gsdesign onemean — Group sequential design for a one-sample mean test

Description

gsdesign onemean computes stopping boundaries and sample sizes for interim analyses of clinical trials using a one-sample mean test with a group sequential design (GSD). 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 test of one mean, see [PSS-2] power onemean.

Quick start

Sample size and stopping boundaries for a two-sided test of \( H_0 : \mu = 10 \) versus \( H_a : \mu \neq 10 \), with default power of 0.8 to detect the difference between the mean under the null hypothesis, \( m_0 = 10 \), and an observed mean of \( m_a = 15 \), with standard deviation of 12 and at default familywise significance level \( \alpha = 0.05 \), using default group sequential specifications of O’Brien–Fleming efficacy boundaries with two analyses (one interim, one final)

\[
gsdesign \text{ onemean} 10 \ 15, \ sd(12)
\]

Same as above, but specified as \( m_0 \) and difference \( m_a - m_0 = 5 \)

\[
gsdesign \text{ onemean} 10, \ diff(5) \ sd(12)
\]

Same as above, but for a one-sided test with familywise significance level \( \alpha = 0.025 \), power of 0.9, and Pocock bounds with three analyses

\[
gsdesign \text{ onemean} 10, \ diff(5) \ sd(12) \ alpha(0.025) \ power(0.9) \ onesided ///
\text{efficacy(pocock)} \ nlooks(3)
\]

Same as above, but use an error-spending O’Brien–Fleming-style efficacy bound and a nonbinding error-spending Pocock-style futility bound

\[
gsdesign \text{ onemean} 10, \ diff(5) \ sd(12) \ alpha(0.025) \ power(0.9) \ onesided ///
\text{efficacy(errobfleming)} \ futility(errpocock) \ nlooks(3)
\]

Same as above, but treat the standard deviation as known, use a nonbinding Hwang–Shih–de Cani futility bound with parameter \( \gamma_f = -1 \), time the looks to occur with 50%, 75%, and 100% of the data, and graph the bounds

\[
gsdesign \text{ onemean} 10, \ diff(5) \ sd(12) \ knownsd \ alpha(0.025) \ power(0.9) ///
\text{onesided efficacy(errobfleming)} \ futility(hsdecani(-1)) ///
\text{information(50 75 100) graphbounds}
\]

Same as above, but use a Kim–DeMets efficacy bound with parameter \( \rho_e = 2.5 \), label the x axis of the graph with information instead of sample size, and do not plot the critical values from a fixed study design

\[
gsdesign \text{ onemean} 10, \ diff(5) \ sd(12) \ knownsd \ alpha(0.025) \ power(0.9) ///
\text{onesided efficacy(kdemets(2.5))} \ futility(hsdecani(-1)) ///
\text{information(50 75 100) graphbounds(xdiminformation nofixed)}
\]
Menu

Statistics > Power, precision, and sample size

Syntax

\texttt{gsdesign onemean }m_0\ m_a\ [\ ,\ \texttt{onemeanopts boundopts}]

where \(m_0\) is the value of the mean under the null hypothesis and \(m_a\) is the value of the mean under the alternative hypothesis.

\begin{tabular}{ll}
\textbf{onemeanopts} & \textbf{Description} \\
\hline
Main & \\
\texttt{alpha(#)} & overall significance level for all tests; default is \texttt{alpha(0.05)} \\
\texttt{power(#)} & overall power for all tests; default is \texttt{power(0.8)} \\
\texttt{beta(#)} & overall probability of type II error for all tests; default is \texttt{beta(0.2)} \\
\texttt{onesided} & request a one-sided test; default is two-sided \\
\texttt{nfractional} & report fractional sample size \\
\texttt{diff(#)} & difference between the alternative mean and the null mean, \(m_a - m_0\); specify instead of the alternative mean \(m_a\) \\
\texttt{sd(#)} & standard deviation; default is \texttt{sd(1)} \\
\texttt{knownsd} & request computation assuming known standard deviation; default is to assume an unknown standard deviation \\
\texttt{force} & allow calculation with unsupported \texttt{power onemean} options \\
\texttt{poweriteration(powiteropts)} & iteration options for the calculation of fixed-study sample size; seldom used \\
\texttt{collect} & is allowed; see [U] 11.1.10 Prefix commands. \\
force and \texttt{powiteration()} do not appear in the dialog box.
\end{tabular}

\begin{tabular}{ll}
\textbf{powiteropts} & \textbf{Description} \\
\hline
\texttt{init(#)} & initial value for fixed-study sample size \\
\texttt{iterate(#)} & maximum number of iterations; default is \texttt{iterate(500)} \\
\texttt{tolerance(#)} & parameter tolerance; default is \texttt{tolerance(1e-12)} \\
\texttt{ftolerance(#)} & function tolerance; default is \texttt{ftolerance(1e-12)}
\end{tabular}
### bounds

