

gsdesign oneproportion — Group sequential design for a one-sample proportion test

Description	Quick start	Menu	Syntax
Options	Remarks and examples	Stored results	Methods and formulas
References	Also see		

Description

`gsdesign oneproportion` computes stopping boundaries and sample sizes for interim analyses of clinical trials using a one-sample proportion 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 proportion, see [PSS-2] [power oneproportion](#).

Quick start

Sample size and stopping boundaries for a two-sided score test of $H_0: \pi = 0.3$ versus $H_a: \pi \neq 0.3$, with default significance level $\alpha = 0.05$ and power of 0.8 to detect the difference between the proportion under the null hypothesis, $p_0 = 0.3$, and an observed proportion of $p_a = 0.4$, using default group sequential specifications of O'Brien–Fleming efficacy boundaries with two analyses (one interim, one final)

```
gsdesign oneproportion 0.3 0.4
```

Same as above, but use Hwang–Shih–de Cani error-spending efficacy bounds with parameter $\gamma = -2$ and four looks

```
gsdesign oneproportion 0.3 0.4, efficacy(hsdecani(-2)) nlooks(4)
```

Same as above, but specified as p_0 and difference $p_a - p_0 = 0.1$

```
gsdesign oneproportion 0.3, diff(0.1) efficacy(hsdecani(-2)) nlooks(4)
```

Same as above, but add nonbinding O'Brien–Fleming-style futility bounds and graph the boundaries

```
gsdesign oneproportion 0.3, diff(0.1) efficacy(hsdecani(-2))          ///
      futility(errob Fleming) nlooks(4) graphbounds
```

Same as above, but time the looks to occur with 40%, 60%, 80%, and 100% of the data

```
gsdesign oneproportion 0.3, diff(0.1) efficacy(hsdecani(-2))          ///
      futility(errob Fleming) information(4 6 8 10) graphbounds
```

Same as above, but remove the efficacy bound and make the futility bound binding

```
gsdesign oneproportion 0.3, diff(0.1) futility(errob Fleming, binding) ///
      information(4 6 8 10) graphbounds
```

Menu

Statistics > Power, precision, and sample size

Syntax

```
gsdesign oneproportion  $p_0$   $p_a$  [ , onepropt boundopts ]
```

where p_0 is the null (hypothesized) proportion or the value of the proportion under the null hypothesis, and p_a is the alternative (target) proportion or the value of the proportion under the alternative hypothesis.

<i>onepropt</i>	Description
Main	
<u>alpha</u> (#)	overall significance level for all tests; default is <code>alpha(0.05)</code>
<u>power</u> (#)	overall power for all tests; default is <code>power(0.8)</code>
<u>beta</u> (#)	overall probability of type II error for all tests; default is <code>beta(0.2)</code>
<u>onesided</u>	request a one-sided test; default is two-sided
<u>nfractional</u>	report fractional sample size
<u>diff</u> (#)	difference between the alternative proportion and the null proportion, $p_a - p_0$; specify instead of the alternative proportion p_a
<u>test</u> (<i>test</i>)	specify the type of test; options are <code>score</code> (the default) and <code>wald</code>
<u>continuity</u>	apply continuity correction to the normal approximation of the discrete distribution
<u>force</u>	allow calculation with unsupported <code>power oneproportion</code> options
<u>poweriteration</u> (<i>powiteropts</i>)	iteration options for the calculation of fixed-study sample size; seldom used

`collect` is allowed; see [U] 11.1.10 Prefix commands.

`force` and `poweriteration()` do not appear in the dialog box.

<i>powiteropts</i>	Description
<u>init</u> (#)	initial value for fixed-study sample size
<u>iterate</u> (#)	maximum number of iterations; default is <code>iterate(500)</code>
<u>tolerance</u> (#)	parameter tolerance; default is <code>tolerance(1e-12)</code>
<u>ftolerance</u> (#)	function tolerance; default is <code>ftolerance(1e-12)</code>

