

power — Power and sample-size analysis for hypothesis tests

- Syntax

Remarks and examples

Also see
- Menu

Stored results
- Description

Methods and formulas
- Options

Reference

Syntax

Compute sample size

```
power method ... [ , power(numlist) power_options ... ]
```

Compute power

```
power method ..., n(numlist) [power_options ... ]
```

Compute effect size and target parameter

```
power method ..., n(numlist) power(numlist) [power_options ... ]
```

<i>method</i>	Description
One sample	
onemean	One-sample mean test (one-sample <i>t</i> test)
oneproportion	One-sample proportion test
onecorrelation	One-sample correlation test
onevariance	One-sample variance test
Two independent samples	
twomeans	Two-sample means test (two-sample <i>t</i> test)
two proportions	Two-sample proportions test
twocorrelations	Two-sample correlations test
two variances	Two-sample variances test
Two paired samples	
pairedmeans	Paired-means test (paired <i>t</i> test)
paired proportions	Paired-proportions test (McNemar’s test)
Analysis of variance	
oneway	One-way ANOVA
two way	Two-way ANOVA
repeated	Repeated-measures ANOVA

<i>power_options</i>	Description
Main	
* <u>alpha</u> (<i>numlist</i>)	significance level; default is <code>alpha(0.05)</code>
* <u>power</u> (<i>numlist</i>)	power; default is <code>power(0.8)</code>
* <u>beta</u> (<i>numlist</i>)	probability of type II error; default is <code>beta(0.2)</code>
* <u>n</u> (<i>numlist</i>)	total sample size; required to compute power or effect size
* <u>n1</u> (<i>numlist</i>)	sample size of the control group
* <u>n2</u> (<i>numlist</i>)	sample size of the experimental group
* <u>nratio</u> (<i>numlist</i>)	ratio of sample sizes, N_2/N_1 ; default is <code>nratio(1)</code> , meaning equal group sizes
<u>compute</u> (<u>n1</u> <u>n2</u>)	solve for N_1 given N_2 or for N_2 given N_1
<u>nfractional</u>	allow fractional sample sizes
<u>direction</u> (<u>upper</u> <u>lower</u>)	direction of the effect for effect-size determination; default is <code>direction(upper)</code> , which means that the postulated value of the parameter is larger than the hypothesized value
<u>onesided</u>	one-sided test; default is two sided
<u>parallel</u>	treat number lists in starred options as parallel when multiple values per option are specified (do not enumerate all possible combinations of values)
Table	
[<u>no</u>] <u>table</u> [(<i>tablespec</i>)]	suppress table or display results as a table; see [PSS] power, table
<u>saving</u> (<i>filename</i> [, <u>replace</u>])	save the table data to <i>filename</i> ; use <code>replace</code> to overwrite existing <i>filename</i>
Graph	
<u>graph</u> [(<i>graphopts</i>)]	graph results; see [PSS] power, graph
Iteration	
<u>init</u> (#)	initial value of the estimated parameter; default is method specific
<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>
[<u>no</u>] <u>log</u>	suppress or display iteration log
[<u>no</u>] <u>dots</u>	suppress or display iterations as dots
<u>notitle</u>	suppress the title

*Starred options may be specified either as one number or as a list of values; see [U] [11.1.8 numlist](#).
Options `n1()`, `n2()`, `nratio()`, and `compute()` are available only for two-independent-samples methods.
Iteration options are available only with computations requiring iteration.
`notitle` does not appear in the dialog box.

Description

The `power` command is useful for planning studies. It performs power and sample-size analysis for studies that use hypothesis testing to form inferences about population parameters. You can compute sample size given power and effect size, power given sample size and effect size, or the minimum detectable effect size and the corresponding target parameter given power and sample size. You can display results in a table ([PSS] [power, table](#)) and on a graph ([PSS] [power, graph](#)).

Options

Main

`alpha(numlist)` sets the significance level of the test. The default is `alpha(0.05)`.

`power(numlist)` sets the power of the test. The default is `power(0.8)`. If `beta()` is specified, this value is set to be $1 - \text{beta}()$. Only one of `power()` or `beta()` may be specified.

`beta(numlist)` sets the probability of a type II error of the test. The default is `beta(0.2)`. If `power()` is specified, this value is set to be $1 - \text{power}()$. Only one of `beta()` or `power()` may be specified.

`n(numlist)` specifies the total number of subjects in the study to be used for power or effect-size determination. If `n()` is specified, the power is computed. If `n()` and `power()` or `beta()` are specified, the minimum effect size that is likely to be detected in a study is computed.

`n1(numlist)` specifies the number of subjects in the control group to be used for power or effect-size determination.

`n2(numlist)` specifies the number of subjects in the experimental group to be used for power or effect-size determination.

`nratio(numlist)` specifies the sample-size ratio of the experimental group relative to the control group, N_2/N_1 for power or effect-size determination for two-sample tests. The default is `nratio(1)`, meaning equal allocation between the two groups.

`compute(n1 | n2)` requests that the `power` command compute one of the group sample sizes given the other one instead of the total sample size for two-sample tests. To compute the control-group sample size, you must specify `compute(n1)` and the experimental-group sample size in `n2()`. Alternatively, to compute the experimental-group sample size, you must specify `compute(n2)` and the control-group sample size in `n1()`.

`nfractional` specifies that fractional sample sizes be allowed. When this option is specified, fractional sample sizes are used in the intermediate computations and are also displayed in the output.

Also see the description and the use of options `n()`, `n1()`, `n2()`, `nratio()`, and `compute()` for two-sample tests in [PSS] [unbalanced designs](#).

`direction(upper | lower)` specifies the direction of the effect for effect-size determination. The default is `direction(upper)`, which means that the postulated value of the parameter is larger than the hypothesized value. `direction(lower)` means that the postulated value is lower than the hypothesized value.

`onesided` indicates a one-sided test. The default is two sided.

`parallel` reports results in parallel over the list of numbers supplied to command arguments and options allowing `numlist`. By default, results are computed over all combinations of the number lists. If the specified number lists are of different sizes, the last value in each of the shorter lists will be used in the remaining computations.

