Title

stata.com

gsdesign twomeans — Group sequential design for a two-sample means test

Description

`gsdesign twomeans` computes stopping boundaries and sample sizes for interim analyses of clinical trials using a two-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 two means, see [PSS-2] **power twomeans**.

Quick start

Sample size and stopping boundaries for a two-sided test of $H_0: \mu_1 = \mu_2$ versus $H_a: \mu_1 \neq \mu_2$, with default significance level $\alpha = 0.05$ and power of 0.8 to detect the difference between a control-group mean of $m_1 = 3$ and an experimental-group mean of $m_2 = 7$, with shared standard deviation of 9, using default group sequential specifications of O'Brien–Fleming efficacy boundaries with two analyses (one interim, one final)

```
gsdesign twomeans 3 7, sd(9)
```

Same as above, but for an upper one-sided test with $\alpha = 0.025$, and replace the O'Brien–Fleming efficacy bound with an error-spending Pocock-style bound with three looks

```
gsdesign twomeans 3 7, sd(9) alpha(0.025) onesided            ///
     efficacy(errpocock) nlooks(3)
```

Same as above, but add a nonbinding error-spending O'Brien–Fleming-style futility bound, and specify the difference between means instead of the experimental-group mean

```
gsdesign twomeans 3, diff(4) sd(9) alpha(0.025) onesided      ///
     efficacy(errpocock) futility(errobfleming) nlooks(3)
```

Same as above, but specify a control-group standard deviation of 6 and an experimental-group standard deviation of 12, and allocate twice as many subjects to the experimental group as the control group

```
gsdesign twomeans 3, diff(4) sd1(6) sd2(12) nratio(2)         ///
     alpha(0.025) onesided efficacy(errpocock)                ///
     futility(errobfleming) nlooks(3)
```

Same as above, but time the looks to occur with 50%, 75%, and 100% of the data, and plot the boundaries

```
gsdesign twomeans 3, diff(4) sd1(6) sd2(12) nratio(2)         ///
     alpha(0.025) onesided efficacy(errpocock)                ///
     futility(errobfleming) information(50 75 100)
```

Menu

Statistics > Power, precision, and sample size

Syntax

`gsdesign twomeans` m_1 m_2 [, *twomeansopts boundopts*]

where m_1 is the mean of the control (reference) group and m_2 is the mean of the experimental (treatment) group.

twomeansopts	Description
Main	
`alpha(#)`	overall significance level for all tests; default is `alpha(0.05)`
`power(#)`	overall power for all tests; default is `power(0.8)`
`beta(#)`	overall probability of type II error for all tests; default is `beta(0.2)`
`onesided`	request a one-sided test; default is two-sided
`nfractional`	report fractional sample size
`nratio(#)`	ratio of sample sizes of experimental to control groups; default is `nratio(1)`, meaning equal group sizes
`diff(#)`	difference between the experimental-group mean and the control-group mean, $m_2 - m_1$; specify instead of the experimental-group mean m_2
`sd(#)`	common standard deviation of the control and the experimental groups assuming equal standard deviations in both groups; default is `sd(1)`
`sd1(#)`	standard deviation of the control group; requires `sd2()`
`sd2(#)`	standard deviation of the experimental group; requires `sd1()`
`knownsds`	request computation assuming known standard deviations for both groups; default is to assume unknown standard deviations
`force`	allow calculation with unsupported `power twomeans` options
`poweriteration(`*powiteropts*`)`	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.

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

boundopts	Description
Bounds	
`efficacy(`*boundary*`)`	boundary for efficacy stopping; if neither `efficacy()` nor `futility()` is specified, the default is `efficacy(obfleming)`
`futility(`*boundary*`[ , binding ])`	boundary for futility stopping; use `binding` to request binding futility bounds (default is nonbinding)
`nlooks(`#`[ , equal ])`	total number of analyses (`nlooks()` − 1 interim analyses and one final analysis); use `equal` to enforce equal information increments; if neither `nlooks()` nor `information()` is specified, the default is `nlooks(2)`
`information(`*numlist*`)`	sequence of information levels for analyses; default is evenly spaced
`nopvalues`	suppress p-values
Graph	
`graphbounds[ (`*graphopts*`) ]`	graph boundaries
`matlistopts(`*general_options*`)`	control the display of boundaries and sample size; seldom used
optimopts	optimization options for boundary calculations; seldom used

