Title

stata.com

ciwidth — Precision and sample-size analysis for CIs

Description

The `ciwidth` command performs precision and sample-size analysis (PrSS) for CIs. You can compute sample size given CI width (or precision) and probability of CI width. Alternatively, you can compute CI width given sample size and probability of CI width. You can also compute probability of CI width given sample size and CI width. You can display results in a table ([PSS-3] **ciwidth, table**) and on a graph ([PSS-3] **ciwidth, graph**).

For power and sample-size analysis for hypothesis tests, see [PSS-2] **power**.

Menu

Statistics > Power, precision, and sample size

Syntax

Compute sample size

```
ciwidth method ..., width(numlist) probwidth(numlist) [ ciwidth_options ]
```

Compute CI width

```
ciwidth method ..., probwidth(numlist) n(numlist) [ ciwidth_options ]
```

Compute probability of CI width

```
ciwidth method ..., width(numlist) n(numlist) [ ciwidth_options ]
```

method	Description
One sample	
`onemean`	CI for one mean
`onevariance`	CI for one variance
Two independent samples	
`twomeans`	CI for comparing two means from independent samples
Two paired samples	
`pairedmeans`	CI for comparing two means from paired samples
User-defined methods	
usermethod	Add your own method to `ciwidth`

ciwidth_options	Description
Main	
*level(*numlist*)	confidence level; default is level(95)
*alpha(*numlist*)	significance level; default is alpha(0.05)
*probwidth(*numlist*)	probability of CI width; required to compute sample size and CI width
*width(*numlist*)	CI width; required to compute sample size and probability of CI width
*n(*numlist*)	total sample size; required to compute CI width and probability of CI width
*n1(*numlist*)	sample size of the control group
*n2(*numlist*)	sample size of the experimental group
*nratio(*numlist*)	ratio of sample sizes, N2/N1; default is nratio(1), meaning equal group sizes
compute(N1 | N2)	solve for N1 given N2 or for N2 given N1
nfractional	allow fractional sample sizes
lower	lower one-sided CI; default is two-sided CI
upper	upper one-sided CI; default is two-sided CI
onesided	synonym for option upper
parallel	treat number lists in starred options or in command arguments as parallel when multiple values per option or argument are specified (do not enumerate all possible combinations of values)
Table	
[no]table[(*tablespec*)]	suppress table or display results as a table; see [PSS-3] **ciwidth, table**
saving(*filename* [, replace])	save the table data to *filename*; use replace to overwrite existing *filename*
Graph	
graph[(*graphopts*)]	graph results; see [PSS-3] **ciwidth, graph**
Iteration	
init(#)	initial value for sample size; default is to use a closed-form normal approximation
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)
[no]log	suppress or display iteration log
[no]dots	suppress or display iterations as dots
notitle	suppress the title

*Specifying a list of values in at least two starred options, or at least two command arguments, or at least one starred option and one argument results in computations for all possible combinations of the values; see [U] **11.1.8 numlist**. Also see the parallel option.

Options n1(), n2(), nratio(), and compute() are available only for two-independent-samples methods.

Iteration options are available only with computations requiring iteration.

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

notitle does not appear in the dialog box.

Options

Main

`level(`*numlist*`)` specifies the confidence level, as a percentage, for CIs. The default is `level(95)` or as set by `set level`; see [R] **level**. If `alpha()` is specified, this value is set to be $100(1-$`alpha()`$)$. Only one of `level()` or `alpha()` may be specified.

`alpha(`*numlist*`)` sets the significance level. Only one of `level()` or `alpha()` may be specified.

`probwidth(`*numlist*`)` specifies the probability of obtaining a CI with the width no larger than a target CI width. The target CI width is either computed by the command or specified in option `width()`. This option is required to compute sample size and CI width.

`width(`*numlist*`)` specifies the target CI width, which represents the precision of the CI. This option is required to compute sample size and probability of CI width. For a two-sided CI, CI width is the distance between the upper and lower limits. For a one-sided CI, it is the distance from the limit to the estimate of the parameter of interest, such as a sample mean.