<i>boundopts</i>	Description
Bounds	
<code>efficacy(<i>boundary</i>)</code>	boundary for efficacy stopping; if neither <code>efficacy()</code> nor <code>futility()</code> is specified, the default is <code>efficacy(obfleming)</code>
<code>futility(<i>boundary</i> [, <i>binding</i>])</code>	boundary for futility stopping; use <code>binding</code> to request binding futility bounds (default is nonbinding)
<code>nlooks(# [, <i>equal</i>])</code>	total number of analyses (<code>nlooks()</code> – 1 interim analyses and one final analysis); use <code>equal</code> to enforce equal information increments; if neither <code>nlooks()</code> nor <code>information()</code> is specified, the default is <code>nlooks(2)</code>
<code>information(<i>numlist</i>)</code>	sequence of information levels for analyses; default is evenly spaced
<code>nopvalues</code>	suppress <i>p</i> -values
Graph	
<code>graphbounds[(<i>graphopts</i>)]</code>	graph boundaries
<code>matlistopts(<i>general_options</i>)</code>	control the display of boundaries and sample size; seldom used
<code>optimopts</code>	optimization options for boundary calculations; seldom used
<code>matlistopts()</code> and <code>optimopts</code> do not appear in the dialog box.	

<i>boundary</i>	Description
<code>obfleming</code>	classical O’Brien–Fleming bound
<code>pocock</code>	classical Pocock bound
<code>wtsiatis(#)</code>	classical Wang–Tsiatis bound with specified parameter value
<code>errpocock</code>	error-spending Pocock-style bound
<code>errob Fleming</code>	error-spending O’Brien–Fleming-style bound
<code>kdemets(#)</code>	error-spending Kim–DeMets bound with specified parameter value
<code>hsdecani(#)</code>	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 information levels if <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>
[no]log	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.

`diff(#)` specifies the difference between the alternative proportion and the null proportion, $p_a - p_0$. You can either specify the alternative proportion p_a as a command argument or specify the difference between the two proportions in `diff()`. If you specify `diff(#)`, the alternative proportion is computed as $p_a = p_0 + \#$.

`test(test)` specifies the type of test that will be used for data analysis. Sample-size calculations depend on the test that will be conducted. `test` is either `score` or `wald`.

`score` requests computations for the score test, which uses the value of the null proportion in the formula for the standard error of the estimator of the proportion. This is the default test, and this test can be performed with command `prtest`; see [R] [prtest](#).

`wald` requests computations for the Wald test, which uses the value of the alternative proportion in the formula for the standard error of the estimator of the proportion.

Note that `power oneproportion option test(binomial)` cannot be used for sample-size calculations and is not compatible with `gsdesign oneproportion`. However, option `continuity` implements a continuity correction that yields an estimate of the sample size that would be required by the exact binomial test at the specified significance level and power. The exact binomial test can be performed with command `bitest`; see [R] [bitest](#). When the exact binomial test is performed, you can use the [significance level approach](#) and compare the p -value from the exact test to the p -value boundaries reported by `gsdesign oneproportion`.

`continuity` requests that the continuity correction of [Levin and Chen \(1999\)](#) be applied to the normal approximation of the discrete distribution. This yields an estimate of the sample size that would be required by the exact binomial test at the specified significance level and power.

Bounds

`efficacy(boundary)` specifies the boundary for efficacy stopping. If neither `efficacy()` nor `utility()` 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 oneproportion` but are not shown in the dialog box:

`force` indicates that `gsdesign oneproportion` should allow unsupported `power oneproportion` options, such as options specifying a cluster randomized design. Even with option `force`, the `power oneproportion` 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] `solvenl()` 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] `solvenl()` for details.

`matlistopts(general_options)` affects the display of the matrix of boundaries and sample sizes. `general_options` are `title()`, `tindent()`, `rowtitle()`, `showcoleq()`, `coleqonly`, `color-coleq()`, `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**.

`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 [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(dfpr)` 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 $\mathbf{g}\mathbf{H}^{-1}\mathbf{g}' < \text{nrtolerance}()$. 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 `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 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.

`errob Fleming` 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

[stata.com](https://www.stata.com)

Remarks are presented under the following headings:

[Introduction](#)

[Using gsdesign oneproportion](#)

[Background for examples](#)

[Computing sample size and stopping boundaries](#)

This entry describes the use of the `gsdesign oneproportion` command for designing a group sequential analysis for a one-sample proportion 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 oneproportion](#) for details about study design for a one-sample proportion test.

Introduction

The analysis of proportions is carried out in clinical trials where the response variable, or endpoint, is binary. Each observation is a Bernoulli outcome with a fixed probability p of observing an event of interest in a population. We assume the outcome is observed a fixed number of times and that individual observations are independent with shared probability of success p . As an example, in a clinical trial of a drug for tuberculosis treatment, the endpoint of interest might be culture status after eight weeks. Each observation is the binary indicator of whether tuberculosis was present or absent from the culture taken from one participant.