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

boundary	Description
`obfleming`	classical O'Brien–Fleming bound
`pocock`	classical Pocock bound
`wtsiatis(#)`	classical Wang–Tsiatis bound with specified parameter value
`errpocock`	error-spending Pocock-style bound
`errobfleming`	error-spending O'Brien–Fleming-style bound
`kdemets(#)`	error-spending Kim–DeMets bound with specified parameter value
`hsdecani(#)`	error-spending Hwang–Shih–de Cani bound with specified parameter value

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

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

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

`diff(#)` specifies the difference between the experimental-group mean and the control-group mean, $m_2 - m_1$. You can either specify the experimental-group mean m_2 as a command argument or specify the difference between the two means in `diff()`. If you specify `diff(#)`, the experimental-group mean is computed as $m_2 = m_1 + \#$.

`sd(#)` specifies the common standard deviation of the control and the experimental groups assuming equal standard deviations in both groups. The default is `sd(1)`.

`sd1(#)` specifies the standard deviation of the control group. If you specify `sd1()`, you must also specify `sd2()`.

`sd2(#)` specifies the standard deviation of the experimental group. If you specify `sd2()`, you must also specify `sd1()`.

`knownsds` requests that standard deviations of each group be treated as known in the computations. By default, standard deviations are treated as unknown and the computations are based on a two-sample t test, which uses Student's t distribution as a sampling distribution of the test statistic. If `knownsds` is specified, the computation is based on a two-sample 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 twomeans` 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 twomeans`, 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 twomeans` but are not shown in the dialog box:

`force` indicates that `gsdesign twomeans` should allow unsupported `power twomeans` options, such as options specifying a cluster randomized design. Even with option `force`, the `power twomeans` options specified must be compatible with sample-size determination, not effect size or power calculation. In addition, *numlist*s 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`, `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**.

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

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

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

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

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

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

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

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

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

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

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

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

`nrtolerance(#)` specifies the tolerance for the scaled gradient. Convergence is declared when $\mathbf{g}\mathbf{H}^{-1}\mathbf{g}' <$ `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 Δ yield bounds that are more conservative at early looks.

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

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

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

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

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

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

Remarks and examples

stata.com

Remarks are presented under the following headings:

This entry describes the use of the `gsdesign twomeans` command for designing a group sequential analysis for a two-sample means 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 twomeans** for details about study design for a two-sample means test.

Introduction

In a classic randomized controlled trial, participants are randomly assigned to one of two groups: the experimental group (which receives the treatment being tested) and the control group (which receives either a placebo or the existing standard of care, if one exists). The two groups are often called arms, making this a two-arm trial. Examples of treatments include new drugs, medical devices, and medical procedures. To determine the efficacy of the treatment, the responses of participants in the experimental arm are compared with the responses of participants in the control arm. When the responses are continuous, a two-sample test of means can be performed to determine whether the mean of the experimental arm is the same as that of the control arm.

`gsdesign twomeans` calculates sample size and stopping boundaries for a group sequential trial comparing the population mean of the experimental group against that of the control group. Specifically, we consider the null hypothesis $H_0: \mu_1 = \mu_2$ versus the two-sided alternative hypothesis $H_a: \mu_1 \neq \mu_2$, the upper one-sided alternative $H_a: \mu_1 > \mu_2$, or the lower one-sided alternative $H_a: \mu_1 < \mu_2$.

The actual test conducted will depend on whether the population standard deviation of both groups is known. 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 two-sample z test. A z test is also commonly used when sample sizes are large, even when the population standard deviations are unknown. This is because the distribution of the test statistic approaches a normal distribution as the sample size increases.

If the sample is not of sufficient size to use a large-sample z test and the standard deviations are unknown but assumed to be equal, then the test statistic has an exact Student's t distribution under the null hypothesis and the corresponding test is referred to as a two-sample t test. If the two unknown standard deviations are not equal, then the distribution of the test statistic under the null hypothesis can be approximated by a t distribution with degrees of freedom estimated using Satterthwaite's method, and the resulting test is known as Satterthwaite's t test.

The required sample size estimated by `gsdesign twomeans` 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. If the analysis is performed using a t test, the p-value from the t test can be compared with the p-values corresponding to the critical values for the boundaries. This is demonstrated in example 2.