`n(`*numlist*`)` specifies the total number of subjects in the study to be used to compute the CI width and probability of CI width.

`n1(`*numlist*`)` specifies the number of subjects in the control group to be used to compute the CI width and probability of CI width.

`n2(`*numlist*`)` specifies the number of subjects in the experimental group to be used to compute the CI width and probability of CI width.

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

`compute(N1 | N2)` requests that the `ciwidth` command compute one of the group sample sizes given the other one, instead of the total sample size, for two-sample CIs. 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 `nfractional` for two-sample CIs in *Fractional sample sizes* of [PSS-4] **Unbalanced designs**.

`lower` specifies a lower one-sided CI and may not be combined with option `upper`. The default is a two-sided CI.

`upper` specifies an upper one-sided CI and may not be combined with option `lower`. The default is a two-sided CI.

`onesided` is a synonym for `upper`, which specifies an upper one-sided CI.

`parallel` requests that computations be performed in parallel over the lists of numbers specified for at least two study parameters as command arguments, starred options allowing *numlist*, or both. That is, when `parallel` is specified, the first computation uses the first value from each list of numbers, the second computation uses the second value, and so on. 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. By default, results are computed over all combinations of the number lists.

For example, let a_1 and a_2 be the list of values for one study parameter, and let b_1 and b_2 be the list of values for another study parameter. By default, `ciwidth` will compute results for all

possible combinations of the two values in the two study parameters: (a_1, b_1), (a_1, b_2), (a_2, b_1), and (a_2, b_2). If `parallel` is specified, `ciwidth` will compute results for only two combinations: (a_1, b_1) and (a_2, b_2).

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 the `graph()` option is specified. `table()` is used to produce custom tables. See [PSS-3] **ciwidth, 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-3] **ciwidth, graph** for details.

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

Iteration

`init(#)` specifies an initial value for the sample size when iteration is used to compute the 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. 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 `ciwidth` but is not shown in the dialog box:

`notitle` prevents the command title from displaying.

Remarks and examples

stata.com

Remarks are presented under the following headings:

This section describes how to perform PrSS analysis for CIs using the `ciwidth` command. For a software-free introduction to PrSS analysis, see [PSS-3] **Intro (ciwidth)**.

Using the ciwidth command

The `ciwidth` command computes sample size, CI width, and probability of CI width for various CIs. You can also add your own methods to the `ciwidth` command as described in [PSS-3] ***ciwidth usermethod***.

By default, all computations are performed for a two-sided CI, and the confidence level is set to 95%. You may change the confidence level by specifying the `level()` option. Alternatively, you can specify the significance level in the `alpha()` option. You can specify the `upper` and `lower` options to request upper and lower one-sided CIs.

To compute sample size, you must specify the CI width in the `width()` option and the probability of CI width in the `probwidth()` option. To compute CI width, you must specify the sample size in the `n()` option and the probability of CI width in the `probwidth()` option. You can also compute the probability of CI width given the sample size in `n()` and CI width in `width()`. For some CIs, you must also specify target values for parameters of interest as command arguments, such as a target variance for `ciwidth onevariance`.

The `probwidth()` option is analogous to the `power()` option in power and sample-size analysis. It accounts for the sampling variability of the CI width. For example, for CIs for means, the CI width depends on the variance, which is commonly estimated from the sample and may vary from one sample to another. To limit the impact of this sample-to-sample variability on the CI precision, you can use the `probwidth()` option to specify the probability that the width of a future CI will not exceed a target value. Without this adjustment, the results are generally conditional on the future data having the same variance as the one used for PrSS analysis and may underestimate the estimated sample size and CI width.

