal hazards. In this case, the average trial would require 272.71 events, nearly the full sample size of 274. This is because the efficacy bounds do not allow for early stopping to accept H_0 , so if the null hypothesis is true, the trial will usually proceed to the final look. If H_a is true, the average trial will require 220.55 events, which is a savings over the 270 events required by the fixed trial.

The table at the end of the output displays the critical values for the stopping boundaries and the corresponding p -values as well as sample sizes at each look, where sample size is reported as the number of events observed. Boundary critical values are reported on the z scale and are designed to be compared against the z statistic from a log-rank test. Command `sts test` (see [ST] [sts test](#)) conducts the log-rank test and reports a χ^2 test statistic, which is not directly comparable with the z scale critical values. However, the square root of the χ^2 test statistic is a z statistic, which can be directly compared with the boundary critical values.

We planned the first look to occur with 66.7% of the data, which corresponds to 183 events. Examining the graph, we see that the entire region from 0 to 182 events is shaded green, the color of the continuation region. This is because the data have not yet been analyzed, so the trial cannot be stopped. The first look will be conducted once 183 deaths have occurred, and a log-rank test will be performed. We denote the square root of the χ^2 test statistic from the first look as z_1 and note that the sign of z_1 depends on whether the observed hazard ratio was greater than 1 (in which case z_1 is positive) or less than 1 (in which case z_1 is negative). If $|z_1| \geq 2.452$, we say that z_1 lies in the rejection region (shaded blue on the graph), and we reject H_0 , terminating the trial early due to treatment efficacy. If $|z_1| < 2.452$, it lies in the continuation region, and we proceed to the final look.

At the final look, there is no continuation region; H_0 must be rejected or accepted. While accepting the null hypothesis is taboo in many disciplines, it has a long history in the context of sequential trials (see [Origins of GSD](#) in [ADAPT] [GSD intro](#)). As before, we take the square root of the χ^2 test statistic and label it z_2 . If $|z_2| \geq 2.003$, then we reject H_0 and conclude that $\Delta \neq 1$, while if $|z_2| < 2.003$, then we accept H_0 .

◀

Computing sample size and boundaries in the presence of censoring

▷ Example 2: GSD for a log-rank test with censoring

In the previous example, we assumed no censoring would occur, so the failure time of all participants would be observed. That is often an unrealistic expectation, and here we adjust the sample size to account for censoring. We divide censoring into two types: administrative censoring, which occurs when the trial ends before all participants have experienced a failure event, and withdrawal, which occurs when a participant withdraws from the study or is lost to follow-up. `gsdesign logrank` takes a conservative

stance on withdrawal, assuming that participants who withdraw do so as soon as the study begins, before they can contribute meaningful data to the trial. For more information about censoring, see [Computing sample size in the presence of censoring](#) in [PSS-2] **power logrank**.

Peng et al. (2023) describe an anticipated withdrawal rate of 10%, which we will incorporate using the `wdprob()` option. Based on a previous study of lenvatinib as a treatment for hepatocellular carcinoma (Kudo et al. 2018), we anticipate that 5% of the participants in the control arm will be alive at the end of the trial. We include the control-group survival probability as command argument `surv1`.

```
. gsdesign logrank 0.05, hratio(0.67) wdprob(0.1) power(0.9)
> efficacy(obfleming) information(0.667 1)

Group sequential design for two-sample comparison of survivor functions
Log-rank test, Freedman method
H0: HR = 1 versus Ha: HR != 1
Efficacy: O'Brien-Fleming

Study parameters:
    alpha = 0.0500    (two-sided)
    power = 0.9000
    delta = 0.6700    (hazard ratio)
    hratio = 0.6700

Censoring and withdrawal:
    s1 = 0.0500
    s2 = 0.1344
    Pr_E = 0.9078
    Pr_w = 0.1000

Expected number of events:
    H0 = 272.71
    Ha = 220.55

Info. ratio = 1.0155
    E fixed =    270
    N fixed =    330
    N max =    336
    N1 max =    168
    N2 max =    168

Fixed-study crit. values = ±1.9600

Critical values, p-values, and sample sizes for a group sequential design
```

Look	Info. frac.	Efficacy		p-value	Events E
		Lower	Upper		
1	0.67	-2.4524	2.4524	0.0142	183
2	1.00	-2.0028	2.0028	0.0452	274

Note: Critical values are for z statistics; otherwise, use p-value boundaries.