Using gsdesign twomeans

`gsdesign twomeans` calculates sample size and stopping boundaries for a group sequential trial comparing the means of two populations. `gsdesign twomeans` can be thought of as a combination of `power twomeans` 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 the means of the control and experimental groups, m_1 and m_2, or by specifying m_1 and the difference $m_2 - m_1$ in the `diff()` option. There is no default value for `diff()`, so either m_1 and m_2 or m_1 and `diff()` must be included as part of the command specification. Another aspect of the effect size is the standard deviation of the responses. This is specified with the `sd()` option if both groups share a common standard deviation and specified with the `sd1()` and `sd2()` options otherwise. The default behavior is to assume a common standard deviation of 1 and to assume that the standard deviation must be estimated from the sample. If the true population standard deviation is known a priori, the `knownsds` option requests that sample-size calculations be performed for a z test, not a t test.

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

The `alpha()`, `power()`, `beta()`, and `onesided` options 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 sizes in each arm are rounded up to whole numbers at each look, but the `nfractional` option can be used to report fractional sample sizes. If `nlooks()` is specified, the default behavior is to divide information evenly among 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 per arm at each look.

Background for examples 1 and 2

Alzheimer's disease (AD) is an incurable neurodegenerative disease characterized by memory loss and progressive cognitive decline. Historically, the only sure way to diagnose AD was through autopsy, but recent research has identified biomarkers that can be used to diagnose AD and track disease progression in living patients.

One of the most promising biomarkers for AD is glucose metabolism in the brain, which can be measured by a type of imaging known as fluorodeoxyglucose positron emission tomography (FDG PET). Mosconi (2005) writes that FDG PET "has revealed glucose metabolic reductions in the parieto-temporal, frontal and posterior cingulate cortices to be the hallmark of AD". FDG PET measures glucose metabolism as standardized uptake value ratios (SUVRs), which can be used to track disease progression, with SUVR levels falling as the disease becomes more severe.

Matthews et al. (2021) conducted a phase 2 clinical trial of the neuroprotective agent riluzole versus a placebo for the treatment of mild AD. They used FDG PET to measure the SUVR of each subject at baseline and again after six months of treatment, and they compared the average change in SUVR in the control arm against that of the treatment arm. The results of their study were encouraging, with smaller declines in SUVR observed in the experimental arm than in the control arm.

Computing sample size and stopping boundaries with known standard deviation

▷ Example 1: Pocock efficacy bounds for a test of two sample means

Suppose that we want design a follow-up study focusing on a target population of Alzheimer's patients suffering from clinical depression. We will consider a placebo-controlled clinical trial, with participants randomized to the treatment and placebo arms at a 1:1 ratio. The SUVR in the posterior cingulate will be measured for each participant at baseline and again after six months of treatment, and the mean change in SUVR from the control arm (μ_1) and experimental arm (μ_2) will be calculated. We will test the null hypothesis $H_0\colon \mu_1 = \mu_2$ versus the two-sided alternative hypothesis $H_a\colon \mu_1 \neq \mu_2$.

Based on previous studies, we anticipate SUVR will decrease by an average of 0.05 in the control arm and 0.01 in the experimental arm, giving $\mu_1 = -0.05$ and $\mu_2 = -0.01$. We will assume both arms have a known standard deviation of 0.035, but this assumption is likely unrealistic and is relaxed in the next example.

We require 80% power to detect the specified difference in means, and we will conduct a two-sided trial with familywise significance level of 5%, using Pocock efficacy bounds with two evenly spaced looks. Except for the efficacy boundary, these design specifications correspond to the default values of the respective options in `gsdesign twomeans`, so they are not specified.

```
. gsdesign twomeans -0.05 -0.01, sd(0.035) knownsds efficacy(pocock)
Group sequential design for a two-sample means test
z test assuming sd1 = sd2 = sd
H0: m2 = m1 versus Ha: m2 != m1
Efficacy: Pocock
Study parameters:
      alpha =  0.0500  (two-sided)
      power =  0.8000
      delta =  0.0400
         m1 = -0.0500
         m2 = -0.0100
         sd =  0.0350
Expected sample size:
         H0 =   27.59
         Ha =   21.22
Info. ratio =  1.1104
    N fixed =      26
      N max =      28
     N1 max =      14
     N2 max =      14
Fixed-study crit. values = ±1.9600
Critical values, p-values, and sample sizes for a group sequential design
```

Look	Info. frac.	Efficacy Lower	Efficacy Upper	p-value	Sample size N1	Sample size N2	Sample size N
1	0.50	-2.1783	2.1783	0.0294	7	7	14
2	1.00	-2.1783	2.1783	0.0294	14	14	28

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

`gsdesign twomeans` begins by displaying a description of the test being performed, the type of bounds, and a summary of the parameters used in the design.