For CIs 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-samples CIs, 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-4] **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 CI width and probability of CI width because of rounding. To obtain conservative results, the

ciwidth command rounds up the sample size to the nearest integer so that the corresponding CI width is no larger than the requested one and the probability of CI width is no smaller than the requested one. You can specify the nfractional option to obtain the corresponding fractional sample size.

Some of ciwidth'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 ciwidth 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 many 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, ciwidth 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 ciwidth 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, ciwidth displays results as text. For multiple results or if the table option is specified, ciwidth 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-3] **ciwidth, table** and [PSS-3] **ciwidth, 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 ciwidth 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 *numlist* are marked with a star in the syntax diagram.

For example, the n() option supports multiple values. You can specify multiple sample sizes as individual values, n(100 150 200), or as a range of values, n(100(50)200); 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 $\#_1$ be the command argument in

```
. ciwidth onevariance #1, ...
```

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

```
. ciwidth onevariance (1 1.5), ...
```

or, more generally,

```
. ciwidth onevariance (numlist), ...
```

When multiple values are specified in multiple options or for multiple command arguments, the ciwidth 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 `ciwidth`. The last value from *numlist* of smaller sizes will be used in the subsequent computations.

For example,

```
. ciwidth onevariance (1 1.5), n(100 200) ...
```

is equivalent to

```
. ciwidth onevariance 1, n(100) ...
. ciwidth onevariance 1.5, n(100) ...
. ciwidth onevariance 1, n(200) ...
. ciwidth onevariance 1.5, n(200) ...
```

When the `parallel` option is specified,

```
. ciwidth onevariance (1 1.5), n(100 200) parallel ...
```

is equivalent to

```
. ciwidth onevariance 1, n(100) ...
. ciwidth onevariance 1.5, n(200) ...
```

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,

```
. ciwidth onevariance (1 1.5 2), n(100 200) parallel ...
```

is equivalent to

```
. ciwidth onevariance 1, n(100) ...
. ciwidth onevariance 1.5, n(200) ...
. ciwidth onevariance 2, n(200) ...
```

PrSS analysis for CIs for one population parameter

The `ciwidth` command provides PrSS analysis for two types of one-sample CIs. `ciwidth onemean` performs PrSS analysis for a CI for one population mean, and `ciwidth onevariance` performs PrSS analysis for a CI for one population variance.

`ciwidth onemean` provides PrSS computations for a one-mean CI. You can perform computations assuming a known or unknown population standard deviation and adjust the results for a finite population. See [PSS-3] **ciwidth onemean**.

`ciwidth onevariance` provides PrSS computations for a one-variance CI. You can perform computations in the variance or standard-deviation metric. See [PSS-3] **ciwidth onevariance**.

Below we show a quick example of PrSS analysis for a one-mean CI. See the individual entries for more examples.

Example 1: PrSS analysis for a one-mean CI

A group of pediatricians would like to study the exposure of infants to television. They are interested in estimating a CI for the average number of hours of television watched by infants per day. Before conducting a study, the pediatricians would like to determine how many infants they need to enroll in the study so that the CI width is not too large. They hypothesize that the average number of hours that infants spend watching television has a standard deviation of 0.8 hours. The group of pediatricians would like to compute the sample size required to produce a two-sided 95% CI with a width of 0.5 hours given this study parameter.

We use `ciwidth onemean` to compute the required sample size. We specify the standard deviation of 0.8 in the `sd()` option and the CI width of 0.5 in the `width()` option. However, this CI width can vary from sample to sample because samples tend to have different variances. To adjust the results for the sampling variability, we can use the `probwidth()` option to specify the probability that the width of the CI does not exceed our target value.

To be 96% certain that the CI width in a future study will be no larger than 0.5 hours, we specify `probwidth(0.96)`. By default, the confidence level is 95%, so we do not need to specify the `level(95)` option.

```
. ciwidth onemean, sd(0.8) probwidth(0.96) width(0.5)

Performing iteration ...

Estimated sample size for a one-mean CI
Student's t two-sided CI

Study parameters:

        level =     95.00
     Pr_width =    0.9600
        width =    0.5000
           sd =    0.8000

Estimated sample size:

            N =        56
```

The pediatricians need to enroll 56 infants in the study to be 96% certain that the 95% CI will be no wider than 0.5 for the average number of hours of television that infants watch per day.