In addition to `Pr_E`, the probability that an event is observed by the end of the study, the output of `gsdesign logrank` now includes additional information about censoring and withdrawal. The survival probabilities of the control and experimental arms are labeled `s1` and `s2`, respectively, and the probability of withdrawal is labeled `Pr_w`.

Now that censoring is incorporated into the design, we must recruit a larger sample of 168 participants in each arm, but the number of events at each look is unchanged, as are the critical values of the efficacy bound. This is to be expected because the power of the log-rank test is calculated in terms of the number of events observed, which is not affected by censoring.



Computing sample size and boundaries with uniform accrual

▷ Example 3: GSD for a log-rank test with uniform accrual

In [example 2](#), we considered the effect of censoring on the study design, but we did not account for the fact that the first participants recruited to the study would be observed for longer than the last participants to join. [Peng et al. \(2023\)](#) describe the recruitment period as lasting 24 months and the follow-up period as lasting an additional 24 months. The first participants to join the study would be monitored for up to 48 months (or until they died or withdrew from the study), while the last participants would only be monitored for 24 months.

If participants are recruited to the study at a uniform rate, [Schoenfeld \(1983\)](#) recommends a sample-size calculation that involves estimating the integral of the survivor function using Simpson's rule. `gsdesign logrank` implements this calculation with the `simpson()` option, which requires estimates of the survival probability in the control group at three points: at the minimum follow-up time, halfway between the minimum and maximum follow-up times, and at the maximum follow-up time. This corresponds to 24 months, 36 months, and 48 months in the LAUNCH study.

Based on the previous work of [Kudo et al. \(2018\)](#) and the assumptions made by [Peng et al. \(2023\)](#) about the tumor burden in their population of interest, we predict that 20% of participants in the control arm will be alive 24 months after they join the study, 10% will be alive after 36 months, and 5% will be alive after 48 months.

```
. gsdesign logrank, hratio(0.67) simpson(0.2 0.1 .05) wdprob(0.1) power(0.9)
> efficacy(obfleming) information(0.667 1)

Group sequential design for two-sample comparison of survivor functions
Log-rank test, Freedman method
H0: HR = 1 versus Ha: HR != 1
Efficacy: O'Brien-Fleming

Study parameters:
  alpha = 0.0500   (two-sided)
  power = 0.9000
  delta = 0.6700   (hazard ratio)
  hratio = 0.6700

Censoring and withdrawal:
  Pr_E = 0.8350
  Pr_w = 0.1000

Expected number of events:
  H0 = 272.71
  Ha = 220.55

Info. ratio = 1.0155
  E fixed =    270
  N fixed =    360
  N max =    364
  N1 max =    182
  N2 max =    182

Fixed-study crit. values = ±1.9600
```

Critical values, p-values, and sample sizes for a group sequential design

Look	Info. frac.	Efficacy		p-value	Events E
		Lower	Upper		
1	0.67	-2.4524	2.4524	0.0142	183
2	1.00	-2.0028	2.0028	0.0452	274

Note: Critical values are for z statistics; otherwise, use p-value boundaries.

In [example 2](#), we calculated the overall probability of observing a failure event, Pr_E , to be 0.908. After taking accrual into account, we now estimate Pr_E to be 0.835. The reduced chance of observing a failure event corresponds to a larger required sample size: 360 participants for a fixed-sample trial (up from 330) and a maximum of 364 participants for the GSD (up from 336).

Altering the assumptions about participant accrual affected the required sample size but not the number of failures needed to attain 90% power or the critical values required to achieve a 5% familywise significance level.



▷ **Example 4: GSD for a one-sided log-rank test with efficacy and futility stopping**

In the previous examples, we endeavored to design a clinical trial modeled after the LAUNCH study with increasingly sophisticated estimates of the probability of observing a failure event. We did not modify the details of the interim analysis, a relatively simple design with an O’Brien–Fleming efficacy bound and a single look at 66.7% of the total number of failure events. Here we depart from the design of [Peng et al. \(2023\)](#) and calculate stopping bounds and sample sizes for a study with both efficacy and futility stopping.