Sometimes an endpoint that can take several values is discretized into a binary endpoint. For example, the World Health Organization defines a low birthweight as below 2,500 grams. In a clinical trial examining the effect of a prenatal nutrition program on the proportion of newborns with low birthweight, each observation is the binary indicator of whether an infant weighs less than 2,500 grams at birth.

Some clinical trials combine multiple endpoints into a single composite endpoint, which can be binary. A heart failure trial might use a composite endpoint such as “cardiovascular death or heart failure hospitalization”. In this case, each observation is an indicator of whether heart failure caused a participant’s death or hospitalization. The outcome from participants who died due to heart failure, were hospitalized due to heart failure, or died following hospitalization for heart failure would be recorded as 1, while the outcome from participants who neither died nor were hospitalized due to heart failure would be recorded as 0.

The gold standard for clinical trials is the randomized controlled 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. The groups are often called arms, and the experimental arm will receive the experimental treatment. The control arm will receive either a placebo (an inactive substance such as a sugar pill, or a “sham” procedure for nonpharmacological trials) or an active control (typically the standard of care, a treatment that has been previously studied and is known to be effective).

However, there are some scenarios where randomizing subjects to a control group would be impractical or unethical, such as the clinical trial of a treatment for a serious condition where there is a moral argument against giving participants a placebo but there is no existing standard of care. In these cases, a single-arm clinical trial is desired.

`gsdesign oneproportion` calculates sample size and stopping boundaries for a group sequential trial comparing the population proportion of one group against a prespecified reference value. Specifically, we consider the null hypothesis $H_0: p = p_0$ versus the two-sided alternative hypothesis $H_a: p \neq p_0$, the upper one-sided alternative $H_a: p > p_0$, or the lower one-sided alternative $H_a: p < p_0$.

Two common hypothesis tests for a one-sample proportion are the small-sample binomial test and the asymptotic (large-sample) normal test. `gsdesign oneproportion` allows sample-size calculations for tests using the large-sample normal approximation of the sampling distribution of the test statistic. This test, implemented with command `prtest`, yields a z statistic that can be compared with the boundary critical values calculated by `gsdesign oneproportion`. For small samples, the exact binomial test can be performed by using the `bitest` command, but the [significance level approach](#) must be used to compare the p -value from the test statistic to the boundary.

Using `gsdesign oneproportion`

`gsdesign oneproportion` calculates sample size and stopping boundaries for a group sequential trial comparing a population proportion against a hypothesized value. `gsdesign oneproportion` can be thought of as a combination of [power oneproportion](#) for sample-size calculations and [gsbounds](#) for stopping boundary calculations.

To compute sample size, you must specify the effect size. There are two ways to do this: by specifying p_0 and p_a , the proportions under the null and alternative hypotheses, respectively, or by specifying p_0 and the difference $p_a - p_0$ in the `diff()` option. There is no default value for `diff()`, so either p_0 and p_a or p_0 and `diff()` must be included as part of the command specification. By default, sample sizes are calculated assuming that a score test will be conducted. To perform sample-size calculations for a Wald test, specify the `test(wald)` option.

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 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 at each look.

Background for examples

Oncology is an area where single-arm trials are becoming increasingly common, and some have even led to approval by regulators. This trend is studied by [Tenhunen et al. \(2020\)](#), who note that regulatory approval is most common in trials where the response rate is compared with a prespecified threshold for success, and many of the approved single-arm trials are lung cancer trials and trials of late-line treatments.

In [example 1](#), we used `[ADAPT] gsdesign onemean` to calculate sample sizes and bounds for a clinical trial with a continuous endpoint for sunitinib malate as a salvage therapy for lung cancer. Salvage therapy, also known as rescue therapy, is a term for treatments that are considered when all standard treatment protocols have failed because they were ineffective or because they caused the patient intolerable side effects. Here we consider the design of a clinical trial of sunitinib as a salvage therapy for advanced unresectable non–small cell lung cancer, where “unresectable” describes tumors that cannot be removed surgically.

The trial’s outcome of interest is the objective response rate (ORR), defined as the proportion of participants that exhibit at least a partial response to therapy ([Delgado and Guddati 2021](#)). Each participant’s response to therapy can be considered a Bernoulli trial, with a response of 1 indicating clinical improvement and a response of 0 indicating lack of improvement.