Table

`notable`, `table`, and `table()` control whether or not results are displayed in a tabular format. `table` is implied if any number list contains more than one element. `notable` is implied with graphical output—when either the `graph` or `graph()` option is specified. `table()` is used to produce custom tables. See [PSS] **power**, **table** for details.

`saving(filename [ , replace])` creates a Stata data file (.dta file) containing the table values with variable names corresponding to the displayed *columns*. `replace` specifies that *filename* be overwritten if it exists. `saving()` is only appropriate with tabular output.

Graph

`graph` and `graph()` produce graphical output; see [PSS] **power**, **graph** for details.

The following options control an iteration procedure used by the `power` command for solving nonlinear equations.

Iteration

`init(#)` specifies an initial value for the estimated parameter. Each `power` method sets its own default value. See the documentation entry of the method for details.

`iterate(#)` specifies the maximum number of iterations for the Newton method. The default is `iterate(500)`.

`tolerance(#)` specifies the tolerance used to determine whether successive parameter estimates have converged. The default is `tolerance(1e-12)`. See *Convergence criteria* in [M-5] **solvenl()** for details.

`ftolerance(#)` specifies the tolerance used 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.

`log` and `nolog` specify whether an iteration log is to be displayed. The iteration log is suppressed by default. Only one of `log`, `nolog`, `dots`, or `nodots` may be specified.

`dots` and `nodots` specify whether a dot is to be displayed for each iteration. The iteration dots are suppressed by default. Only one of `dots`, `nodots`, `log`, or `nolog` may be specified.

The following option is available with `power` but is not shown in the dialog box:

`notitle` prevents the command title from displaying.

Remarks and examples

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

Remarks are presented under the following headings:

Using the power command

Specifying multiple values of study parameters

One-sample tests

Two-sample tests

Paired-sample tests

Analysis of variance models

Tables of results

Power curves

This section describes how to perform power and sample-size analysis using the `power` command. For a software-free introduction to power and sample-size analysis, see [PSS] **intro**.

Using the power command

The `power` command computes sample size, power, or minimum detectable effect size and the corresponding target parameter for various hypothesis tests.

All computations are performed for a two-sided hypothesis test where, by default, the significance level is set to 0.05. You may change the significance level by specifying the `alpha()` option. You can specify the `onesided` option to request a one-sided test.

By default, the `power` command computes sample size for the default power of 0.8. You may change the value of power by specifying the `power()` option. Instead of power, you can specify the probability of a type II error in the `beta()` option.

To compute power, you must specify the sample size in the `n()` option.

To compute power or sample size, you must also specify a magnitude of the effect desired to be detected by a hypothesis test. `power`'s methods provide several ways in which an effect can be specified. For example, for a one-sample mean test, you can specify either the target mean or the difference between the target mean and a reference mean; see [PSS] [power onemean](#).

You can also compute the smallest magnitude of the effect or the minimum detectable effect size (MDES) and the corresponding target parameter that can be detected by a hypothesis test given power and sample size. To compute MDES, you must specify both the desired power in the `power()` option or the probability of a type II error in the `beta()` option and the sample size in the `n()` option. In addition to the effect size, `power` also reports the estimated value of the parameter of interest, such as the mean under the alternative hypothesis for a one-sample test or the experimental-group proportion for a two-sample test of independent proportions. By default, when the postulated value is larger than the hypothesized value, the `power` command assumes an effect in the upper direction, the `direction(upper)` option. You may request an estimate of the effect in the opposite, lower, direction by specifying the `direction(lower)` option.

For hypothesis tests comparing two independent samples, you can compute one of the group sizes given the other one instead of the total sample size. In this case, you must specify the label of the group size you want to compute in the `compute()` option and the value of the other group size in the respective `n#()` option. For example, if we wanted to find the size of the second group given the size of the first group, we would specify the combination of options `compute(n2)` and `n1(#)`.

A balanced design is assumed by default for two independent-sample tests, but you can request an unbalanced design. For example, you can specify the allocation ratio n_2/n_1 between the two groups in the `nratio()` option or the individual group sizes in the `n1()` and `n2()` options. See [PSS] [unbalanced designs](#) for more details about various ways of specifying an unbalanced design.

For sample-size determination, the reported integer sample sizes may not correspond exactly to the specified power because of rounding. To obtain conservative results, the `power` command rounds up the sample size to the nearest integer so that the corresponding power is at least as large as the requested power. You can specify the `nfractional` option to obtain the corresponding fractional sample size.

Some of `power`'s computations require iteration. The defaults chosen for the iteration procedure should be sufficient for most situations. In a rare situation when you may want to modify the defaults, the `power` command provides options to control the iteration procedure. The most commonly used is the `init()` option for supplying an initial value of the estimated parameter. This option can be useful in situations where the computations are sensitive to the initial values. If you are performing computations for a large number of combinations of various study parameters, you may consider reducing the default maximum number of iterations of 500 in the `iterate()` option so that the command is not spending time on calculations in difficult-to-compute regions of the parameter space. By default, `power` suppresses the iteration log. If desired, you can specify the `log` option to display

the `iteration` log or the `dots` option to display iterations as dots to monitor the progress of the iteration procedure.

The `power` command can produce results for one study scenario or for multiple study scenarios when multiple values of the parameters are specified; see [Specifying multiple values of study parameters](#) below for details.

For a single result, `power` displays results as text. For multiple results or if the `table` option is specified, `power` displays results in a table. You can also display multiple results on a graph by specifying the `graph` option. Graphical output suppresses the table of the results; use the `table` option to also see the tabular output. You can customize the default tables and graphs by specifying suboptions within the respective options `table()` and `graph()`; see [\[PSS\] power, table](#) and [\[PSS\] power, graph](#) for details.

You can also save the tabular output to a Stata dataset by using the `saving()` option.

Specifying multiple values of study parameters

The `power` command can produce results for one study scenario or for multiple study scenarios when multiple values of the parameters are supplied to the supported options. The options that support multiple values specified as a *numlist* are marked with a star in the syntax diagram.

For example, the `power()` option supports multiple values. You can specify multiple powers as individual values, `power(0.8 0.85 0.9)`, or as a range of values, `power(0.8(0.05)0.9)`; see [\[U\] 11.1.8 numlist](#) for other specifications.