<table>
<thead>
<tr>
<th>Description</th>
<th>Syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>boundary for efficacy stopping; if neither efficacy() nor futility() is specified, the default is efficacy(obfleming)</td>
<td>efficacy(boundary)</td>
</tr>
<tr>
<td>boundary for futility stopping; use binding to request binding futility bounds (default is nonbinding)</td>
<td>futility(boundary[, binding])</td>
</tr>
<tr>
<td>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)</td>
<td>nlooks(#[, equal])</td>
</tr>
<tr>
<td>sequence of information levels for analyses; default is evenly spaced</td>
<td>information(numlist)</td>
</tr>
<tr>
<td>suppress p-values</td>
<td>nopvalues</td>
</tr>
</tbody>
</table>

### Graph

<table>
<thead>
<tr>
<th>Description</th>
<th>Syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>graph boundaries</td>
<td>graphbounds[(graphopts)]</td>
</tr>
<tr>
<td>control the display of boundaries and sample size; seldom used</td>
<td>matlistopts(general_options)</td>
</tr>
<tr>
<td>optimization options for boundary calculations; seldom used</td>
<td>optimopts</td>
</tr>
</tbody>
</table>

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

### boundary

<table>
<thead>
<tr>
<th>Description</th>
<th>Syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>classical O’Brien–Fleming bound</td>
<td>obfleming</td>
</tr>
<tr>
<td>classical Pocock bound</td>
<td>pocock</td>
</tr>
<tr>
<td>classical Wang–Tsiatis bound with specified parameter value</td>
<td>wttsiatis(#)</td>
</tr>
<tr>
<td>error-spending Pocock-style bound</td>
<td>errpocock</td>
</tr>
<tr>
<td>error-spending O’Brien–Fleming-style bound</td>
<td>errobfleming</td>
</tr>
<tr>
<td>error-spending Kim–DeMets bound with specified parameter value</td>
<td>kdemets(#)</td>
</tr>
<tr>
<td>error-spending Hwang–Shih–de Cani bound with specified parameter value</td>
<td>hsdecani(#))</td>
</tr>
</tbody>
</table>
### Graphopts

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

### Optimopts

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

diff(#) specifies the difference between the alternative mean and the null mean, \( m_a - m_0 \). You can either specify the alternative mean \( m_a \) as a command argument or specify the difference between the two means in diff(). If you specify diff(#), the alternative mean is computed as \( m_a = m_0 + #. \)

sd(#) specifies the sample standard deviation or the population standard deviation. The default is sd(1). By default, sd() specifies the sample standard deviation. If knownsd is specified, sd() specifies the population standard deviation.

knownsd requests that the standard deviation be treated as known in the computation. By default, the standard deviation is treated as unknown and the computation is based on a \( t \) test, which uses a Student’s \( t \) distribution as a sampling distribution of the test statistic. If knownsd is specified, the computation is based on a \( z \) test, which uses a normal distribution as the sampling distribution of the test statistic. In either case, critical values for efficacy and futility boundaries calculated by gsdesign onemean are reported on the standardized \( z \) scale. When a \( t \) test is performed, you can use the significance level approach and compare the \( p \)-value from the \( t \) test to the \( p \)-value boundaries reported by gsdesign onemean, as demonstrated in example 2.

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 onemean` but are not shown in the dialog box:

- `force` indicates that `gsdesign onemean` should allow unsupported `power onemean` options, such as options specifying a finite population correction or a cluster randomized design. Even with option `force`, the `power onemean` 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.

`poweriteration(powiteropts)` controls the iterative algorithm used to calculate the fixed-study sample size. This is seldom used.

`powiteropts` are the following:

- `init(#)` specifies an initial value for the sample size when iteration is used to compute the fixed-study sample size. The default is to use a closed-form normal approximation to compute an initial sample size.

- `iterate(#)` specifies the maximum number of iterations for the Newton method during calculation of the fixed-study sample size. The default is `iterate(500)`.

- `tolerance(#)` specifies the tolerance used to determine whether successive parameter estimates have converged when calculating the fixed-study sample size. The default is `tolerance(1e-12)`. See `Convergence criteria` in [M-5] `solve` for details.

- `ftolerance(#)` specifies the tolerance used when calculating the fixed-study sample size to determine whether the proposed solution of a nonlinear equation is sufficiently close to 0 based on the squared Euclidean distance. The default is `ftolerance(1e-12)`. See `Convergence criteria` in [M-5] `solve` for details.