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`), the maximum sample size of the GSD (`N max`), and the maximum sample sizes for each group (`N1 max` and `N2 max`). The information ratio is the ratio of the maximum sample size of the GSD to the fixed-study sample size.

We can compare the expected sample sizes with the sample size for a fixed study and the maximum sample size of the GSD. If the null hypothesis of equal change in SUVR between the control and the experimental arms were true, the average trial would require 27.59 participants, nearly the full sample size of 28. This is because the efficacy bounds do not allow for early stopping to accept H_0, so if the null hypothesis is true, the trial will usually proceed to the final look. If H_a is true, the average trial will require 21.22 participants, which is a savings over the 26 participants required by the fixed trial.

We also see the critical value for a fixed study with an equivalent significance level. The critical values of ± 1.96 would be used to reject H_0 at the 0.05 level if a fixed study design were conducted instead of a GSD.

Finally, `gsdesign twomeans` displays a table with the critical values and p-values for the efficacy stopping boundaries as well as the sample sizes at each look. Pocock efficacy bounds use the same critical value at all looks, and to maintain a familywise type I error of 0.05, the z statistic must meet or exceed ± 2.178 at any look to reject H_0.

To plot the bounds for visual inspection, we rerun the previous command but add the `graphbounds` option.

```
. gsdesign twomeans -0.05 -0.01, sd(0.035) knownsds efficacy(pocock) graphbounds
```

(*output omitted*)

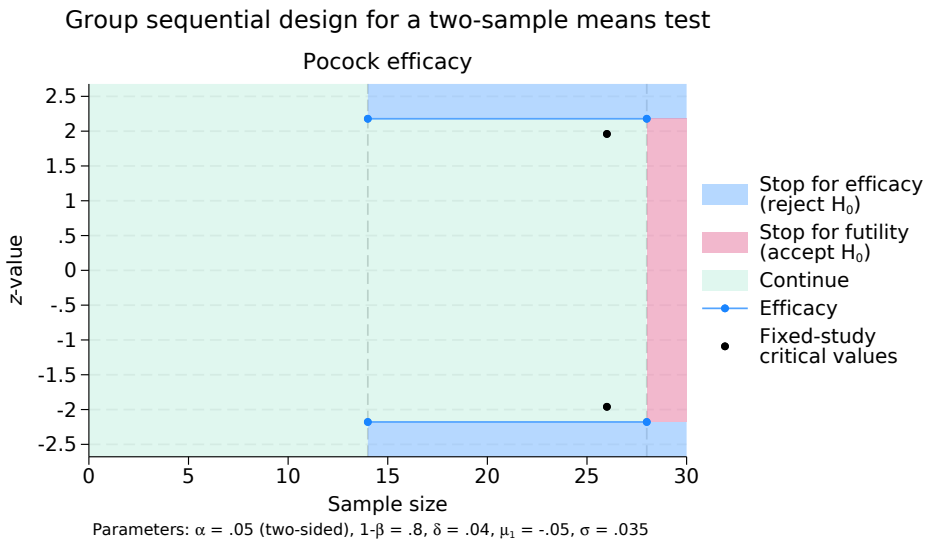

Figure 1. Two-sided Pocock efficacy bounds for a test of the equality of two means

On the graph, the horizontal axis is the total sample size (the sum of the sample sizes of both arms), and the vertical axis is the z-value of the test statistic. The efficacy bounds are marked as blue lines, with the location of looks indicated by blue dots. For comparison, the critical values of an equivalently powered fixed study are marked with black dots.

The rejection region is shaded blue, the acceptance region red, and the continuation region green. Before the first look, which occurs once results have been collected from 14 participants (7 in the control arm and 7 in the experimental arm), it is impossible to reject H_0 because no test has been conducted, so the entire range of z-values is in the continuation region. Beginning at the first look, z-values equal to or more extreme than ± 2.1783 are in the rejection region. The efficacy-only design does not permit early stopping to accept H_0, so the acceptance region begins at the second and final look, and it encompasses z-values less extreme than ± 2.1783.