All `ciwidth` commands have a similar output format. Information about the type of CI is displayed first. The input and implied values of the study parameters are displayed next under `Study parameters`. The estimated parameters, in this case the sample size, are displayed last.

Now suppose that we come across a pilot study reporting that infants watch an average of 2.5 hours of television per day, with a standard deviation of 0.8 hours. Now that the value of our standard deviation is known, rather than hypothesized, we can specify the `knownsd` option. With this option specified, `ciwidth onemean` will perform computations for a normal two-sided CI.

```
. ciwidth onemean, sd(0.8) width(0.5) knownsd

Estimated sample size for a one-mean CI
Normal two-sided CI

Study parameters:

        level =     95.00
        width =    0.5000
           sd =    0.8000

Estimated sample size:

            N =        40
```

Now that our standard deviation is known, the required sample size is smaller.

Let's suppose that we anticipate on having 50 infants in the study, and we want to see how the probability of obtaining the target CI width of 0.5 changes with this sample size. We now specify the `n(50)` option in addition to the CI width and standard deviation.

```
. ciwidth onemean, sd(0.8) n(50) width(0.5)
Estimated probability of width for a one-mean CI
Student's t two-sided CI
Study parameters:
        level =     95.00
            N =        50
        width =    0.5000
           sd =    0.8000
Estimated probability of width:
     Pr_width =    0.8501
```

With a sample of 50 infants, we are 85% certain that the CI width in a future study will be no larger than 0.5 hours, given a standard deviation of 0.8 for the average number of hours of television that infants watch per day. As expected, a smaller sample size leads to less certainty about the CI width. ◁

PrSS analysis for CIs comparing two independent samples

The `ciwidth twomeans` command provides PrSS analysis for CIs comparing the means from two independent samples. You can perform computations assuming known or unknown and equal or unequal population standard deviations. See [PSS-3] **ciwidth twomeans**.

Below we show a quick example of PrSS analysis for the CI comparing two means. See the individual entry for more examples.

▷ Example 2: PrSS analysis for a two-means difference CI

A pharmaceutical company would like to conduct a study to compare a new weight-loss drug with an older drug. Investigators are planning to compare the average weight loss in the two drugs by constructing a CI for the difference between the means of the two groups. The average weight loss for people taking the old drug for 3 months has a standard deviation of 5.5 pounds. The new drug is expected to produce greater weight loss, with a smaller standard deviation of 5 pounds for the same period of time. Investigators want to find out how many subjects they need to recruit to obtain a two-sided 95% CI for a difference between the two means with a target width of 6 pounds.

We use `ciwidth twomeans` to perform the PrSS analysis. We specify the control- and experimental-group standard deviations in the respective `sd1()` and `sd2()` options, along with the `knownsds` option. We also specify the target CI width of 6 pounds in the `width()` option.

```
. ciwidth twomeans, sd1(5.5) sd2(5) width(6) knownsds
Estimated sample sizes for a two-means-difference CI
Normal two-sided CI
Study parameters:
        level =     95.00
        width =    6.0000
          sd1 =    5.5000
          sd2 =    5.0000
Estimated sample sizes:
            N =        48
  N per group =        24
```

We need a sample of 48 subjects, 24 per group. Notice that our results assume that a future study will have the same standard deviations, 5.5 and 5, in the two groups.

◁

PrSS analysis for CIs comparing paired samples

`ciwidth pairedmeans` provides PrSS analysis for a CI for a difference between the means from two paired samples. You can perform computations assuming a known or unknown population standard deviation of the paired differences and adjust for a finite population. See [PSS-3] **ciwidth pairedmeans**. Below we show a quick example of using `ciwidth pairedmeans`.