[Socinski et al. \(2008\)](#) report the results of a phase 2 clinical trial of sunitinib for participants with advanced non–small cell lung cancer that had progressed despite treatment with the standard of care, a platinum-based chemotherapy regimen. They found an ORR of 11.1%, which might seem small but can be considered a victory because the probability of clinical improvement without treatment is effectively 0% ([Shatola et al. 2020](#)).

In this trial, there can be no **active control** because all standard treatment protocols have failed, and it is not ethical to recruit participants to a placebo control arm knowing that they stand no chance of improvement. Fortunately, a control arm is not necessary to compare the ORR of the experimental arm with a prespecified clinically relevant ORR, so a single-arm trial can be used.

Computing sample size and stopping boundaries

▷ Example 1: Efficacy bounds for a large-sample test of one proportion

We use `gsdesign oneproportion` to calculate efficacy bounds and sample sizes for this situation. The ORR of untreated patients with advanced unresectable non–small cell lung cancer is 0%, and we define a clinically meaningful threshold for success to be an ORR of 5%. Typically, the null hypothesis in a clinical trial is “no treatment effect”, but we are uninterested in clinically irrelevant improvements in ORR, so we modify our null hypothesis to be “no meaningful treatment effect”.

We will test $H_0: p \leq 0.05$ against $H_a: p > 0.05$, which is identical to testing whether sunitinib is [substantially superior](#) to no treatment, with a superiority margin of $\delta = 0.05$ (Chow et al. 2018, chap. 4.1.2). We require 90% power to detect the difference between $p_0 = 0.05$ and $p_a = 0.111$ at a familywise significance level of 2.5%. We plan on conducting a large-sample score test, and we employ group sequential specifications of Pocock efficacy boundaries with two analyses (one interim, one final).

```
. gsdesign oneproportion 0.05 0.111, alpha(0.025) power(0.9) onesided
> efficacy(pocock)
```

```
Group sequential design for a one-sample proportion test
```

```
Score z test
```

```
H0: p = p0 versus Ha: p > p0
```

```
Efficacy: Pocock
```

```
Study parameters:
```

```
alpha = 0.0250 (upper one-sided)
```

```
power = 0.9000
```

```
delta = 0.0610
```

```
p0 = 0.0500
```

```
pa = 0.1110
```

```
Expected sample size:
```

```
H0 = 202.50
```

```
Ha = 143.78
```

```
Info. ratio = 1.1001
```

```
N fixed = 186
```

```
N max = 204
```

```
Fixed-study crit. value = 1.9600
```

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

Look	Info. frac.	Efficacy		Sample size N
		Upper	p-value	
1	0.50	2.1783	0.0147	102
2	1.00	2.1783	0.0147	204

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

`gsdesign oneproportion` displays the specified study parameters, including `p0`, the proportion under the null hypothesis; `pa`, the proportion under the alternative hypothesis; and `delta`, the difference between `pa` and `p0`.

The next section of the output displays the expected sample size, which is the average sample size 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`), and the maximum sample size of the GSD (`N max`). The information ratio is the ratio of the sample size at the final look of the GSD to the sample size from a fixed study design.

Expected sample size is calculated under both the null and the alternative hypotheses. Because this design does not include futility bounds that would allow stopping to accept H_0 , the expected sample size under the null hypothesis is 202.5 participants, nearly the full sample size of 204. If the alternative hypothesis is true, the ability to stop early for treatment efficacy yields an expected sample size of 143.78, a savings over the 186 subjects required by the fixed design.

The table at the end of the output displays the stopping boundaries and sample sizes at each look, but it is informative to examine the bounds visually as well. We rerun the previous command with the `graphbounds` option to produce a graph of the stopping boundaries.

```
. gsdesign oneproportion 0.05 0.111, alpha(0.025) power(0.9) onesided
> efficacy(pocock) graphbounds
(output omitted)
```