◁

Unknown standard deviation and hypothesis tests on means

▷ Example 2: Unknown standard deviation, specifying difference between means

In the previous example, we relied on the assumption that the population standard deviation was known to be 0.035 in both arms, which led to sample-size calculations based on a two-sample z test. Here we relax that assumption and assume that the standard deviation will be estimated from the sample. We anticipate the standard deviation of the control group will be 0.05, while the standard deviation of the experimental group will be 0.035. This yields sample sizes for a t test, which is demonstrated below.

Additionally, instead of specifying μ_2 directly, here we use the `diff()` option to specify the difference in means between the two arms. We omit the `graphbounds` option because the graph is minimally changed from the previous example.

```
. gsdesign twomeans -0.05, diff(0.04) sd1(0.05) sd2(0.035) efficacy(pocock)

Group sequential design for a two-sample means test
Satterthwaite's t test assuming unequal variances
H0: m2 = m1 versus Ha: m2 != m1

Efficacy: Pocock

Study parameters:
      alpha =  0.0500  (two-sided)
      power =  0.8000
      delta =  0.0400
         m1 = -0.0500
         m2 = -0.0100
       diff =  0.0400
        sd1 =  0.0500
        sd2 =  0.0350

Expected sample size:
         H0 =   43.35
         Ha =   33.60

Info. ratio =  1.1104
    N fixed =      40
      N max =      44
     N1 max =      22
     N2 max =      22

Fixed-study crit. values = ±1.9600

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

Look	Info. frac.	Efficacy Lower	Efficacy Upper	p-value	Sample size N1	Sample size N2	Sample size N
1	0.50	-2.1783	2.1783	0.0294	11	11	22
2	1.00	-2.1783	2.1783	0.0294	22	22	44

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

Specifying the difference in means instead of the experimental group mean has not changed the study parameters, but changing our assumptions about the standard deviations has increased the fixed-study sample size from 26 in the previous example to 40 here. The information ratio is unchanged, but the sample sizes required by the GSD have increased correspondingly, as have the expected sample sizes under H_0 and H_a.

The testing procedure has also changed. Instead of comparing the z statistic directly with the efficacy critical values, a t test is performed, and we use the significance level approach described in [ADAPT] **gsbounds**. The table at the bottom of the output provides the p-value corresponding to each critical value. We can compare the p-value for the t test with these p-value boundaries.

Suppose the first look is conducted with 11 observations from each arm. From the data we collect, we have a mean change in SUVR of -0.014 in the experimental arm with standard deviation 0.038 and a mean change in SUVR of -0.062 in the control arm with standard deviation 0.057. We conduct a t test using `ttesti`, the immediate form of the [R] **ttest** command. We type `ttesti 11 -0.014 0.038 11 -0.062 0.057, unequal`, with the first three arguments specifying the experimental group sample size, mean, and standard deviation, respectively, and the following three arguments specifying the control group sample size, mean, and standard deviation. Option `unequal` indicates that we do not assume that the population standard deviations of the two groups are equal and instructs `ttesti` to use Satterthwaite's method to estimate the degrees of freedom for the t test.

```
. ttesti 11 -0.014 0.038 11 -0.062 0.057, unequal
Two-sample t test with unequal variances
```

	Obs	Mean	Std. err.	Std. dev.	[95% conf.	interval]
x	11	-.014	.0114574	.038	-.0395287	.0115287
y	11	-.062	.0171861	.057	-.1002931	-.0237069
Combined	22	-.038	.0113582	.0532747	-.0616207	-.0143793
diff		.048	.0206552		.0045018	.0914982