`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.

`optimopts` 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–Tsiatis 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.


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 Gould, Pitblado, and Poi (2010).

- 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. shwonrtolerance 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 $gH^{-1}g' < nrtolerance()$. The default is `nrtolerance(1e-16)`.

`nonrtolerance` 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 `wtsiatis(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 `wtsiatis(0.5)`.

`wtsiatis(#)` 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 $\Delta$ 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 $\rho$ 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 $\gamma$ 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 onemean
- Background for examples
- Computing sample size and stopping boundaries
- Unknown standard deviation and hypothesis tests on mean
- Stopping for both efficacy and futility

This entry describes the use of the gsdesign onemean command for designing a group sequential
analysis for a one-sample mean test. See [ADAPT] GSD intro for a general introduction to GSDs
for clinical trials; see [ADAPT] gsbounds 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 onemean for details about study design for a one-sample mean test.

Introduction

The gold standard for clinical trials is the randomized controlled trial, a clinical trial where
participants are randomly assigned to one of two groups: one group receives the experimental
treatment while the other group is kept as a control. If no existing treatments are comparable with
the experimental treatment, the control group will typically receive a placebo. When a standard of
care exists, there is often an ethical argument against using a placebo. In this case, an active control
is used, in which control-group participants receive the existing standard of care.

Sometimes, however, it is desirable to omit the control group entirely and perform a single-arm
trial. This could occur with trials not intended for regulatory submission (such as early-phase trials),
trials where a placebo would be unethical and there is no existing comparable treatment, or clinical
trials where the population of interest is small and recruiting sufficient participants to create both
treatment and control groups would be difficult.

If the outcome of interest, or endpoint, of a randomized controlled trial is continuous, researchers
will commonly want to test whether the mean of the treatment group is equal to the mean of the
control group. If there is no control group, the observed mean of the treatment group, \( \mu \), is compared
with reference value \( \mu_0 \), which is the value under the null hypothesis.

Here we consider the null hypothesis \( H_0 : \mu = \mu_0 \) versus the two-sided alternative hypothesis
\( H_a : \mu \neq \mu_0 \), the upper one-sided alternative \( H_a : \mu > \mu_0 \), or the lower one-sided alternative
\( H_a : \mu < \mu_0 \). The actual test conducted will depend on whether the standard deviation of the
sampling process is known a priori. In the case of a known standard deviation, the test statistic
follows a standard normal distribution under the null hypothesis, and the corresponding test is known
as a \( z \) test. In the case of an unknown standard deviation, an estimate of the population standard
deviation is used to form a test statistic that follows Student’s $t$ distribution under the null hypothesis, and the corresponding test is a $t$ test. The estimate of the population standard deviation improves with increasing sample size, and the distribution of the test statistic approaches a normal distribution, enabling the use of a $z$ test with large samples, even with unknown standard deviation.

Historically, clinical trials were not analyzed until after the trial was completed and all data were collected. This enabled the use of traditional statistical methods such as those designed for agricultural field experiments, where all data are typically collected over a short period during the harvest. However, data from clinical trials tend to be collected over time, providing incentive to analyze the incomplete trial data while the study is still underway. Several methods have been developed to enable interim analysis of clinical trial data while controlling the type I error rate, and group sequential methods are among the most popular designs.

A GSD plans for a sequence of looks, or analyses of the clinical trial data. If the result at an interim look is sufficiently unambiguous, the trial can be stopped early. GSDs with efficacy bounds allow the trial to be terminated early for treatment efficacy if the interim test statistic crosses the efficacy bound, and designs with futility bounds allow early termination for futility if the interim test statistic crosses the futility bound.

The required sample size estimated by `gsdesign onemean` will depend on whether the standard deviation is known, but the stopping boundaries will not; they are reported on a standardized $z$ scale. The critical values from the boundaries may be compared directly with the $z$ statistic from a $z$ test but must be transformed before being compared with the $t$ statistic from a $t$ test. This is demonstrated in example 2.

### Using `gsdesign onemean`

`gsdesign onemean` calculates sample size and stopping boundaries for a group sequential trial comparing the population mean of one group against a prespecified reference value. `gsdesign onemean` can be thought of as a combination of `power onemean` for sample-size calculations and `gsbounds` for stopping boundary calculations.