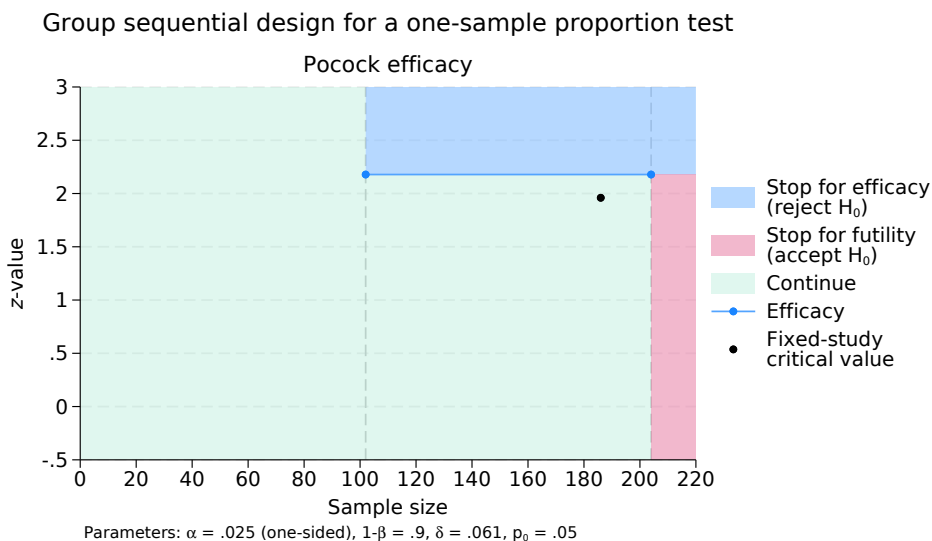


Figure 1. Pocock efficacy bounds for a test of one proportion

On the graph, the horizontal axis marks the sample size collected, and the vertical axis represents the z value of the test statistic. No tests are conducted until 102 observations have been collected, so for the first 101 observations, all z -values lie within the continuation region. Once data from 102 participants have been collected, a score test will be performed.

The z statistic from that test, z_1 , will be compared with the efficacy critical value, marked with a blue dot. If $z_1 \geq 2.178$, it lies in the blue rejection region, so we will reject H_0 and the trial will be stopped for efficacy; but if $z_1 < 2.178$, it lies in the green continuation region, and the trial will continue to the second and final look.

At the final look, the ORR will be calculated using data from all 204 participants, and a score test will be conducted. As before, if $z_2 \geq 2.178$, then H_0 will be rejected. But this time, if $z_2 < 2.178$, it will lie within the red acceptance region because there is no continuation region at the final look; so if $z_2 < 2.178$, we will accept H_0 . The concept of accepting the null hypothesis has a long history in the context of sequential hypothesis testing (see [ADAPT] **GSD intro** for details).

► Example 2: Efficacy and futility bounds with uneven information increments

We continue the scenario from [example 1](#), but we adjust the design by adding futility boundaries and additional interim looks at the data. Previously, we used a classical Pocock efficacy bound, which is characterized by having the same critical values at all looks. The classical Wang–Tsiatis efficacy bound offers an alternative that is known for having very conservative critical values at early looks but uses a final critical value that is close to the critical value of an equivalently powered fixed-sample test. Here we choose an error-spending approximation of the O’Brien–Fleming efficacy bound, which is similar in shape to the classical O’Brien–Fleming bound but is constructed using the error-spending method. See [Methods and formulas](#) in [\[ADAPT\] gsbounds](#) for more information about the error-spending approximation of the classical O’Brien–Fleming bound.

In addition to the error-spending O’Brien–Fleming-style efficacy bound, we add a nonbinding error-spending Hwang–Shih–de Cani futility bound with parameter $\gamma_f = -3$. Nonbinding futility bounds offer the option of stopping early to accept H_0 if the futility bound is crossed, but if the trial continues after crossing a nonbinding futility bound, the familywise type I error is still controlled. We also add two more looks in between the first look (with half the data) and the final look (with the complete dataset). Option `information()` allows us to schedule those looks to occur with 75% and 90% of the data.

```
. gsdesign oneproportion 0.05 0.111, alpha(0.025) power(0.9) onesided
> efficacy(errob Fleming) futility(hsdecani(-3))
> information(50 75 90 100) graphbounds

Group sequential design for a one-sample proportion test
Score z test
HO: p = p0 versus Ha: p > p0
Efficacy: Error-spending O'Brien-Fleming style
Futility: Error-spending Hwang-Shih-de Cani, nonbinding, gamma = -3.0000
Study parameters:
  alpha = 0.0250 (upper one-sided)
  power = 0.9000
  delta = 0.0610
  p0 = 0.0500
  pa = 0.1110
Expected sample size:
  HO = 124.28
  Ha = 147.09
Info. ratio = 1.0852
  N fixed = 186
  N max = 201
Fixed-study crit. value = 1.9600
Critical values, p-values, and sample sizes for a group sequential design
```