```
    diff = mean(x) - mean(y)                                      t =   2.3239
H0: diff = 0                     Satterthwaite's degrees of freedom =  17.4227
    Ha: diff < 0                 Ha: diff != 0                 Ha: diff > 0
 Pr(T < t) = 0.9838         Pr(|T| > |t|) = 0.0325          Pr(T > t) = 0.0162
```

The t statistic of 2.324 cannot be compared directly with the efficacy critical values, but the p-value for the two-sided test, $p_1 = 0.0325$, can be compared with the p-value equivalent of the critical value at the first look. Because $p_1 > 0.0294$, we cannot reject H_0 at this look, and the trial continues to the second and final look. At the second look, we conduct another t test and calculate the p-value, p_2. If $p_2 \leq 0.0294$, then H_0 is rejected; otherwise, H_0 is accepted.

◁

Background for example 3

Wilkinson et al. (2011) published the results of a clinical trial of levalbuterol used as a continuous nebulization for the treatment of acute pediatric asthma exacerbations (asthma attacks). Levalbuterol was found to be inferior to the standard of care, nebulization with racemic albuterol, which was an unexpected result. To determine whether levalbuterol is more effective at higher doses, we wish to conduct a similar study using 7.5 mg of levalbuterol instead of the 3.75 mg dose used by Wilkinson et al.

Study participants are children aged 6 to 17 who have previously been diagnosed with asthma by a physician and who present to the emergency department with acute asthma exacerbation of moderate severity. Participants are randomly assigned to either the treatment or the control group. Participants in the treatment group receive 7.5 mg of levalbuterol administered via nebulizer over the course of one hour, while participants in the control group receive the standard of care, which is a one-hour nebulization with 7.5 mg of racemic albuterol.

Upon hospital admission, each participant's one-second forced expiratory volume is assessed. This is a measurement of how much air the participant can exhale in one second, and higher values indicate better lung function. A second measurement of expiratory volume is conducted two hours after treatment, and the change in one-second forced expiratory volume (ΔFEV1) is calculated as the percent improvement (or percent decline, for negative ΔFEV1) compared with the participant's baseline value.

Efficacy and futility stopping

▷ Example 3: Error-spending efficacy and futility bounds

Suppose we wish to design a clinical trial that will compare the average ΔFEV1 in the control arm, μ_1, against the average ΔFEV1 in the experimental arm, μ_2. We will test the null hypothesis $H_0 : \mu_1 = \mu_2$ versus the one-sided alternative $H_a : \mu_1 < \mu_2$ with a familywise significance level of 2.5%. We require 90% power to detect the difference between a 50% increase in mean ΔFEV1 in the control arm and a 60% mean increase in the experimental arm, with a common standard deviation of 35. Suppose that we are particularly concerned about adverse events in the group receiving high-dose levalbuterol, so we will randomize participants to the experimental and control arms in a 2:1 ratio, ensuring a larger sample size (and more power) to detect adverse events in the experimental arm.

Depending on the recruitment rate, this clinical trial could take months or even years to complete. There is an ethical imperative not to expose participants to inferior treatments, so if high-dose levalbuterol is more effective than racemic albuterol, we would want to know as soon as possible. To this end, we employ an error-spending O'Brien–Fleming-style efficacy bound. If high-dose levalbuterol is not superior to racemic albuterol, we want to terminate the trial early for futility, so we also specify a nonbinding Kim–DeMets futility bound with parameter $\rho_f = 2$. If a nonbinding futility bound is crossed, the trial can be stopped for futility, but if the trial is continued, the familywise type I error is still controlled at the desired level. We specify a four-look design and graph the bounds for inspection.

```
. gsdesign twomeans 50 60, sd(35) nratio(2) alpha(0.025) power(0.9) onesided
> efficacy(errobfleming) futility(kdemets(2)) nlooks(4) graphbounds

Group sequential design for a two-sample means test
t test assuming sd1 = sd2 = sd
H0: m2 = m1 versus Ha: m2 > m1

Efficacy: Error-spending O'Brien-Fleming style
Futility: Error-spending Kim-DeMets, nonbinding, rho = 2.0000

Study parameters:
      alpha =  0.0250  (upper one-sided)
      power =  0.9000
     nratio =  2.0000
      delta = 10.0000
         m1 = 50.0000
         m2 = 60.0000
         sd = 35.0000

Expected sample size:
         H0 =  353.85
         Ha =  470.42

Info. ratio =  1.0859
    N fixed =     582
      N max =     632
     N1 max =     211
     N2 max =     421

Fixed-study crit. value = 1.9600

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

Look	Info. frac.	Efficacy Upper	Efficacy p-value	Futility Lower	Futility p-value
1	0.25	4.3326	0.0000	-0.8088	0.7907
2	0.50	2.9631	0.0015	0.3702	0.3556
3	0.75	2.3590	0.0092	1.2438	0.1068
4	1.00	2.0141	0.0220	2.0141	0.0220