▷ Example 3: PrSS analysis for a CI comparing paired means

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 to produce a two-sided 95%-level CI for the difference between the two means with a target width of 2 feet. The forester also wants to be at least 90% certain that the produced CI will have the width no larger than 2 feet.

We use `ciwidth pairedmeans` for our PrSS analysis. We supply the common value of standard deviations in the `sd()` option and the correlation of 0.4 in the `corr()` option. We also specify the CI width of 2 in the `width()` option and the probability of CI width of 0.9 in the `probwidth()` option.

```
. ciwidth pairedmeans, sd(4.5) corr(0.4) probwidth(0.9) width(2)

Performing iteration ...

Estimated sample size for a paired-means-difference CI
Student's t two-sided CI assuming sd1 = sd2 = sd

Study parameters:

        level =   95.0000           sd =    4.5000
     Pr_width =    0.9000         corr =    0.4000
        width =    2.0000
         sd_d =    4.9295

Estimated sample size:

            N =       113
```

The forester needs 113 pairs of pine trees to be 90% certain that the obtained two-sided 95% CI will have the width as narrow as 2 feet.

Note that the estimates of the means, 27.5 and 30 feet, do not affect the computation of the required sample size. We mention the average height of the trees for context of their standard deviations, which are used for the computation.

◁

Tables of results

When ciwidth 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 the results as text:

```
. ciwidth onemean, width(0.5) probwidth(0.9)
Performing iteration ...
Estimated sample size for a one-mean CI
Student's t two-sided CI
Study parameters:
        level =     95.00
     Pr_width =    0.9000
        width =    0.5000
           sd =    1.0000
Estimated sample size:
            N =        77
```

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

```
. ciwidth onemean, width(0.5) probwidth(0.9) table
Performing iteration ...
Estimated sample size for a one-mean CI
Student's t two-sided CI
```

level	N	Pr_width	width	sd
95	77	.9	.5	1

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

```
. ciwidth onemean, width(0.4 0.45 0.5) probwidth(0.9)
Performing iteration ...
Estimated sample size for a one-mean CI
Student's t two-sided CI
```

level	N	Pr_width	width	sd
95	116	.9	.4	1
95	93	.9	.45	1
95	77	.9	.5	1

In this example, we specified multiple sample sizes.

Default tables can be modified by specifying the table() option. For example, we can change the column labels:

```
. ciwidth onemean, width(0.4 0.45 0.5) probwidth(0.9) table(, labels(N "Sample
> size"))
Performing iteration ...
Estimated sample size for a one-mean CI
Student's t two-sided CI
```

level	Sample size	Pr_width	width	sd
95	116	.9	.4	1
95	93	.9	.45	1
95	77	.9	.5	1

Or we can select which columns we want to display:

```
. ciwidth onemean, width(0.4 0.45 0.5) probwidth(0.9) table(N width)
Performing iteration ...
Estimated sample size for a one-mean CI
Student's t two-sided CI
```

N	width
116	.4
93	.45
77	.5

The order of displayed columns follows the specified order. For more examples, see [PSS-3] **ciwidth, table**.