Futility stopping is the complement of efficacy stopping, and it allows the trial to end early if interim results are overwhelmingly unfavorable. This is done by calculating a futility bound, which we will draw on the graph in red. Much as the efficacy bound shown in [figure 1](#) divided the range of interim test statistics into continuation and rejection regions, the futility bound further partitions the range of test statistics by defining an acceptance region. With efficacy-only stopping, the acceptance region only existed at the final look, but with futility stopping, it is possible to accept H_0 before the scheduled end of the trial.

There are two types of futility boundaries: nonbinding (the default) and binding. In a trial with binding futility bounds, if an interim test statistic lies in the acceptance region, the trial must be terminated for futility; if it continues, the familywise type I error will not be controlled at the desired significance level. In contrast, if an interim test statistic crosses a nonbinding futility bound, the [Data Monitoring Committee](#) can decide to halt the trial or allow it to continue without risk of overrunning the specified alpha level.

We choose to implement a nonbinding Hwang–Shih–de Cani futility boundary with parameter $\gamma_f = -4$. Hwang–Shih–de Cani bounds are calculated using the error-spending approach, which makes them incompatible with classical O’Brien–Fleming bounds. Fortunately, there is an error-spending approximation of the O’Brien–Fleming bound that we can use instead (see [Methods and formulas](#) in [\[ADAPT\] gsbounds](#) for more information about classical and error-spending bounds). We specify error-spending O’Brien–Fleming-style efficacy bounds with the `efficacy(errob Fleming)` option.

We anticipate that the hazard ratio will be less than 1, indicating that the experimental treatment of lenvatinib plus transarterial chemoembolization is superior to the control treatment of lenvatinib alone. As such, we request a one-sided test and reduce the significance level to half of what it was with a two-sided alternative hypothesis. Finally, we add an additional interim analysis once half of the total number of failure events have been observed.

```
. gsdesign logrank, hratio(0.67) simpson(0.2 0.1 0.05) wdprob(0.1) onesided
> power(0.9) alpha(0.025) efficacy(errobfleming)
> futility(hsdecani(-4)) information(0.5 0.667 1) graphbounds
```

Group sequential design for two-sample comparison of survivor functions
Log-rank test, Freedman method
H0: HR = 1 versus Ha: HR < 1

Efficacy: Error-spending O'Brien-Fleming style
Futility: Error-spending Hwang-Shih-de Cani, nonbinding, gamma = -4.0000

Study parameters:
alpha = 0.0250 (lower one-sided)
power = 0.9000
delta = 0.6700 (hazard ratio)
hratio = 0.6700

Censoring and withdrawal:
Pr_E = 0.8350
Pr_w = 0.1000

Expected number of events:
H0 = 183.14
Ha = 211.81

Info. ratio = 1.0306
E fixed = 270
N fixed = 360
N max = 370
N1 max = 185
N2 max = 185

Fixed-study crit. value = -1.9600

Critical values, p-values, and sample sizes for a group sequential design

Look	Info. frac.	Efficacy		Futility		Events E
		Lower	p-value	Upper	p-value	
1	0.50	-2.9626	0.0015	-0.0672	0.4732	139
2	0.67	-2.5374	0.0056	-0.6491	0.2581	185
3	1.00	-1.9945	0.0230	-1.9945	0.0230	278

Note: Critical values are for z statistics; otherwise, use p-value boundaries.