In addition to options, you may specify multiple values of command arguments, values specified after the command name. For example, let $\#_1$ and $\#_2$ be the first and the second command arguments in

```
. power method  $\#_1$   $\#_2$ , ...
```

If we want to specify multiple values for the command arguments, we must enclose these values in parentheses. For example,

```
. power method (1 2) (1 2 3), ...
```

or, more generally,

```
. power method (numlist) (numlist), ...
```

When multiple values are specified in multiple options or for multiple command arguments, the `power` command computes results for all possible combinations formed by the values from every option and command argument. In some cases, you may want to compute results in parallel for specific sets of values of the specified parameters. To request this, you can specify the `parallel` option. If the specified number lists are of varying sizes, *numlist* with the maximum size determines the number of final results produced by `power`. The last value from *numlist* of smaller sizes will be used in the subsequent computations.

For example,

```
. power method (1 2), power(0.8 0.9)
```

is equivalent to

```
. power method 1, power(0.8)
. power method 2, power(0.8)
. power method 1, power(0.9)
. power method 2, power(0.9)
```

When the `parallel` option is specified,

```
. power method (1 2), power(0.8 0.9) parallel
```

is equivalent to

```
. power method 1, power(0.8)
. power method 2, power(0.9)
```

When the `parallel` option is specified and `numlist` is of different sizes, the last value of the shorter `numlist` is used in the subsequent computations. For example,

```
. power method (1 2 3), power(0.8 0.9) parallel
```

is equivalent to

```
. power method 1, power(0.8)
. power method 2, power(0.9)
. power method 3, power(0.9)
```

One-sample tests

The `power` command provides PSS computations for four one-sample tests. `power onemean` performs PSS analysis for a one-sample mean test; `power oneproportion` performs PSS analysis for a one-sample proportion test; `power onecorrelation` performs PSS analysis for a one-sample correlation test; and `power onevariance` performs PSS analysis for a one-sample variance test.

`power onemean` provides PSS computations for a one-sample t test assuming known or unknown population standard deviation. It also provides a way to adjust computations for a finite population sample. See [PSS] [power onemean](#).

`power oneproportion` provides PSS computations for a test that compares one proportion with a reference value. By default, the computations are based on a large-sample z test that uses the normal approximation of the distribution of the test statistic. You may choose between two large-sample tests: the score test or Wald test. You may also compute power for the small-sample binomial test by specifying the `test(binomial)` option. See [PSS] [power oneproportion](#).

`power onecorrelation` provides PSS computations for a test that compares one correlation with a reference value. The computations are based on a Fisher's z transformation of a correlation coefficient. See [PSS] [power onecorrelation](#).

`power onevariance` provides PSS computations for a test that compares one variance with a reference value. The computations are based on a χ^2 test of the ratio of the variance to its reference value. You can perform computations in the variance or standard deviation metric. See [PSS] [power onevariance](#).

All one-sample methods compute sample size given power and target parameter, power given sample size and target parameter, or MDES and the corresponding target parameter given power and sample size.

For PSS determination, an effect may be supplied by specifying the null and alternative values of the target parameter as command arguments `#0` and `#a`:

```
. power onesample #0 #a, ...
```

Instead of the alternative value `#a`, you can specify the ratio of the alternative value to the null value in the `ratio()` option and the null value as `#0` for `power onevariance`,

```
. power onevariance #0, ratio(#) ...
```

or you can specify the difference between the alternative value and the null value in the `diff()` option and the null value as `#0` for other methods,

```
. power onesample #0, diff(#) ...
```

For sample-size determination, the reported sample size is rounded up to the nearest integer. This ensures that the corresponding actual power is at least as large as the specified power. You can specify the `nfractional` option to obtain the corresponding fractional sample size, or you can recompute the actual power using the reported rounded value; see [Fractional sample sizes](#) in [PSS] **unbalanced designs** for details.

Below we show a quick example of PSS analysis for a one-sample mean test. See entries of the one-sample methods for more examples.

► Example 1: PSS analysis for a one-sample mean test

A group of pediatricians would like to study the exposure of infants to television. The group wants to investigate whether the average number of hours watched per day by infants between 3 and 12 months of age is greater than 2 hours. Before conducting a study, pediatricians would like to determine how many infants they need to enroll in the study. The analysis will use the one-sample *t* test to compare the mean of the obtained sample with the reference value. An earlier pilot study reported an average of 2.5 hours watched per day with a standard deviation of 0.8. Pediatricians would like to compute the sample size required to detect a mean of 2.5 using a two-sided test with 5% significance level and 80% power. Although pediatricians suspect that the effect is in the upper direction—more than two hours watched on average—they prefer to obtain the required sample size for a two-sided test instead of a one-sided test.

We use `power onemean` to compute the required sample size. We specify the reference or null value of 2 and the comparison or alternative value of 2.5 as command arguments. We also specify the standard deviation of 0.8 in the `sd()` option. We omit the `alpha(0.05)` and `power(0.8)` options because the desired values are the defaults for these options. The default test is two sided, so we do not need to supply any additional information to the command.

```
. power onemean 2 2.5, sd(0.8)
Performing iteration ...
Estimated sample size for a one-sample mean test
t test
Ho: m = m0 versus Ha: m != m0
Study parameters:
      alpha =    0.0500
      power =    0.8000
      delta =    0.6250
      m0 =     2.0000
      ma =     2.5000
      sd =     0.8000

Estimated sample size:
      N =          23
```

All `power` commands have a similar output format. Information about the test and tested hypothesis is displayed first. The input and implied values of the study parameters are displayed next under **Study parameters**. The estimated parameters, such as the sample size in this example, are displayed last.

Pediatricians need to enroll 23 infants in the study to detect a standardized difference of 0.625 between the alternative mean of 2.5 and the null mean of 2 given a standard deviation of 0.8 using a 5%-level two-sided one-sample *t* test with 80% power.

The pediatricians believe that they have resources to enroll more infants. They wish to compute the power that corresponds to the sample size of 50. To compute the corresponding power, we specify a sample size of 50 in the `n()` option:

```
. power onemean 2 2.5, sd(0.8) n(50)
Estimated power for a one-sample mean test
t test
Ho: m = m0 versus Ha: m != m0
Study parameters:
      alpha =    0.0500
      N      =     50
      delta =    0.6250
      m0     =    2.0000
      ma     =    2.5000
      sd     =    0.8000
Estimated power:
      power =    0.9911
```

The power increases to 99% for a larger sample of 50 infants.