To compute sample size, you must specify the effect size in one of two ways: by specifying $m_0$ and $m_a$, the means under the null and alternative hypotheses, respectively, or by specifying the difference $m_a - m_0$ in the `diff()` option. There is no default value for `diff()`, so either $m_0$ and $m_a$ or `diff()` must be included as part of the command specification. $m_0$ may be specified with `diff()`; if $m_0$ is omitted, it is assumed to be 0. Another aspect of the effect size is the standard deviation of the response. This is specified with the `sd()` option, and the default standard deviation is 1. By default, the true population standard deviation is assumed to be unknown, and sample-size calculations are based on a $t$ test; if the population standard deviation is known, the `knownsd` option requests sample-size computations based on a $z$ test.

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 sign of 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 size is rounded up to a whole number at each look, but option `nfractional` can be used to report fractional sample sizes. If `nlooks()` is specified, the default behavior is to divide information evenly among looks 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 at each look.

**Background for examples**

Oncology is an area where single-arm trials have become increasingly popular, and single-arm oncology trials have even led to regulatory approval (Tenhunen et al. 2020). There are multiple reasons for this trend, but Simon et al. (2015) identify several common factors of single-arm oncology trials that have been approved by regulators, including “unprecedented effect on tumor response is observed in a setting of high unmet medical need” as well as the existence of a well-defined target population.

Salvage therapy, also known as rescue therapy, is not a specific cancer treatment but rather a term to describe treatments that are considered for a patient when all standard treatment protocols have failed. We consider the design of a clinical trial of sunitinib malate as a salvage therapy for advanced unresectable non–small cell lung cancer, where “unresectable” describes tumors that cannot be removed surgically. Sunitinib is a tyrosine kinase inhibitor that has been used in the treatment of renal cancer and gastrointestinal stromal tumors, and it has been considered more recently as a treatment for other cancers. Novello et al. (2009) report the results of a phase 2 clinical trial of sunitinib for participants with advanced non–small cell lung cancer that did not respond to platinum-based chemotherapy, the standard of care.

The trial’s endpoint, or outcome of interest, is the tumor shrinkage rate (TSR) of the participant’s largest tumor, defined as

\[
    \text{TSR} = \frac{(D_b - D_a)}{(D_b \times t)} \times 100\%\,
\]

where \(D_b\) is the longest diameter of the tumor before treatment, \(D_a\) is the longest diameter after treatment, and \(t\) is the time elapsed in days. This endpoint was chosen based on the results of Yu et al. (2019), who identified TSR as a useful predictor of long-term outcomes in participants with advanced unresectable non–small cell lung cancer who received treatment with tyrosine kinase inhibitors.

By focusing on participants with unresectable tumors that have not responded to existing treatments, we have clearly defined a target population with high unmet medical need. There can be no active control because all standard treatment protocols have failed, and the only ethical justification for a placebo control would be if it were necessary to draw valid conclusions about the effect of sunitinib on TSR. As it so happens, non–small cell lung cancer is a well-studied disease, and it is known that spontaneous tumor shrinkage in patients with advanced-stage disease is exceedingly rare (Shatola et al. 2020).
Computing sample size and stopping boundaries

Suppose that we wish to conduct a single-arm trial, where participants will undergo a CT scan to measure tumor diameter both before and after a four-week course of sunitinib. We will test the null hypothesis \( H_0: \mu = \mu_0 \) versus the upper one-sided alternative \( H_a: \mu > \mu_0 \), where \( \mu \) is the mean TSR observed in study participants and \( \mu_0 = 0 \), indicating no tumor shrinkage under the null hypothesis. We choose a one-sided test because tumor shrinkage is of interest, not tumor growth. If we were interested in testing whether sunitinib slowed the rate of tumor growth, we would need a control group to compare against.

Example 1: Efficacy bounds for a one-sample mean test

Yu et al. (2019) observed that participants with \( \text{TSR} \geq 0.49\% \) had a nearly 50% increase in overall survival time and in progression-free survival time when compared with participants with \( \text{TSR} < 0.49\% \), so we desire 80% power to detect \( \mu_a = 0.49 \). We will conduct a one-sided trial with a familywise significance level of 2.5%, and we will assume that TSR has a known standard deviation of 1.1%. Assuming a known standard deviation is often unrealistic, and this assumption is relaxed in example 2. The stopping rule we employ is a Pocock efficacy bound, with four looks at the data: three interim analyses and one final analysis. We graph the bounds for visual inspection.

```
gsdesign onemean 0 0.49, sd(1.1) knownsd alpha(0.025) efficacy(pocock) > nlooks(4) onesided graphbounds
```

Group sequential design for a one-sample mean test

\[ z \text{ test } H_0: m = m_0 \text{ versus } H_a: m > m_0 \]

Efficacy: Pocock

Study parameters:

- \( \alpha = 0.0250 \) (upper one-sided)
- \( \text{power} = 0.8000 \)
- \( \delta = 0.4455 \)
- \( m_0 = 0.0000 \)
- \( m_a = 0.4900 \)
- \( \text{sd} = 1.1000 \)