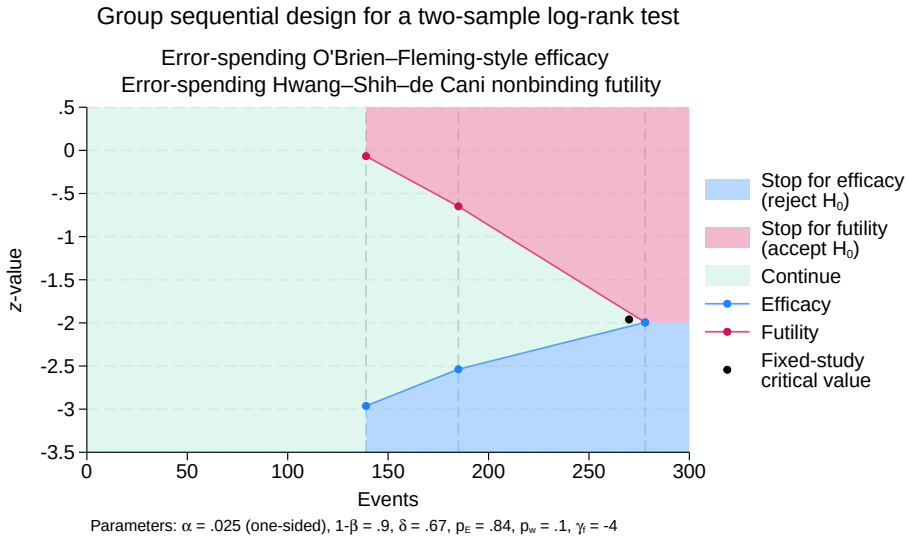


Figure 2. Error-spending efficacy and futility bounds for a one-sided log-rank test

The output of `gsdesign logrank` starts off quite similar to that of [example 3](#), but the alternative hypothesis is now reported as $H_a: HR < 1$. By halving the significance level when transitioning to a one-sided test, we have kept the number of participants and events required by a fixed study unchanged.

Compared with the design in [example 3](#), the maximum sample size has increased from 364 to 370, but the expected number of events has changed much more dramatically. Without futility stopping, 272.71 events were expected under H_0 , but that has decreased to 183.14 events now that the trial can be stopped early to accept the null hypothesis. The expected number of events under H_a has decreased as well, from 220.55 to 211.81, due to the additional opportunity to stop the trial for efficacy once half of the data have been collected.

Once 139 events have been recorded, the log-rank test is conducted and z_1 , the square root of the χ^2 statistic, is calculated. If the hazard ratio is less than 1, then z_1 is negative; if the hazard ratio is greater than 1 (meaning the control is outperforming the experimental treatment), then z_1 is positive. If $z_1 \leq -2.963$, then the test statistic lies within the rejection region, so we reject H_0 and terminate the trial early due to treatment efficacy. If $z_1 > -0.067$, then it lies within the acceptance region and we have the option of terminating the trial due to futility. If z_1 lies in the continuation region of $(-2.963, -0.067]$, then the trial must continue.

The second look occurs once 185 failures have been observed, and the testing procedure is similar except the continuation region has shrunk to $(-2.537, -0.649]$. At the final look, the efficacy and futility critical values are the same, leaving no continuation region. If $z_3 \leq -1.995$, then we reject H_0 ; otherwise, H_0 is accepted.

Stored results

gsdesign logrank stores the following in `r()`:

Scalars

<code>r(alpha)</code>	overall significance level (familywise type I error)
<code>r(beta)</code>	overall probability of a type II error
<code>r(binding)</code>	1 for binding futility bounds, 0 for nonbinding
<code>r(delta)</code>	effect size
<code>r(E_fixed)</code>	number of events in a fixed study design
<code>r(E_fixedfrac)</code>	fractional number of events in a fixed study design
<code>r(E_max)</code>	maximum number of events if the study continues to completion
<code>r(efparam)</code>	efficacy parameter (if <code>wtsiatis()</code> , <code>kdemets()</code> , or <code>hsdecani()</code> specified)
<code>r(ESS0)</code>	expected sample size under null hypothesis
<code>r(ESS1)</code>	expected sample size under alternative hypothesis
<code>r(futparam)</code>	futility parameter (if <code>wtsiatis()</code> , <code>kdemets()</code> , or <code>hsdecani()</code> specified)
<code>r(hratio)</code>	hazard ratio (unless <code>lnhratio()</code> specified)
<code>r(info_ratio)</code>	ratio of maximum information required to that of a fixed study design
<code>r(lnhratio)</code>	log hazard-ratio (if <code>lnhratio()</code> specified)
<code>r(N_fixed)</code>	sample size of a fixed study design
<code>r(N_fixedfrac)</code>	fractional sample size of a fixed study design
<code>r(N_max)</code>	maximum sample size if the study continues to completion
<code>r(N1_fixed)</code>	sample size of the control group in a fixed study design
<code>r(N1_fixedfrac)</code>	fractional sample size of the control group in a fixed study design
<code>r(N1_max)</code>	maximum sample size of the control group if the study continues to completion
<code>r(N2_fixed)</code>	sample size of the experimental group in a fixed study design
<code>r(N2_fixedfrac)</code>	fractional sample size of the experimental group in a fixed study design
<code>r(N2_max)</code>	maximum sample size of the experimental group if the study continues to completion
<code>r(nfractional)</code>	1 if <code>nfractional</code> is specified, 0 otherwise
<code>r(nlooks)</code>	number of analyses
<code>r(nratio)</code>	specified ratio of sample sizes, $N2/N1$
<code>r(nratio_a)</code>	attained ratio of sample sizes
<code>r(onesided)</code>	1 for a one-sided test, 0 otherwise
<code>r(power)</code>	specified overall power
<code>r(power_a)</code>	attained overall power
<code>r(Pr_E)</code>	probability of an event (failure)
<code>r(Pr_w)</code>	proportion of withdrawals
<code>r(s1)</code>	survival probability in the control group (if specified)
<code>r(s2)</code>	survival probability in the experimental group (if specified)
<code>r(stop)</code>	0 for futility bounds, 1 for efficacy bounds, 2 for both
<code>r(t_min)</code>	minimum time (if <code>st1()</code> specified)
<code>r(t_max)</code>	maximum time (if <code>st1()</code> specified)
<code>r(z_fixed)</code>	critical value for an equivalent fixed study design

Macros

<code>r(cmd)</code>	gsdesign
<code>r(cmdline)</code>	command as typed
<code>r(direction)</code>	upper, lower, or two-sided
<code>r(efbnd)</code>	pocock, obfleming, wtsiatis, errpocock, errobfleming, kdemets, or hsdecani
<code>r(effect)</code>	hratio or lnhratio
<code>r(futbnd)</code>	pocock, obfleming, wtsiatis, errpocock, errobfleming, kdemets, or hsdecani
<code>r(method)</code>	logrank
<code>r(survvar)</code>	name of the variable containing survival probabilities (if <code>st1()</code> specified)
<code>r(test)</code>	Freedman or Schoenfeld
<code>r(timevar)</code>	name of the variable containing time points (if <code>st1()</code> specified)

Matrices

<code>r(aspent)</code>	cumulative alpha spent per look (stored with efficacy-only stopping or when futility bounds are binding)
<code>r(aspent_fstop)</code>	cumulative alpha spent per look if futility stopping does occur (stored when futility bounds are nonbinding)
<code>r(aspent_nofstop)</code>	cumulative alpha spent per look if futility stopping does not occur (stored when futility bounds are nonbinding)
<code>r(bounds)</code>	stopping boundaries
<code>r(bspent)</code>	cumulative beta spent per look (when futility bounds are specified)
<code>r(bspent_a)</code>	attained cumulative beta spent per look (when futility bounds are specified)
<code>r(design)</code>	sample size and stopping boundaries at interim looks
<code>r(info_frac)</code>	specified information fraction
<code>r(info_frac_a)</code>	fraction of attained information
<code>r(info_level)</code>	specified information level
<code>r(p_crit)</code>	p -values corresponding to boundary critical values
<code>r(sampsize)</code>	sample size at interim looks
<code>r(simpmat)</code>	control-group survival probabilities (if <code>simpson()</code> is specified)

Methods and formulas

Sample sizes at interim analyses are calculated as the product of the [information fraction](#), the [information ratio](#), and the sample size of a fixed-sample study.

See [Methods and formulas](#) in [\[ADAPT\] gsbounds](#) for the formulas used to calculate the stopping boundaries, information fraction, and information ratio. See [Methods and formulas](#) in [\[PSS-2\] power logrank](#) for the formulas used to calculate the sample size for a fixed study. See [Methods and formulas](#) in [\[ADAPT\] gsdesign](#) for the formulas used to calculate the expected sample size.