The pediatricians also want to find out what is the smallest mean difference they can detect with the larger sample of 50 infants while keeping the power at 80%. They assume the effect to be in the upper direction for this computation. To compute the minimum detectable difference, we specify both the sample size in the `n()` option and the power in the `power()` option.

```
. power onemean 2, sd(0.8) n(50) power(0.8)
Performing iteration ...
Estimated target mean for a one-sample mean test
t test
Ho: m = m0 versus Ha: m != m0; ma > m0
Study parameters:
      alpha =    0.0500
      power =    0.8000
      N      =     50
      m0     =    2.0000
      sd     =    0.8000
Estimated effect size and target mean:
      delta =    0.4042
      ma     =    2.3233
```

The smallest standardized difference that can be detected given the study parameters is about 0.4, with a corresponding mean of 2.32.

◀

Two-sample tests

The `power` command provides PSS computations for four two-sample tests. `power twomeans` performs PSS analysis for a two-sample means test; `power twoproportions` performs PSS analysis for a two-sample proportions test; `power twocorrelations` performs PSS analysis for a two-sample correlations test; and `power twovariances` performs PSS analysis for a two-sample variances test.

`power twomeans` provides PSS computations for a two-sample means test that compares the means of two independent populations. The computations provided assume known or unknown and equal or unequal population standard deviations of the two groups. See [PSS] [power twomeans](#).

`power twoproportions` provides PSS computations for a two-sample proportions test that compares the proportions in two independent populations with binary outcomes. Three tests are supported: the large-sample Pearson's χ^2 test, the large-sample likelihood-ratio test, and the small-sample Fisher's exact test. Several effect specifications are available. For example, you can specify the effect of interest as a risk difference, or a relative risk, or an odds ratio. See [PSS] [power twoproportions](#).

`power twocorrelations` provides PSS computations for a two-sample correlations test that compares the correlation coefficients of two independent populations. The computations are based on a Fisher's z transformation of a correlation coefficient. See [PSS] [power twocorrelations](#).

`power twovariances` provides PSS computations for a two-sample variances test that compares the variances of two independent populations. The computations are based on an F test of the ratio of variances. You can perform computations in the variance or standard deviation metric. See [PSS] [power twovariances](#).

All two-sample methods compute sample size given power and the control-group and experimental-group values of the target parameter, power given sample size and the control-group and experimental-group values of the target parameter, or MDSES and the corresponding target value of the parameter in the experimental group given power, sample size, and the control-group parameter value.

To compute sample size or power, you can specify the magnitude of the effect of interest in two ways: by directly specifying the alternative values of the target parameter in two groups or by specifying the control-group alternative value and the corresponding relation of the experimental-group value to the control-group alternative value.

The two alternative values are specified as command arguments: the alternative value of the target parameter in the control or reference group, $\#_{a1}$, and the alternative value of the target parameter in the experimental or comparison group, $\#_{a2}$:

```
. power twosample #a1 #a2, ...
```

The experimental-group alternative value, $\#_{a2}$, may be omitted if an option containing the relationship between the two alternative values is specified. For example, for `power twomeans` and `power twocorrelations`, such an option is `diff()`, and it specifies the difference between the experimental-group and control-group alternative values:

```
. power twomeans #a1, diff(#) ...
```

For `power twovariances`, such an option is `ratio()`, and it contains the ratio of the experimental-group alternative value to the control-group value:

```
. power twovariances #a1, ratio(#) ...
```

`power twoproportions` provides several alternative specifications in which a difference between the two populations may be expressed. For example, you can express the “difference” as an odds ratio of the experimental group to the control group,

```
. power twoproportions #a1, oratio(#) ...
```

or as a relative risk,

```
. power twoproportions #a1, rrisk() ...
```

In addition to the total sample size, two-sample methods provide a way to solve for one of the group sizes when the other group size is fixed. This can be achieved by specifying the `compute()` option. To compute the size of the first group, you must specify the `compute(n1)` option and the size of the second group in the `n2()` option. To compute the size of the second group, you must specify the `compute(n2)` option and the size of the first group in the `n1()` option.

To compute power, you can specify a total sample size in the `n()` option, group sample sizes in the `n1()` and `n2()` options, or one of the group sample sizes and its ratio, n_2/n_1 , in the `nratio()` option; see [\[PSS\] unbalanced designs](#) for more specifications.

Below we show a quick example of PSS analysis for a two-sample means test. See entries of the two-sample methods for more examples.

► Example 2: PSS analysis for a two-sample mean test

A pharmaceutical company would like to conduct a study to compare a new weight-loss drug with an older drug. Investigators are planning to use a two-sample t test to compare the average weight loss for the two drugs. The average weight loss of people taking the old drug for 3 months is 12 pounds, with a standard deviation of 5.5 pounds. The new drug is expected to produce an average weight loss of 16 pounds, with a standard deviation of 5 pounds for the same period of time. Investigators want to find out how many subjects they need to recruit into the study to detect the specified difference using a 5% level two-sided test with 90% power.

We use `power twomeans` to perform PSS analyses. We specify the control-group mean 12 and the experimental-group mean 16 as command arguments after the command name. We specify the respective standard deviations in the `sd1()` and `sd2()` options. The default power is set to 0.8, so we specify `power(0.9)` to request 90% power.

```
. power twomeans 12 16, sd1(5.5) sd2(5) power(0.9)
Performing iteration ...
Estimated sample sizes for a two-sample means test
Satterthwaite's t test assuming unequal variances
Ho: m2 = m1 versus Ha: m2 != m1
Study parameters:
      alpha =    0.0500
      power =    0.9000
      delta =    4.0000
      m1 =    12.0000
      m2 =    16.0000
      sd1 =    5.5000
      sd2 =    5.0000
Estimated sample sizes:
      N =          76
      N per group =    38
```

We need a sample of 76 subjects, 38 per group, to detect a difference of 4 between the control-group mean of 12 and the experimental-group mean of 16 given the respective standard deviations of 5.5 and 5 with 90% power using a 5%-level two-sided two-sample means t test.