Expected sample size:

- \( H_0 = 47.45 \)
- \( H_a = 32.02 \)

Info. ratio = 1.2025

N fixed = 40

N max = 48

Fixed-study crit. value = 1.9600

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

<table>
<thead>
<tr>
<th>Look</th>
<th>Info. frac.</th>
<th>Efficacy Upper p-value</th>
<th>Sample size N</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.25</td>
<td>2.3613</td>
<td>0.0091</td>
</tr>
<tr>
<td>2</td>
<td>0.50</td>
<td>2.3613</td>
<td>0.0091</td>
</tr>
<tr>
<td>3</td>
<td>0.75</td>
<td>2.3613</td>
<td>0.0091</td>
</tr>
<tr>
<td>4</td>
<td>1.00</td>
<td>2.3613</td>
<td>0.0091</td>
</tr>
</tbody>
</table>

Note: Critical values are for \( z \) statistics; otherwise, use p-value boundaries.
gsdesign onemean — Group sequential design for a one-sample mean test

Figure 1. Sample size for a test of one mean with one-sided Pocock efficacy bounds

Parameter: \( \alpha = 0.025 \) (one-sided), \( 1-\beta = 0.8 \), \( \delta = 0.45 \), \( \mu_0 = 0 \), \( \sigma = 1.1 \)

Next we see the information ratio, the sample size for a fixed study with an equivalent significance level and power \( (N_{\text{fixed}}) \), and the maximum sample size of the GSD \( (N_{\text{max}}) \). The information ratio is the ratio of the maximum sample size of the GSD to the fixed-study sample size. We then see the critical value for a fixed study with an equivalent significance level.

At the end of the display is a table of stopping boundaries, \( p \)-values, and sample sizes for the four looks. Because we indicated that the population standard deviation was known, the appropriate analysis is a \( z \) test, which yields a \( z \) statistic. The first analysis, look 1, will be conducted once data have been collected from 12 participants. The second and third looks take place with 24 and 36 observations, respectively, and the final analysis occurs once data from all 48 participants have been collected. A defining feature of Pocock efficacy bounds is that all looks use the same critical value, which in this case is 2.361. If the test statistic at the first look, which we will label \( z_1 \), is greater than or equal to 2.361, then it lies in the rejection region, shaded blue on the graph. In this case, \( H_0 \) will be rejected and the study will terminate early due to treatment efficacy. If \( z_1 \) is less than 2.361, then it is said to lie in the continuation region and the study will proceed to the next look. The same procedure is repeated at the second and third looks. At the final look, there is no continuation region, only rejection and acceptance regions. \( H_0 \) is rejected if \( z_4 \geq 2.361 \), and \( H_0 \) is accepted if \( z_4 < 2.361 \). In addition to displaying the bounds visually, the graph also marks the critical value from a fixed study with equivalent power and significance level.
Example 2: Unknown standard deviation, specifying difference between means

Instead of specifying the alternative mean of 0.49 as in example 1, we can specify the difference between the mean TSRs under the null and alternative hypotheses in the `diff()` option. Additionally, we will omit option `knownsd` because we do not know the population standard deviation a priori. This yields sample sizes for a $t$ test, which will be demonstrated. The overall shape of the bounds is unchanged from figure 1, so we omit the graph.

```
. gsdesign onemean, diff(0.49) sd(1.1) alpha(0.025) efficacy(pocock)
> nlooks(4) onesided
```

Group sequential design for a one-sample mean test
$t$ test
$H_0$: $m = m_0$ versus $Ha$: $m > m_0$

Efficacy: Pocock

Study parameters:
- $alpha = 0.0250$ (upper one-sided)
- $power = 0.8000$
- $delta = 0.4455$
- $m_0 = 0.0000$
- $ma = 0.4900$
- $diff = 0.4900$
- $sd = 1.1000$

Expected sample size:
- $H_0 = 49.44$
- $Ha = 33.48$

Info. ratio = 1.2025

<table>
<thead>
<tr>
<th>Look</th>
<th>Info. frac.</th>
<th>Efficacy p-value</th>
<th>Sample size</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.25</td>
<td>2.3613</td>
<td>0.0091</td>
</tr>
<tr>
<td>2</td>
<td>0.50</td>
<td>2.3613</td>
<td>0.0091</td>
</tr>
<tr>
<td>3</td>
<td>0.75</td>
<td>2.3613</td>
<td>0.0091</td>
</tr>
<tr>
<td>4</td>
<td>1.00</td>
<td>2.3613</td>
<td>0.0091</td>
</tr>
</tbody>
</table>

Notes: Critical values are for $z$ statistics; otherwise, use p-value boundaries.
Requested information fraction not attained.