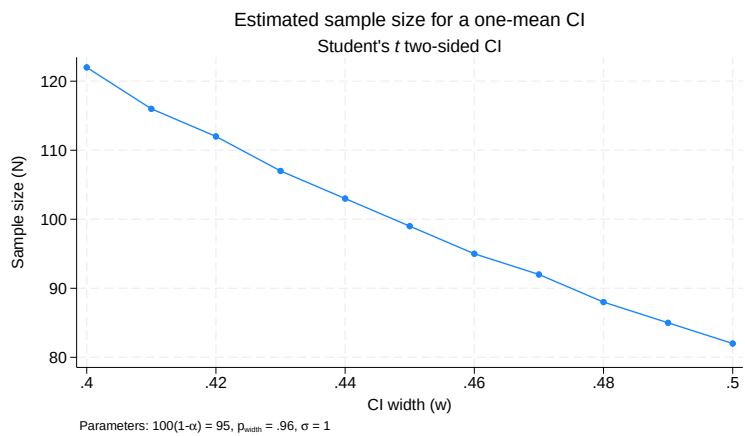
Sample-size and other curves

During the planning stage of a study, it is difficult to decide on the number of subjects to enroll on the basis of only one set of study parameters. It is common to investigate the effect of various study scenarios on CI precision or sample size. We can plot the estimated CI width, probability of CI width, or sample size versus one of the study parameters. The `ciwidth` command provides the `graph` and `graph()` options to produce such curves.

More precisely, when `graph` is specified, the estimated parameter such as CI width or sample size is plotted on the y axis, and the varying parameter is plotted on the x axis.

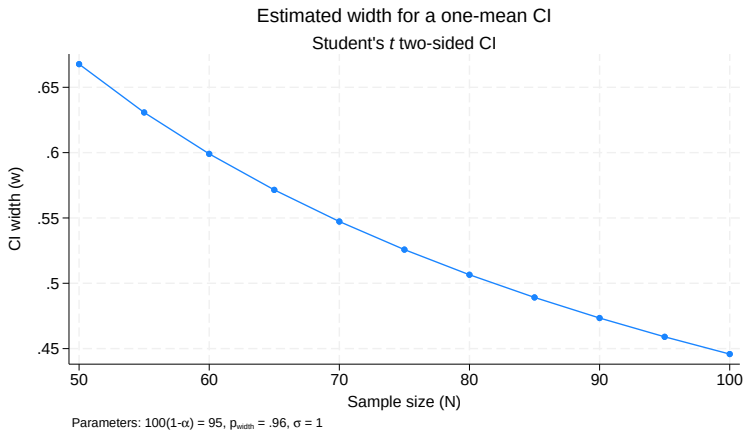
For example, we compute the sample size and plot it as a function of CI width for a range of CI width values between 0.4 and 0.5 with a step size of 0.01:

```
. ciwidth onemean, width(0.4(0.01)0.5) probwidth(0.96) graph
```



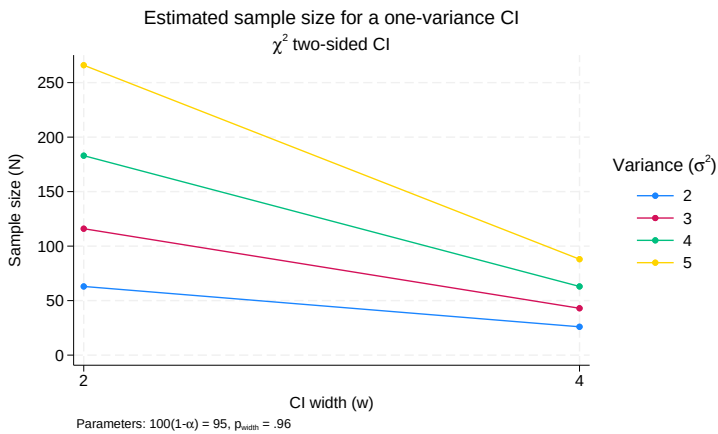
Or we can compute CI width and plot it as a function of sample size:

```
. ciwidth onemean, n(50(5)100) probwidth(0.96) graph
```



We can produce curves for multiple values of multiple parameters, such as the sample variances and the CI widths in this example:

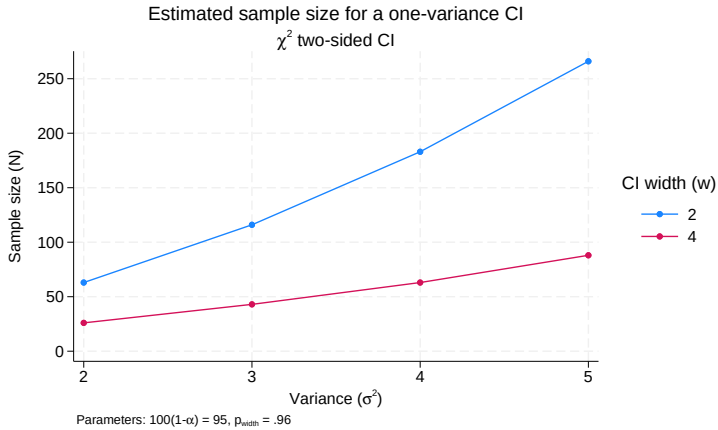
```
. ciwidth onevariance (2 3 4 5), width(2 4) probwidth(0.96) graph
```



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

For example, in the above, we could plot the sample size versus sample variances by using the `graph(xdimension(v))` option.

```
. ciwidth onevariance (2 3 4 5), width(2 4) probwidth(0.96)
> graph(xdimension(v))
```



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-3] **ciwidth, graph**.

Add your own methods to ciwidth

The `ciwidth` command provides several built-in methods, but sometimes, you may want to compute sample size or CI width yourself. For example, you may need to do this by simulation, or you may want to use a method that is not available in any software package. `ciwidth` makes it easy for you to add your own method. All you need to do is to write a program that computes sample size, probability of CI width, or CI width, and the `ciwidth` command will do the rest for you. It will deal with the support of multiple values in options and with automatic generation of graphs and tables of results.

See *A quick example* and *More examples: Compute probability of CI width for a one-proportion CI* in [PSS-3] ***ciwidth usermethod*** for examples.

Stored results

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

Scalars

`r(level)`	confidence level
`r(alpha)`	significance level
`r(N)`	total sample size
`r(N_a)`	actual sample size
`r(N1)`	sample size of the control group
`r(N2)`	sample size of the experimental group
`r(nratio)`	ratio of sample sizes, `N2/N1`
`r(nratio_a)`	actual ratio of sample sizes
`r(nfractional)`	`1` if `nfractional` is specified, `0` otherwise
`r(onesided)`	`1` for a one-sided CI, `0` otherwise

r(Pr_width)	probability of CI width
r(Pr_width_a)	actual probability of CI width (for sample-size determination when probwidth() specified)
r(width)	CI width
r(width_a)	actual CI width (for sample-size determination of some methods)
r(separator)	number of lines between separator lines in the table
r(divider)	1 if divider is requested in the table, 0 otherwise
r(init)	initial value for estimated parameter
r(maxiter)	maximum number of iterations
r(iter)	number of iterations performed
r(tolerance)	requested parameter tolerance
r(deltax)	final parameter tolerance achieved
r(ftolerance)	requested distance of the objective function from zero
r(function)	final distance of the objective function from zero
r(converged)	1 if iteration algorithm converged, 0 otherwise
Macros	
r(type)	ci
r(method)	the name of the specified ciwidth method
r(onesidedci)	upper or lower (for a one-sided CI)
r(columns)	displayed table columns
r(labels)	table column labels
r(widths)	table column widths
r(formats)	table column formats
Matrices	
r(pss_table)	table of results

Also see *Stored results* in the method-specific manual entries for the full list of stored results.

Methods and formulas

By default, the ciwidth command rounds sample sizes to integers and uses integer values in the computations. To ensure conservative results, the command rounds down the input sample sizes and rounds up the output sample sizes. See *Fractional sample sizes* in [PSS-4] **Unbalanced designs** for details.

Some of ciwidth's methods require iteration, such as a sample-size determination for a Student's t CI in ciwidth onemean. The methods use the Mata function solvenl(), and its Newton's method described in *Newton-type methods* in [M-5] **solvenl( )**, to solve nonlinear equations.

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

Also see

[PSS-3] **Intro (ciwidth)** — Introduction to precision and sample-size analysis for confidence intervals

[PSS-5] **Glossary**

[R] **ci** — Confidence intervals for means, proportions, and variances