The default test is two sided. You may specify the `onesided` option to request a one-sided test. The default design is also balanced; see [\[PSS\] unbalanced designs](#) for examples of unbalanced designs.

The investigators hope to keep the sample size under 60 and would like to compute the power corresponding to this sample size. To compute the corresponding power, we specify the `n(60)` option instead of the `power()` option:

```
. power twomeans 12 16, sd1(5.5) sd2(5) n(60)
Estimated power for a two-sample means test
Satterthwaite's t test assuming unequal variances
Ho: m2 = m1 versus Ha: m2 != m1
Study parameters:
      alpha =    0.0500
      N =      60
N per group =    30
      delta =    4.0000
      m1 =    12.0000
      m2 =    16.0000
      sd1 =    5.5000
      sd2 =    5.0000
Estimated power:
      power =    0.8259
```

The power decreases to 83% for the smaller sample of 60 subjects.

To keep the power at 90%, the investigators want to compute the smallest difference between the experimental-group mean and the control-group mean (in the upper direction) given the sample of 60 subjects. For this computation, we specify both options `n(60)` and `power(0.9)`:

```
. power twomeans 12, sd1(5.5) sd2(5) n(60) power(0.9)
Performing iteration ...
Estimated experimental-group mean for a two-sample means test
Satterthwaite's t test assuming unequal variances
Ho: m2 = m1 versus Ha: m2 != m1; m2 > m1
Study parameters:
      alpha =    0.0500
      power =    0.9000
      N =      60
N per group =    30
      m1 =    12.0000
      sd1 =    5.5000
      sd2 =    5.0000
Estimated effect size and experimental-group mean:
      delta =    4.4744
      m2 =    16.4744
```

The smallest detectable mean difference is 4.47, with a corresponding value of the experimental-group mean of 16.47.



Paired-sample tests

The `power` command provides PSS computations for two tests of paired samples. `power pairedmeans` performs PSS analysis for a two-sample paired-means test, and `power pairedproportions` performs PSS analysis for a two-sample paired-proportions test.

`power pairedmeans` provides PSS computations for a two-sample paired *t* test assuming known or unknown population standard deviation of the differences between paired observations. You can specify standard deviations of each group and a correlation between paired observations, or you can directly specify the standard deviation of the differences between observations. You can obtain results for a nonzero null hypothesis of a difference between the two paired means. The command also provides a way to adjust computations for a finite population sample. See [PSS] [power pairedmeans](#).

`power pairedproportions` provides PSS computations for a two-sample paired-proportions test that compares proportions in two paired (correlated) samples. The computations are based on McNemar's test of marginal homogeneity. You can specify either the discordant proportions or the marginal proportions. A number of effect specifications are available. For example, you can specify the effect of interest as a relative risk or an odds ratio. See [PSS] [power pairedproportions](#).

Both paired methods compute sample size given power and target parameter, power given sample size and target parameter, or MDES and the corresponding target parameter given power and sample size.

For power and sample-size determination of `power pairedmeans`, an effect may be supplied by specifying the alternative values of the two means, pretreatment and posttreatment, as command arguments m_{a1} and m_{a2} :

```
power pairedmeans  $m_{a1}$   $m_{a2}$ , ...
```

Instead of the alternative value m_{a2} , you can specify the difference between the two alternative values in the `altdiff()` option and the alternative pretreatment mean value m_{a1} :

```
power pairedmeans  $m_{a1}$ , altdiff() ...
```

You may omit both alternative values and specify only the difference between them in the `altdiff()` option:

```
power pairedmeans, altdiff() ...
```

By default, the null value of the difference between the pretreatment and posttreatment means is zero, but you may change it by specifying the `nulldiff()` option.

For PSS determination of `power pairedproportions`, there are a number of ways of specifying an effect of interest; see [Alternative ways of specifying effect](#) in [PSS] [power pairedproportions](#). Two main specifications include the specification of discordant proportions and the specification of marginal probabilities. Specifically, you can supply the information about the effect of interest as discordant proportions p_{12} and p_{21} ,

```
power pairedproportions  $p_{12}$   $p_{21}$ , ...
```

or as marginal proportions p_{1+} and p_{+1} :

```
power pairedproportions  $p_{1+}$   $p_{+1}$ , corr(numlist) ...
```

When you specify marginal proportions, you must also specify the correlation between paired observations in the `corr()` option.

For sample-size determination, the reported sample size is rounded up to the nearest integer. This ensures that the corresponding actual power is at least as large as the specified power. You can specify the `nfractional` option to obtain the corresponding fractional sample size or you can recompute the actual power using the reported rounded value; see [Fractional sample sizes](#) in [PSS] [unbalanced designs](#) for details.

Below we show a quick example of PSS analyses for a two-sample paired-means test. See [PSS] [power pairedmeans](#) and [PSS] [power pairedproportions](#) for more examples.

► Example 3: PSS analysis for a two-sample paired-means test

A forester would like to study the effects of a fertilizer treatment on heights of Virginia pine trees. The trees are planted in pairs with only one of them receiving the fertilizer treatment. The average height of untreated trees is 27.5 feet, with a standard deviation of 4.5 feet. The fertilizer treatment is expected to increase the average height to 30 feet, with the same standard deviation of 4.5 feet. The

correlation between the paired tree heights is expected to be 0.4. The forester would like to know how many pairs of pine trees need to be planted so that a 5%-level two-sided paired-means t test detects the anticipated difference with 80% power.

We use `power pairedmeans` for power and sample-size analysis. We supply the alternative pretreatment and posttreatment means of 27.5 and 30, respectively, as command arguments after the command name. The standard deviations of the two groups are the same, so we specify their common value in the `sd()` option. We specify the correlation of 0.4 in the `corr()` option. The default value for power is 0.8 and for significance level is 0.05, so we omit the corresponding options `power(0.8)` and `alpha(0.05)`.

```
. power pairedmeans 27.5 30, sd(4.5) corr(0.4)
Performing iteration ...
Estimated sample size for a two-sample paired-means test
Paired t test assuming sd1 = sd2 = sd
Ho: d = d0 versus Ha: d != d0
Study parameters:
      alpha =    0.0500          ma1 =   27.5000
      power =    0.8000          ma2 =   30.0000
      delta =    0.5072          sd  =    4.5000
      d0 =     0.0000          corr =    0.4000
      da =     2.5000
      sd_d =    4.9295
Estimated sample size:
      N =          33
```

The forester needs 33 pairs of pine trees to run the experiment.