This alternate method of specifying the difference in means has not changed the fixed-study parameters reported by `gsdesign`, nor has it changed the critical values of the efficacy boundaries. What has changed is the required sample size to achieve 80% power with a fixed study, which has increased from 40 participants in example 1 to 42 participants now that we do not assume a known standard deviation. The information ratio is the same, but the larger fixed-study sample size yields larger required samples in the GSD as well, with a maximum sample of 50 participants and correspondingly larger expected sample sizes under the null and alternative hypotheses.

The biggest change, however, is the process of testing our hypothesis at the interim analyses. Previously, a $z$ test was performed and the resulting $z$ statistic was compared directly with the critical
values of the efficacy bounds. Now that the standard deviation is estimated, we use a $t$ test, which yields $t$ statistics. These $t$ statistics should not be compared directly with the standardized $z$ critical values, as the note under the table indicates. Instead, the $p$-value from the test can be compared with the boundaries by using the significance level approach described in [ADAPT] gsbounds.

Imagine that the first look is conducted; the mean TSR of the first 13 participants is 0.9 and the sample standard deviation is 1.3. We conduct a $t$ test using ttesti, the immediate form of the [R] ttest command. We specify ttesti 13 0.9 1.3 0, with the sample size as the first number, the observed mean as the second, the sample standard deviation as the third, and the mean under $H_0$ as the fourth.

```
. ttesti 13 0.9 1.3 0
One-sample t test

<table>
<thead>
<tr>
<th></th>
<th>Obs</th>
<th>Mean</th>
<th>Std. err.</th>
<th>Std. dev.</th>
<th>[95% conf. interval]</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>13</td>
<td>.9</td>
<td>.3605551</td>
<td>1.3</td>
<td>.1144179 1.685582</td>
</tr>
</tbody>
</table>

mean = mean(x) $t$ = 2.4962
H0: mean = 0 Degrees of freedom = 12
Ha: mean < 0 Ha: mean != 0 Ha: mean > 0
Pr(T < $t$) = 0.9859 Pr(|T| > |$t$|) = 0.0281 Pr(T > $t$) = 0.0141
```

The output displays a $t$ statistic of 2.496 with 12 degrees of freedom. ttesti also displays the $p$-value of the upper one-sided test: $\Pr(T > t) = 0.0141$. This value, $p_1$, can be compared directly with the $p$-value of the efficacy bound. Because $p_1 > 0.0091$, we cannot reject $H_0$ at this look.

As the sample size increases and the degrees of freedom of the $t$ test increase, the $t$ distribution approaches a normal distribution. In practice, it is common to see a $z$ test used with large samples, even when the population standard deviation is unknown and is estimated from the sample. But with small samples, the significance level approach is best.

### Stopping for both efficacy and futility

> Example 3: Stopping for both efficacy and futility, graphing bounds

The previous two examples have used stopping rules allowing for efficacy stopping, but further efficiency gains can be realized if the trial is also allowed to stop early due to treatment futility. This is done by creating a framework for accepting $H_0$, a practice that is unheard-of in some areas of statistical analysis but commonly discussed in the literature about GSDs.

Continuing with the scenario of example 2, we add a Wang–Tsiatis futility bound with parameter $\Delta_f = 0.3$. The boundary is nonbinding, the default for futility bounds, meaning that if the trial crosses the futility bound, stopping is optional. If the trial continues despite an interim result below the futility bound, the familywise type I error is still controlled at the desired significance level of 2.5%.
Group sequential design for a one-sample mean test

H0: m = m0 versus Ha: m > m0

Efficacy: Pocock
Futility: Wang–Tsiatis, nonbinding, Delta = 0.3000

Study parameters:
- alpha = 0.0250 (upper one-sided)
- power = 0.8000
- delta = 0.4455
- m0 = 0.0000
- ma = 0.4900
- diff = 0.4900
- sd = 1.1000

Expected sample size:
- H0 = 23.96
- Ha = 37.78

Info. ratio = 1.5281
- N fixed = 42
- N max = 64

Fixed-study crit. value = 1.9600

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

<table>
<thead>
<tr>
<th>Look</th>
<th>Info. frac.</th>
<th>Efficacy Upper p-value</th>
<th>Futility Lower p-value</th>
<th>Sample size N</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.25</td>
<td>2.3613</td>
<td>0.0091</td>
<td>0.2776</td>
</tr>
<tr>
<td>2</td>
<td>0.50</td>
<td>2.3613</td>
<td>0.0091</td>
<td>1.1831</td>
</tr>
<tr>
<td>3</td>
<td>0.75</td>
<td>2.3613</td>
<td>0.0091</td>
<td>1.8321</td>
</tr>
<tr>
<td>4</td>
<td>1.00</td>
<td>2.3613</td>
<td>0.0091</td>
<td>2.3613</td>
</tr>
</tbody>
</table>

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

Figure 2. Sample size for a test of one mean with one-sided efficacy and futility bounds
The fixed-study properties are unchanged from example 2, but the information ratio is increased, reflecting the larger maximum sample size of the new GSD. However, the expected sample size under $H_0$ has decreased dramatically due to the ability to terminate the trial early for futility.