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

Look	Sample size N1	N2	N
1	53	106	159
2	106	211	317
3	158	316	474
4	211	421	632

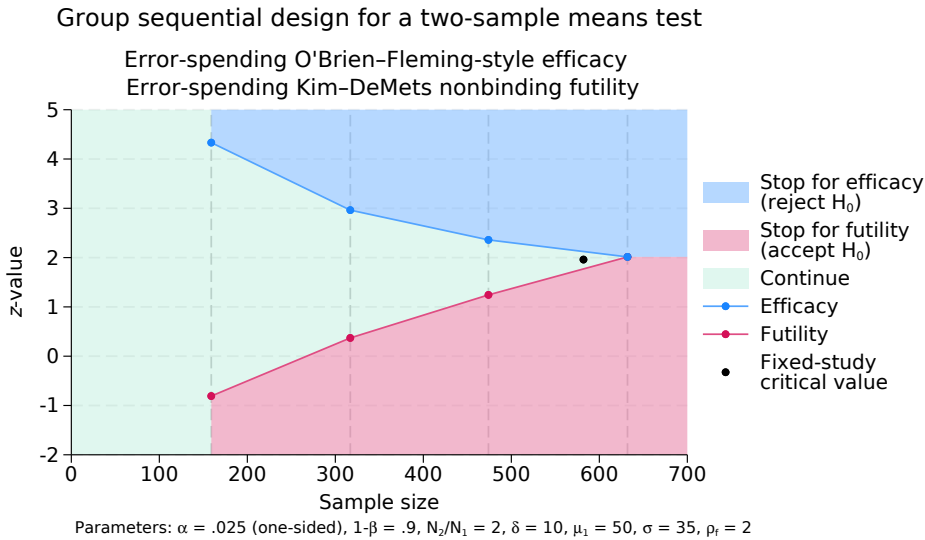

Figure 2. Sample size for a test of the equality of two means with efficacy and futility bounds

The output begins with a description of the test being performed, the types of boundaries, and a summary of the parameters used in the design.

Next it displays the expected sample sizes under the null and alternative hypotheses, the information ratio, the sample size that would be required for an equivalently powered fixed study, the maximum sample size for a GSD, and the critical value for a fixed study.

The sample size for a corresponding fixed study is the same sample size that would be calculated had we run `power twomeans 50 60, sd(35) nratio(2) power(0.9) alpha(0.025) onesided`. The fixed-study critical value of 1.96 would be used to reject H_0 at the 0.025 level using a fixed study design.

The expect sample sizes under the null and alternative hypotheses are both smaller than the fixed-study sample size. This is not surprising, because this design incorporates both efficacy and futility stopping.

At the bottom of the output is a table with the critical values and p-values for the stopping boundaries as well as the sample sizes at each look. O'Brien–Fleming boundaries are very conservative at early looks, with final critical values only slightly larger than those of an equivalent fixed-sample design. The error-spending approximation of the classical O'Brien–Fleming bounds shares this property, yielding efficacy critical values at the first look of 4.333, but only 2.014 at the final look, a minor increase over the fixed-study critical values.

While the population standard deviation was not assumed to be known when designing this trial, the large sample sizes involved enable the use of a large-sample z test. The first look is conducted when we have data from 53 controls and 106 experimental participants, and the test statistic, z_1, is compared with the boundary critical values. If $z_1 \geq 4.333$, we reject H_0 and terminate the trial early for efficacy. Even if we terminate the trial after the first look, we will have data about adverse events for over 100 experimental participants because we randomized twice as many participants to the experimental arm as the control arm. If $z_1 < -0.809$, we may accept H_0 and terminate the trial for futility, but if the trial is continued, the familywise type I error is still controlled. If $z_1 \in [-0.809, 4.333)$, then the trial must continue to the next look.

At the second look, the testing procedure is the same, but the critical values of the efficacy bounds and the futility bounds have narrowed, shrinking the continuation region to $z_2 \in [0.37, 2.963)$. If the trial continues to the third look, the continuation region is further reduced to $z_3 \in [1.244, 2.359)$. At the fourth and final look, the futility critical values equal the efficacy critical values and there is no continuation region: If $z_4 \geq 2.014$, then H_0 is rejected; otherwise, H_0 is accepted. The boundaries are displayed on the graph, and the critical value for a fixed study with equivalent significance level and power is marked with a black dot.