The forester has resources to plant more trees and would like to compute the power corresponding to the larger sample. To compute power given sample size, we specify sample size in the `n()` option:

```
. power pairedmeans 27.5 30, sd(4.5) corr(0.4) n(50)
Estimated power for a two-sample paired-means test
Paired t test assuming sd1 = sd2 = sd
Ho: d = d0 versus Ha: d != d0
Study parameters:
      alpha =    0.0500          ma1 =   27.5000
      N =         50          ma2 =   30.0000
      delta =    0.5072          sd  =    4.5000
      d0 =     0.0000          corr =    0.4000
      da =     2.5000
      sd_d =    4.9295
Estimated power:
      power =    0.9400
```

The power increases to 0.94.

The forester may also wish to know the smallest detectable difference between average tree heights of the fertilized group and of the control group that can be detected with 80% power and sample size of 50. To compute this value, we specify both options `n(50)` and `power(0.8)`:

```
. power pairedmeans 27.5, sd(4.5) corr(0.4) n(50) power(0.8)
Performing iteration ...
Estimated target parameters for a two-sample paired-means test
Paired t test assuming sd1 = sd2 = sd
Ho: d = d0 versus Ha: d != d0; da > d0
Study parameters:
      alpha =    0.0500          ma1 =   27.5000
      power =    0.8000          sd  =    4.5000
      N      =     50          corr =    0.4000
      d0     =    0.0000
      sd_d   =    4.9295
Estimated effect size and target parameters:
      delta =    0.4042
      da    =    1.9924
      ma2   =   29.4924
```

The smallest detectable difference is 1.99, with a corresponding value of the average tree height for the fertilized trees of 29.5.



Analysis of variance models

The `power` command provides PSS computations for three types of analysis of variance (ANOVA) designs: one way, two way, and repeated measures. `power oneway` performs PSS analysis for a one-way ANOVA. `power twoway` performs PSS analysis for a two-way ANOVA. `power repeated` performs PSS analysis for a repeated-measures ANOVA.

`power oneway` provides PSS computations for a one-way ANOVA model. You can choose between the overall F test of the equality of group means and a test of a mean contrast. You can either specify group means or specify their variability in the computations. See [PSS] [power oneway](#).

`power twoway` provides PSS computations for a two-way fixed-effects ANOVA model. You can choose the overall F test of the main effect of a row factor, a column factor, or a row-by-column interaction. You can either specify cell means or specify the variance explained by the tested effect. See [PSS] [power twoway](#).

`power repeated` provides PSS computations for one-way and two-way fixed-effects repeated-measures ANOVA models. You can choose the overall F test of the main effect of a between-subjects factor, a within-subject factor, or a between-within factor interaction. You can either specify cell means or specify the variance explained by the tested effect. See [PSS] [power repeated](#).

All methods compute sample size given power and effect size, power given sample size and effect size, or effect size given power and sample size.

For power and sample-size determination of `power oneway`, an effect may be supplied by specifying the alternative values of group means as command arguments m_{a1} , m_{a2} , m_{a3} , and so on:

```
power oneway m_{a1} m_{a2} [ m_{a3} ... ], ...
```

Instead of the alternative group means, you can specify the variance of the group means in the `varmeans()` option and the number of groups in the `ngroups()` option:

```
power oneway, ngroups() varmeans() ...
```

For power and sample-size determination of `power twoway` and `power repeated`, an effect may be supplied by specifying the alternative values of cell means as command arguments $m_{a1,1}$, $m_{a1,2}$, and so on, in a matrix form:

```
power twoway ma1,1 ma1,2 [...] \ ma2,1 ma2,2 [...], ...
power repeated ma1,1 ma1,2 [...] [\ ma2,1 ma2,2 [...]], ...
```

Instead of the alternative cell means, you can specify the variance of the tested effect in the `vareffect()` option and the dimensions of the cell-means matrix: number of rows and columns for `power twoway` and number of groups and repeated measures for `power repeated`:

```
power twoway, nrows() ncols() factor() vareffect() ...
power repeated, ngroups() nrepeated() factor() vareffect() ...
```

The means can also be supplied as a matrix at the command line. For example, suppose that we have three groups.

```
power oneway ma1 ma2 ma3, ...
```

The above command would be equivalent to

```
matrix means = (ma1, ma2, ma3)
power oneway means, ...
```

There are also other alternative specifications of an effect with these commands. See the specific entry of each command.

For sample-size determination, the reported sample size is rounded up to the nearest integer. This ensures that the corresponding actual power is at least as large as the specified power. You can specify the `nfractional` option to obtain the corresponding fractional sample size, or you can recompute the actual power using the reported rounded value; see [Fractional sample sizes](#) in [PSS] **unbalanced designs** for details.

Below we show a quick example of PSS analysis for a one-way ANOVA model. See [PSS] **power oneway**, [PSS] **power twoway**, and [PSS] **power repeated** for more examples.

► Example 4: PSS analysis for a one-way ANOVA model

Researchers would like to compare the effects of four drugs on systolic blood pressure. They would like to use a one-way ANOVA model to test the equality of mean blood-pressure measurements across four drugs. To conduct a study, the researchers need an estimate for the number of subjects to be enrolled in a study. From a previous pilot study, the variance between group means was estimated to be 57, and the error variance was estimated to be 115. The researchers would like to compute the required sample size to detect the effect size of $0.7040 = \sqrt{57/115}$ with 80% power using a 5%-level F test of the equality of means assuming a balanced design.

We use `power oneway` to compute the sample size. We specify the number of groups and the estimates of variances in the corresponding options. The default value for power is 0.8 and for significance level is 0.05, so we omit the corresponding options `power(0.8)` and `alpha(0.05)`.

```
. power oneway, ngroups(4) varmeans(57) varerror(115)
Estimated sample size for one-way ANOVA
F test for group effect
Ho: delta = 0 versus Ha: delta != 0
Study parameters:
      alpha =    0.0500
      power =    0.8000
      delta =    0.7040
      N_g =      4
      Var_m =   57.0000
      Var_e =  115.0000
Estimated sample sizes:
      N =      28
      N per group =    7
```

The researchers need to recruit 28 subjects, 7 subjects per group, for this study.