As before, the Pocock efficacy bounds use the same critical $z$-value of 2.361 for all looks, which corresponds to a $p$-value of 0.0091. Efficacy critical values are unaffected by the addition of nonbinding futility bounds, but as is demonstrated in example 2 of [ADAPT] gsdesign, using binding futility bounds would impact the critical values of efficacy bounds.

As in the previous example, we will conduct $t$ tests at each look, and we will compare the $p$-values from each $t$ test with the corresponding efficacy and futility critical $p$-values. The graph, however, uses the standardized $z$ scale on the vertical axis to facilitate comparison with graphs of other stopping boundaries.

The first test is conducted after 16 observations have been collected, and we will let $p_1$ denote the $p$-value from the first look. If $p_1 > 0.391$, then the test statistic lies in the acceptance region, and researchers have the option to stop the trial early for futility. If $0.391 \geq p_1 > 0.0091$, then the test statistic falls within the continuation region, and the trial must proceed to the next look. If $p_1 \leq 0.0091$, then it is inside the rejection region and $H_0$ is rejected, terminating the trial early due to treatment efficacy. If the trial does not stop at the first look, a similar procedure is performed at the second look once 32 observations have been recorded. This time, the futility boundary $p$-value has decreased to 0.118, shrinking the continuation region and expanding the acceptance region. If the trial does not stop at the second look, the procedure is repeated at the third look with 48 observations, where the futility $p$-value has decreased to 0.034. If the trial proceeds to the final analysis with 64 observations, there is no continuation region, because the futility bound critical value is identical to the efficacy bound $p$-value of 0.0091. If $p_4 \leq 0.0091$, then $H_0$ is rejected; otherwise, $H_0$ is accepted.
## Stored results

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

### Scalars

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>r(alpha)</code></td>
<td>overall significance level (familywise type I error)</td>
</tr>
<tr>
<td><code>r(beta)</code></td>
<td>overall probability of a type II error</td>
</tr>
<tr>
<td><code>r(binding)</code></td>
<td>1 for binding futility bounds, 0 for nonbinding</td>
</tr>
<tr>
<td><code>r(delta)</code></td>
<td>effect size</td>
</tr>
<tr>
<td><code>r(diff)</code></td>
<td>difference between the alternative and null means</td>
</tr>
<tr>
<td><code>r(effparam)</code></td>
<td>efficacy parameter (if <code>wtsiatis()</code>, <code>kdemets()</code>, or <code>hsdecani()</code> specified)</td>
</tr>
<tr>
<td><code>r(ESS0)</code></td>
<td>expected sample size under null hypothesis</td>
</tr>
<tr>
<td><code>r(ESS1)</code></td>
<td>expected sample size under alternative hypothesis</td>
</tr>
<tr>
<td><code>r(futparam)</code></td>
<td>futility parameter (if <code>wtsiatis()</code>, <code>kdemets()</code>, or <code>hsdecani()</code> specified)</td>
</tr>
<tr>
<td><code>r(info_ratio)</code></td>
<td>ratio of maximum information required to that of a fixed study design</td>
</tr>
<tr>
<td><code>r(knownsd)</code></td>
<td>1 if option <code>knownsd</code> is specified, 0 otherwise</td>
</tr>
<tr>
<td><code>r(m0)</code></td>
<td>mean under the null hypothesis</td>
</tr>
<tr>
<td><code>r(ma)</code></td>
<td>mean under the alternative hypothesis</td>
</tr>
<tr>
<td><code>r(N_fixed)</code></td>
<td>sample size of a fixed study design</td>
</tr>
<tr>
<td><code>r(N_fixedfrac)</code></td>
<td>fractional sample size of a fixed study design</td>
</tr>
<tr>
<td><code>r(N_max)</code></td>
<td>maximum sample size if the study continues to completion</td>
</tr>
<tr>
<td><code>r(nfractional)</code></td>
<td>1 if <code>nfractional</code> is specified, 0 otherwise</td>
</tr>
<tr>
<td><code>r(nlooks)</code></td>
<td>number of analyses</td>
</tr>
<tr>
<td><code>r(onesided)</code></td>
<td>1 for a one-sided test, 0 otherwise</td>
</tr>
<tr>
<td><code>r(pow_converged)</code></td>
<td>1 if power calculation iteration algorithm converged, 0 otherwise</td>
</tr>
<tr>
<td><code>r(pow_deltax)</code></td>
<td>final parameter tolerance achieved for power calculation</td>
</tr>
<tr>
<td><code>r(pow_ftolerance)</code></td>
<td>requested distance of power calculation objective function from 0</td>
</tr>
<tr>
<td><code>r(pow_function)</code></td>
<td>final distance of power calculation objective function from 0</td>
</tr>
<tr>
<td><code>r(pow_maxiter)</code></td>
<td>maximum number of iterations performed for power calculation</td>
</tr>
<tr>
<td><code>r(pow_tolerance)</code></td>
<td>requested parameter tolerance for power calculation</td>
</tr>
<tr>
<td><code>r(power)</code></td>
<td>specified overall power</td>
</tr>
<tr>
<td><code>r(power_a)</code></td>
<td>attained overall power</td>
</tr>
<tr>
<td><code>r(sd)</code></td>
<td>standard deviation</td>
</tr>
<tr>
<td><code>r(stop)</code></td>
<td>0 for futility bounds, 1 for efficacy bounds, 2 for both</td>
</tr>
<tr>
<td><code>r(z_fixed)</code></td>
<td>critical value for an equivalent fixed study design</td>
</tr>
</tbody>
</table>