References

- Freedman, L. S. 1982. Tables of the number of patients required in clinical trials using the logrank test. *Statistics in Medicine* 1: 121–129. <https://doi.org/10.1002/sim.4780010204>.
- Hwang, I. K., W. J. Shih, and J. S. de Cani. 1990. Group sequential designs using a family of type I error probability spending functions. *Statistics in Medicine* 9: 1439–1445. <https://doi.org/10.1002/sim.4780091207>.
- Kim, K., and D. L. DeMets. 1987. Design and analysis of group sequential tests based on the type I error spending rate function. *Biometrika* 74: 149–154. <https://doi.org/10.1093/biomet/74.1.149>.
- Kudo, M., R. S. Finn, S. Qin, K.-H. Han, K. Ikeda, F. Piscaglia, A. Baron, J.-W. Park, G. Han, J. Jassem, J. F. Blanc, A. Vogel, D. Komov, T. R. J. Evans, C. Lopez, C. Dutcus, M. Guo, K. Saito, S. Kraljevic, T. Tamai, M. Ren, and A.-L. Cheng. 2018. Lenvatinib versus sorafenib in first-line treatment of patients with unresectable hepatocellular carcinoma: A randomised phase 3 non-inferiority trial. *Lancet* 391: 1163–1173. [https://doi.org/10.1016/s0140-6736\(18\)30207-1](https://doi.org/10.1016/s0140-6736(18)30207-1).
- Lan, K. K. G., and D. L. DeMets. 1983. Discrete sequential boundaries for clinical trials. *Biometrika* 70: 659–663. <https://doi.org/10.1093/biomet/70.3.659>.
- O’Brien, P. C., and T. R. Fleming. 1979. A multiple testing procedure for clinical trials. *Biometrics* 35: 549–556. <https://doi.org/10.2307/2530245>.
- Peng, Z., W. Fan, B. Zhu, G. Wang, J. Sun, C. Xiao, F. Huang, R. Tang, Y. Cheng, Z. Huang, Y. Liang, H. Fan, L. Qiao, F. Li, W. Zhuang, B. Peng, J. Wang, J. Li, and M. Kuang. 2023. Lenvatinib combined with transarterial chemoembolization as first-line treatment for advanced hepatocellular carcinoma: A phase III, randomized clinical trial (LAUNCH). *Journal of Clinical Oncology* 41: 117–127. <https://doi.org/10.1200/JCO.22.00392>.
- Pitblado, J. S., B. P. Poi, and W. W. Gould. 2024. *Maximum Likelihood Estimation with Stata*. 5th ed. College Station, TX: Stata Press.
- Pocock, S. J. 1977. Group sequential methods in the design and analysis of clinical trials. *Biometrika* 64: 191–199. <https://doi.org/10.1093/biomet/64.2.191>.

- Schoenfeld, D. A. 1981. The asymptotic properties of nonparametric tests for comparing survival distributions. *Biometrika* 68: 316–319. <https://doi.org/10.2307/2335833>.
- . 1983. Sample-size formula for the proportional-hazards regression model. *Biometrics* 39: 499–503. <https://doi.org/10.2307/2531021>.
- Wang, S. K., and A. A. Tsiatis. 1987. Approximately optimal one-parameter boundaries for group sequential trials. *Biometrics* 43: 193–199. <https://doi.org/10.2307/2531959>.

Also see

- [ADAPT] **GSD intro** — Introduction to group sequential designs
- [ADAPT] **gs** — Introduction to commands for group sequential design
- [ADAPT] **gsbounds** — Boundaries for group sequential trials
- [ADAPT] **gsdesign** — Study design for group sequential trials
- [ADAPT] **Glossary**
- [PSS-2] **power logrank** — Power analysis for the log-rank test
- [ST] **sts test** — Test equality of survivor functions

Stata, Stata Press, Mata, NetCourse, and NetCourseNow are registered trademarks of StataCorp LLC. Stata and Stata Press are registered trademarks with the World Intellectual Property Organization of the United Nations. StataNow is a trademark of StataCorp LLC. Other brand and product names are registered trademarks or trademarks of their respective companies. Copyright © 1985–2025 StataCorp LLC, College Station, TX, USA. All rights reserved.



For suggested citations, see the FAQ on [citing Stata documentation](#).