Unfortunately, the researchers can afford to recruit only 20 subjects. They wish to compute the power corresponding to this smaller sample size. To compute power, we additionally specify sample size in the `n()` option:

```
. power oneway, n(20) ngroups(4) varmeans(57) varerror(115)
Estimated power for one-way ANOVA
F test for group effect
Ho: delta = 0 versus Ha: delta != 0
Study parameters:
      alpha =    0.0500
      N =      20
      N per group =    5
      delta =    0.7040
      N_g =      4
      Var_m =   57.0000
      Var_e =  115.0000
Estimated power:
      power =    0.6400
```

The power decreases to 0.64.

The researchers are not satisfied with such a low power. They now would like to compute the smallest effect size and the corresponding variance of means that can be detected with the power of 80% and the sample size of 20. To compute effect size, we specify both power and sample size in respective options:

```
. power oneway, n(20) power(0.8) ngroups(4) varerror(115)
Performing iteration ...
Estimated between-group variance for one-way ANOVA
F test for group effect
Ho: delta = 0 versus Ha: delta != 0
Study parameters:
    alpha =    0.0500
    power =    0.8000
      N =      20
N per group =      5
  N_g =      4
Var_e = 115.0000
Estimated effect size and between-group variance:
    delta =    0.8353
    Var_m =    80.2329
```

The smallest detectable effect size is 0.8353, with a corresponding value of the between-group variance of 80.2329.



Tables of results

When power is used to perform computations for a single set of study parameters, the results can be displayed either as text or in a table. The default is to display results as text:

```
. power onemean 0 0.2
Performing iteration ...
Estimated sample size for a one-sample mean test
t test
Ho: m = m0 versus Ha: m != m0
Study parameters:
    alpha =    0.0500
    power =    0.8000
    delta =    0.2000
    m0 =    0.0000
    ma =    0.2000
    sd =    1.0000
Estimated sample size:
      N =      199
```

You can specify the `table` option to display results in a table:

```
. power onemean 0 0.2, table
Performing iteration ...
Estimated sample size for a one-sample mean test
t test
Ho: m = m0 versus Ha: m != m0
```

alpha	power	N	delta	m0	ma	sd
.05	.8	199	.2	0	.2	1

For multiple sets of study parameters, when command arguments or options contain number lists, the results are automatically displayed in a table:

```
. power onemean 0 (0.2 0.5)
Performing iteration ...
Estimated sample size for a one-sample mean test
t test
Ho: m = m0 versus Ha: m != m0
```

alpha	power	N	delta	m0	ma	sd
.05	.8	199	.2	0	.2	1
.05	.8	34	.5	0	.5	1

In this example we specified two values for the second argument.

Default tables can be modified by specifying the `table()` option. For example, we can change the order in which the columns are displayed:

```
. power onemean 0 (0.2 0.5), table(alpha power N m0 ma sd delta)
Performing iteration ...
Estimated sample size for a one-sample mean test
t test
Ho: m = m0 versus Ha: m != m0
```

alpha	power	N	m0	ma	sd	delta
.05	.8	199	0	.2	1	.2
.05	.8	34	0	.5	1	.5

Or we can change column labels:

```
. power onemean 0 (0.2 0.5), table(, labels(N "Sample size"))
Performing iteration ...
Estimated sample size for a one-sample mean test
t test
Ho: m = m0 versus Ha: m != m0
```

alpha	power	Sample size	delta	m0	ma	sd
.05	.8	199	.2	0	.2	1
.05	.8	34	.5	0	.5	1

Or we can select which columns we want to display:

```
. power onemean 0 (0.2 0.5), table(alpha beta N m0 ma sd)
Performing iteration ...
Estimated sample size for a one-sample mean test
t test
Ho: m = m0 versus Ha: m != m0
```

alpha	beta	N	m0	ma	sd
.05	.2	199	0	.2	1
.05	.2	34	0	.5	1

For more examples, see [\[PSS\] power, table](#).

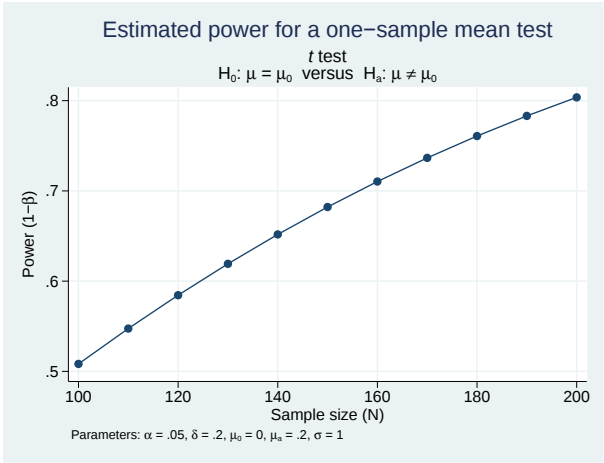
Power curves

During the planning stage of a study, it is difficult to decide on a number of subjects to be enrolled in a study on the basis of only one set of study parameters. It is common to investigate the effect of various study scenarios on power. Power curves, or plots of estimated power versus one of the study parameters, are commonly used for this purpose.

The `power` command provides the `graph` and `graph()` options to produce power curves. More precisely, when `graph` is specified, the estimated parameter such as power or sample size is plotted on the y axis, and the varying parameter is plotted on the x axis.