◁

Stored results

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

Scalars

`r(alpha)`	overall significance level (familywise type I error)
`r(beta)`	overall probability of a type II error
`r(binding)`	`1` for binding futility bounds, `0` for nonbinding
`r(delta)`	effect size
`r(diff)`	difference between the experimental- and control-group means
`r(effparam)`	efficacy parameter (if `wtsiatis()`, `kdemets()`, or `hsdecani()` specified)
`r(ESS0)`	expected sample size under null hypothesis
`r(ESS1)`	expected sample size under alternative hypothesis
`r(futparam)`	futility parameter (if `wtsiatis()`, `kdemets()`, or `hsdecani()` specified)
`r(info_ratio)`	ratio of maximum information required to that of a fixed study design
`r(knownsds)`	`1` if option `knownsds` is specified, `0` otherwise
`r(m1)`	control-group mean
`r(m2)`	experimental-group mean
`r(N_fixed)`	sample size of a fixed study design
`r(N_fixedfrac)`	fractional sample size of a fixed study design
`r(N_max)`	maximum sample size if the study continues to completion
`r(N1_fixed)`	sample size of the control group in a fixed study design
`r(N1_fixedfrac)`	fractional sample size of the control group in a fixed study design
`r(N1_max)`	maximum sample size of the control group if the study continues to completion
`r(N2_fixed)`	sample size of the experimental group in a fixed study design
`r(N2_fixedfrac)`	fractional sample size of the experimental group in a fixed study design
`r(N2_max)`	maximum sample size of the experimental group if the study continues to completion
`r(nfractional)`	`1` if `nfractional` is specified, `0` otherwise
`r(nlooks)`	number of analyses
`r(nratio)`	specified ratio of sample sizes, $N2/N1$
`r(nratio_a)`	attained ratio of sample sizes
`r(onesided)`	`1` for a one-sided test, `0` otherwise
`r(pow_converged)`	`1` if power calculation iteration algorithm converged, `0` otherwise
`r(pow_deltax)`	final parameter tolerance achieved for power calculation
`r(pow_ftolerance)`	requested distance of power calculation objective function from 0
`r(pow_function)`	final distance of power calculation objective function from 0
`r(pow_init)`	initial value for power calculation sample size
`r(pow_iter)`	number of iterations performed for power calculation
`r(pow_maxiter)`	maximum number of iterations for power calculation
`r(pow_tolerance)`	requested parameter tolerance for power calculation
`r(power)`	specified overall power
`r(power_a)`	attained overall power
`r(sd)`	common standard deviation of both groups (if `sd1()` and `sd2()` not specified)
`r(sd1)`	standard deviation of the control group
`r(sd2)`	standard deviation of the experimental group
`r(stop)`	`0` for futility bounds, `1` for efficacy bounds, `2` for both
`r(unequal)`	`0` if `sd1 = sd2`, `1` otherwise
`r(z_fixed)`	critical value for an equivalent fixed study design

Macros

r(cmd)	gsdesign
r(cmdline)	command as typed
r(direction)	upper, lower, or two-sided
r(effbnd)	pocock, obfleming, wtsiatis, errpocock, errobfleming, kdemets, or hsdecani
r(futbnd)	pocock, obfleming, wtsiatis, errpocock, errobfleming, kdemets, or hsdecani
r(method)	twomeans

Matrices

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

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.

Matthews, D. C., X. Mao, K. Dowd, D. Tsakanikas, C. S. Jiang, C. Meuser, R. D. Andrews, A. S. Lukic, J. Lee, N. Hampilos, N. Shafiian, M. Sano, P. D. Mozley, H. Fillit, B. S. McEwen, D. C. Shungu, and A. C. Pereira. 2021. Riluzole, a glutamate modulator, slows cerebral glucose metabolism decline in patients with Alzheimer's disease. *Brain* 144: 3742–3755. https://doi.org/10.1093/brain/awab222.

Mosconi, L. 2005. Brain glucose metabolism in the early and specific diagnosis of Alzheimer's disease. *European Journal of Nuclear Medicine and Molecular Imaging* 32: 486–510. https://doi.org/10.1007/s00259-005-1762-7.

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.

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.

Wilkinson, M., B. Bulloch, P. Garcia-Filion, and L. Keahey. 2011. Efficacy of racemic albuterol versus levalbuterol used as a continuous nebulization for the treatment of acute asthma exacerbations: A randomized, double-blind, clinical trial. *Journal of Asthma* 48: 188–193. https://doi.org/10.3109/02770903.2011.554939.

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

[ADAPT] **Glossary**

[PSS-2] **power twomeans** — Power analysis for a two-sample means test

[R] **ttest** — t tests (mean-comparison tests)

[R] **ztest** — z tests (mean-comparison tests, known variance)