Look	Info. frac.	Efficacy		Futility		Sample size N
		Upper	p-value	Lower	p-value	
1	0.50	2.9626	0.0015	0.2963	0.3835	101
2	0.75	2.3590	0.0092	1.1477	0.1255	151
3	0.90	2.1649	0.0152	1.6551	0.0490	181
4	1.00	2.0731	0.0191	2.0731	0.0191	201

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

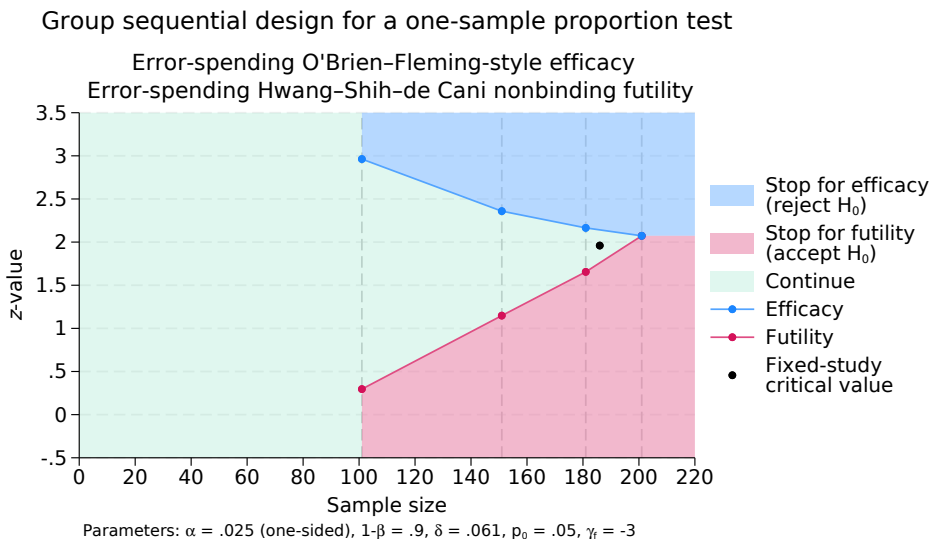


Figure 2. One proportion test with error-spending efficacy and futility bounds

Compared with the design in [example 1](#), the design with futility stopping has nearly the same maximum sample size (201 versus 204 observations) and expected sample size under H_a (147.09 versus 143.78 observations) but a much smaller expected sample size under H_0 (124.28 versus 202.5 observations).

The improved efficiency when H_0 is true is due to the ability to accept the null hypothesis at interim analyses. If $z_1 < 0.296$, the trial can be ended for futility with only 101 observations. As the study proceeds from one look to the next, the continuation region shrinks as the efficacy and futility bounds get closer together. By the third look, with 90% of the data (181 observations), the continuation region has shrunk to $z_3 \in [1.655, 2.165)$. At the final look, there is no continuation region because the efficacy and futility critical values are equal. If $z_4 \geq 2.073$, then we reject H_0 ; otherwise, we accept H_0 .

◀

► Example 3: Futility-only stopping

We continue the scenario from [example 2](#). In that example, we allowed for early stopping for efficacy (to reject H_0) as well as for futility (to accept H_0). This enabled a substantial reduction in sample size compared with a fixed study design, but sometimes there are reasons for intentionally choosing a less efficient design.

One argument against early stopping for efficacy is that a larger sample size allows for a better characterization of adverse events, which are harmful side effects of the treatment and negative medical outcomes not associated with an underlying disease. [Socinski et al. \(2008\)](#) report the incidence of several adverse events among participants taking sunitinib, including fatigue, pain, hypertension, and pulmonary hemorrhage. If sunitinib is effective at treating non-small cell lung cancer, it will be important to fully understand its side effects before using it to treat the general population. But if sunitinib is not an effective treatment, it would be both wasteful and unethical not to stop the trial as soon as the lack of efficacy is apparent.