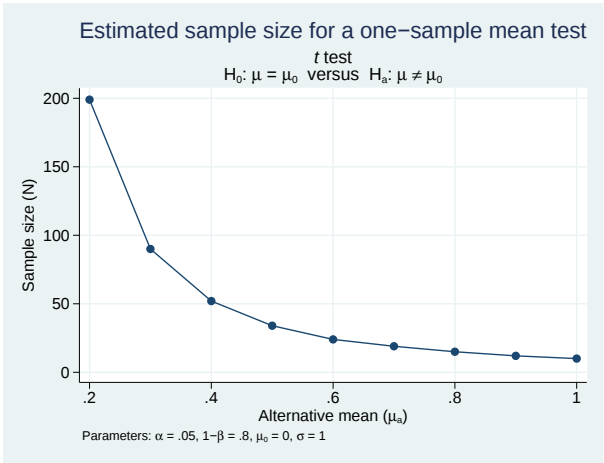
For example, we compute power and plot it as a function of sample size for a range of sample-size values between 100 and 200 with a step size of 10:

```
. power onemean 0 0.2, n(100(10)200) graph
```



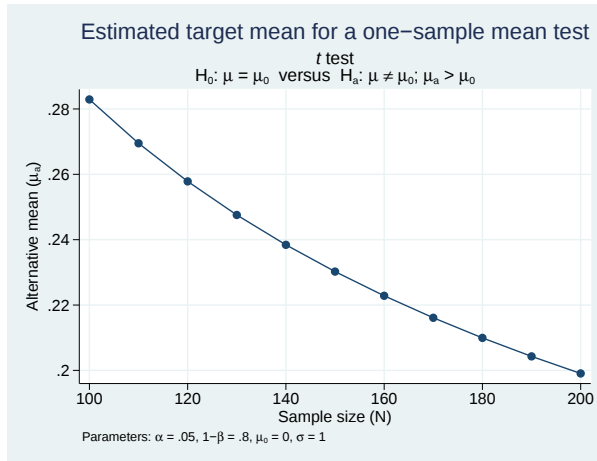
Or we can compute sample size and plot it as a function of the alternative mean when the mean ranges between 0.2 and 1 with a step size of 0.1:

```
. power onemean 0 (0.2(0.1)1), graph
```



Or we can compute the alternative mean for a given power of 80% and a range of sample-size values between 100 and 200 with a step size of 10, and plot it against the sample size:

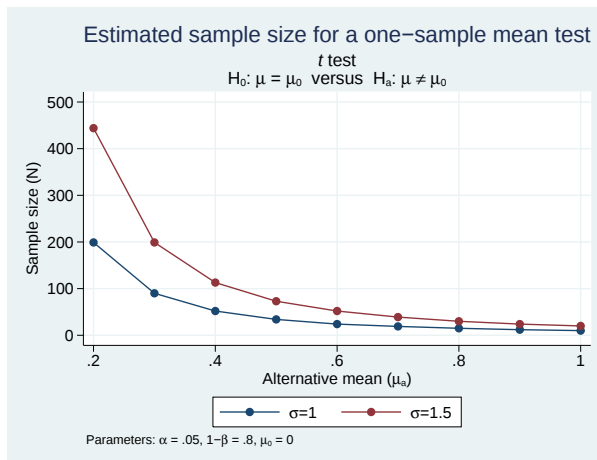
```
. power onemean 0, n(100(10)200) power(0.8) graph
```



The above graphs are the default graphs produced by `power`, `graph`. Similarly to tabular output, you can customize graphical output by specifying the `graph()` option.

For example, we modify the look of the default graph by using the `graph(nosimplelabels legend(title(""))) option`. `nosimplelabels` requests that the graph legend include the column symbol and an equal sign; `legend(title(""))` requests that the legend not have a title.

```
. power onemean 0 (0.2(0.1)1), sd(1 1.5) graph(nosimplelabels legend(title(""))) graph
```



By default, when a graph is produced, the tabular output is suppressed. You can specify the `table` option if you also want to see results in a table.

For more examples, see [\[PSS\] power, graph](#).

Stored results

`power` stores the following in `r()`:

Scalars

<code>r(alpha)</code>	significance level
<code>r(power)</code>	power
<code>r(beta)</code>	probability of a type II error
<code>r(delta)</code>	effect size
<code>r(N)</code>	total sample size
<code>r(N_a)</code>	actual sample size
<code>r(N1)</code>	sample size of the control group
<code>r(N2)</code>	sample size of the experimental group
<code>r(nratio)</code>	ratio of sample sizes, N_2/N_1
<code>r(nratio_a)</code>	actual ratio of sample sizes
<code>r(nfractional)</code>	1 if <code>nfractional</code> is specified; 0 otherwise
<code>r(onesided)</code>	1 for a one-sided test; 0 otherwise
<code>r(separator)</code>	number of lines between separator lines in the table
<code>r(divider)</code>	1 if <code>divider</code> is requested in the table; 0 otherwise
<code>r(init)</code>	initial value of the estimated parameter
<code>r(maxiter)</code>	maximum number of iterations
<code>r(iter)</code>	number of iterations performed
<code>r(tolerance)</code>	requested parameter tolerance
<code>r(deltax)</code>	final parameter tolerance achieved
<code>r(ftolerance)</code>	requested distance of the objective function from zero
<code>r(function)</code>	final distance of the objective function from zero
<code>r(converged)</code>	1 if iteration algorithm converged; 0 otherwise

Macros

<code>r(type)</code>	<code>test</code>
<code>r(method)</code>	the name of the specified method
<code>r(direction)</code>	upper or lower
<code>r(columns)</code>	displayed table columns
<code>r(labels)</code>	table column labels
<code>r(widths)</code>	table column widths
<code>r(formats)</code>	table column formats

Matrix

<code>r(pss_table)</code>	table of results
---------------------------	------------------

Also see *Stored results* in the method-specific manual entries for additional stored results.

Methods and formulas

By default, the `power` command rounds sample sizes to integers and uses integer values in the computations. To ensure conservative results, the commands round down the input sample sizes and rounds up the output sample sizes. See *Fractional sample sizes* in [PSS] **unbalanced designs** for details.

Some of `power`'s methods require iteration. For example, the sample size for a two-sided test is usually solved iteratively from the two-sided power equation. Most methods use Mata function `solvenl()` and its Newton's method described in *Newton-type methods* in [M-5] `solvenl()` to solve a nonlinear power equation. Other methods use a bisection method to find a root of a nonlinear power equation.

See *Methods and formulas* in the method-specific manual entries for details.

Reference

Batistatou, E., C. Roberts, and S. Roberts. 2014. [Sample size and power calculations for trials and quasi-experimental studies with clustering](#). *Stata Journal* 14: 159–175.

Also see

[PSS] [intro](#) — Introduction to power and sample-size analysis

[PSS] [Glossary](#)

[ST] [stpower](#) — Sample size, power, and effect size for survival analysis