### Macros

<table>
<thead>
<tr>
<th>Macro</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>r(cmd)</code></td>
<td><code>gsdesign</code></td>
</tr>
<tr>
<td><code>r(cmdline)</code></td>
<td>command as typed</td>
</tr>
<tr>
<td><code>r(direction)</code></td>
<td>upper, lower, or two-sided</td>
</tr>
<tr>
<td><code>r(effbnd)</code></td>
<td><code>pocock</code>, <code>obfleming</code>, <code>wtsiatis</code>, <code>errpocock</code>, <code>errobfleming</code>, <code>kdemets</code>, or <code>hsdecani</code></td>
</tr>
<tr>
<td><code>r(futbnd)</code></td>
<td><code>pocock</code>, <code>obfleming</code>, <code>wtsiatis</code>, <code>errpocock</code>, <code>errobfleming</code>, <code>kdemets</code>, or <code>hsdecani</code></td>
</tr>
<tr>
<td><code>r(method)</code></td>
<td><code>onemean</code></td>
</tr>
</tbody>
</table>

### Matrices

<table>
<thead>
<tr>
<th>Matrix</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>r(aspent)</code></td>
<td>cumulative alpha spent per look (stored with efficacy-only stopping or when futility bounds are binding)</td>
</tr>
<tr>
<td><code>r(aspent_fstop)</code></td>
<td>cumulative alpha spent per look if futility stopping does occur (stored when futility bounds are nonbinding)</td>
</tr>
<tr>
<td><code>r(aspentnofstop)</code></td>
<td>cumulative alpha spent per look if futility stopping does not occur (stored when futility bounds are nonbinding)</td>
</tr>
<tr>
<td><code>r(bounds)</code></td>
<td>stopping boundaries</td>
</tr>
<tr>
<td><code>r(bspent)</code></td>
<td>cumulative beta spent per look (when futility bounds are specified)</td>
</tr>
<tr>
<td><code>r(bspent_a)</code></td>
<td>attained cumulative beta spent per look (when futility bounds are specified)</td>
</tr>
<tr>
<td><code>r(info_frac)</code></td>
<td>specified information fraction</td>
</tr>
<tr>
<td><code>r(info_frac_a)</code></td>
<td>fraction of attained information</td>
</tr>
<tr>
<td><code>r(info_level)</code></td>
<td>specified information level</td>
</tr>
<tr>
<td><code>r(p_crit)</code></td>
<td>p-values corresponding to boundary critical values</td>
</tr>
<tr>
<td><code>r(sampsize)</code></td>
<td>sample size at interim looks</td>
</tr>
</tbody>
</table>
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 onemean for the formulas used to calculate the sample size for a fixed study. See Methods and formulas in [ADAPT] gsd for the formulas used to calculate the expected sample size.

References


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] gsdesign twomeans — Group sequential design for a two-sample means test
[ADAPT] Glossary

[PSS-2] power onemean — Power analysis for a one-sample mean test
[R] ttest — t tests (mean-comparison tests)
[R] ztest — z tests (mean-comparison tests, known variance)