To avoid stopping for efficacy before side effects can be characterized, we modify the design from [example 2](#) by removing the efficacy bound, but the rest of the design remains the same.

```
. gsdesign oneproportion 0.05 0.111, alpha(0.025) power(0.9) onesided
> futility(hsdecani(-3)) information(50 75 90 100) graphbounds
```

Group sequential design for a one-sample proportion test
 Score z test
 H0: $p = p_0$ versus Ha: $p > p_0$
 Futility: Error-spending Hwang-Shih-de Cani, nonbinding, $\gamma = -3.0000$

Study parameters:
 alpha = 0.0250 (upper one-sided)
 power = 0.9000
 delta = 0.0610
 p0 = 0.0500
 pa = 0.1110

Expected sample size:
 H0 = 123.48
 Ha = 197.78

Info. ratio = 1.0658
 N fixed = 186
 N max = 198

Fixed-study crit. value = 1.9600

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

Look	Info. frac.	Efficacy		Futility		Sample size N
		Upper	p-value	Lower	p-value	
1	0.50			0.2748	0.3917	99
2	0.75			1.1214	0.1311	148
3	0.90			1.6221	0.0524	178
4	1.00	1.9600	0.0250	1.9600	0.0250	198

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

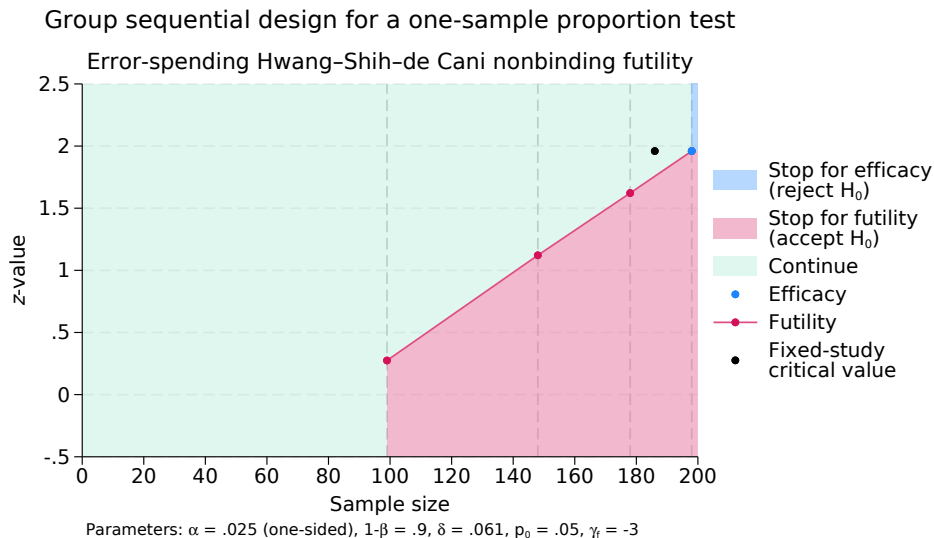


Figure 3. One proportion test with futility bounds

The expected sample size under H_a is nearly the maximum sample size (197.78 versus 198 observations), but the expected sample size under H_0 is a third less than the fixed-study sample size (123.48 versus 186 observations). This achieves our goal of collecting data from as many patients as possible if sunitinib is effective but saving participants from unnecessarily receiving the treatment if it is ineffective.

Efficacy stopping is not an option during the first three looks, so the efficacy critical value for these looks is reported as missing. At each of these looks, the trial can be terminated for futility if the test statistic is below the futility boundary, but because the futility bound is nonbinding, the familywise type I error will be controlled even if the trial is continued.

At the final look, it is at last possible to reject H_0 . If $z_4 \geq 1.96$, then we reject the null hypothesis; otherwise, we accept H_0 . On the graph, the efficacy bound is displayed as a single blue dot at the final look because that is the only time efficacy stopping is allowed. Interestingly, the nonbinding futility-only design uses the same critical value as the fixed design. To understand why, we display the contents of matrix `r(aspent_nofstop)`, the cumulative type I error spent per look if the trial does not stop for futility (even if it were to cross the futility bound at an interim look).

```
. matlist r(aspent_nofstop), format(%50.3g)
```

	alpha spent assuming no futility stopping
Look 1	0
Look 2	0
Look 3	0
Look 4	.025

Just like a fixed-sample trial with a one-sided significance level of 2.5%, this design spends its entire allotment of type I error during a single analysis. Practically speaking, this means that if H_0 is true, it is impossible to reject the null (and commit a type I error) during the first three looks. If H_0 is true and this trial is repeated many times (each time continuing until the final look even if the futility bound is crossed during an interim analysis), then 2.5% of the time we will erroneously

reject the null hypothesis at the final look. Viewed from this perspective, this error-spending regimen is essentially the same as that of a fixed-sample design, which is why it uses the same critical value.



Stored results

gsdesign oneproportion 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(continuity)</code>	1 if continuity correction is used, 0 otherwise
<code>r(delta)</code>	effect size
<code>r(diff)</code>	difference between the alternative and null proportions
<code>r(effparam)</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(info_ratio)</code>	ratio of maximum information required to that of a fixed study design
<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(nfractional)</code>	1 if <code>nfractional</code> is specified, 0 otherwise
<code>r(nlooks)</code>	number of analyses
<code>r(onesided)</code>	1 for a one-sided test, 0 otherwise
<code>r(p0)</code>	proportion under the null hypothesis
<code>r(pa)</code>	proportion under the alternative hypothesis
<code>r(pow_converged)</code>	1 if power calculation iteration algorithm converged, 0 otherwise
<code>r(pow_deltax)</code>	final parameter tolerance achieved for power calculation
<code>r(pow_ftolerance)</code>	requested distance of power calculation objective function from 0
<code>r(pow_function)</code>	final distance of power calculation objective function from 0
<code>r(pow_init)</code>	initial value for power calculation sample size
<code>r(pow_iter)</code>	number of iterations performed for power calculation
<code>r(pow_maxiter)</code>	maximum number of iterations for power calculation
<code>r(pow_tolerance)</code>	requested parameter tolerance for power calculation
<code>r(power)</code>	specified overall power
<code>r(power_a)</code>	attained overall power
<code>r(stop)</code>	0 for futility bounds, 1 for efficacy bounds, 2 for both
<code>r(z_fixed)</code>	critical value for an equivalent fixed study design

Macros

<code>r(cmd)</code>	<code>gsdesign</code>
<code>r(cmdline)</code>	command as typed
<code>r(direction)</code>	upper, lower, or two-sided
<code>r(effbnd)</code>	<code>pocock</code> , <code>obfleming</code> , <code>wtsiatis</code> , <code>errpocock</code> , <code>errob Fleming</code> , <code>kdemets</code> , or <code>hsdecani</code>
<code>r(futbnd)</code>	<code>pocock</code> , <code>obfleming</code> , <code>wtsiatis</code> , <code>errpocock</code> , <code>errob Fleming</code> , <code>kdemets</code> , or <code>hsdecani</code>
<code>r(method)</code>	<code>oneproportion</code>
<code>r(test)</code>	score or wald

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

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 oneproportion](#) 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

- Chow, S.-C., J. Shao, H. Wang, and Y. Lohknygina. 2018. *Sample Size Calculations in Clinical Research*. 3rd ed. Boca Raton, FL: CRC Press.
- Delgado, A., and A. K. Guddati. 2021. Clinical endpoints in oncology—A primer. *American Journal of Cancer Research* 11: 1121–1131.
- Gould, W. W., J. S. Pitblado, and B. P. Poi. 2010. *Maximum Likelihood Estimation with Stata*. 4th ed. College Station, TX: Stata Press.
- 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>.
- 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>.
- Levin, B., and X. Chen. 1999. Is the one-half continuity correction used once or twice to derive a well-known approximate sample size formula to compare two independent binomial distributions? *American Statistician* 53: 62–66. <https://doi.org/10.1080/00031305.1999.10474431>.
- 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>.
- 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>.
- Shatola, A., K. N. Nguyen, E. Kamangar, and M. E. Daly. 2020. Spontaneous regression of non-small cell lung cancer: A case report and literature review. *Cureus* 12: e6639. <https://doi.org/10.7759/cureus.6639>.
- Socinski, M. A., S. Novello, J. R. Brahmer, R. Rosell, J. M. Sanchez, C. P. Belani, R. Govindan, J. N. Atkins, H. H. Gillenwater, C. Pallares, L. Tye, P. Selaru, R. C. Chao, and G. V. Scagliotti. 2008. Multicenter, phase II trial of sunitinib in previously treated, advanced non–small-cell lung cancer. *Journal of Clinical Oncology* 26: 650–656. <https://doi.org/10.1200/JCO.2007.13.9303>.

- Tenhunen, O., F. Lasch, A. Schiel, and M. Turpeinen. 2020. Single-arm clinical trials as pivotal evidence for cancer drug approval: A retrospective cohort study of centralized European marketing authorizations between 2010 and 2019. *Clinical Pharmacology and Therapeutics* 108: 653–660. <https://doi.org/10.1002/cpt.1965>.
- 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] **gsdesign twoproportions** — Group sequential design for a two-sample proportions test
- [ADAPT] **Glossary**
- [PSS-2] **power oneproportion** — Power analysis for a one-sample proportion test
- [R] **bitest** — Binomial probability test
- [R] **prtest** — Tests of proportions