

# Package ‘statpsych’

September 9, 2021

**Type** Package

**Title** Statistical Methods for Psychologists

**Version** 1.0.0

**Description** Implements confidence interval and sample size methods that are especially useful in psychological research. The methods can be applied in 1-group, 2-group, paired-samples, and multiple-group designs and to a variety of parameters including means, medians, proportions, slopes, standardized mean differences, standardized linear contrasts of means, plus several measures of correlation and association. The confidence intervals and sample size functions are applicable to single parameters as well as differences, ratios, and linear contrasts of parameters. The sample size functions can be used to approximate the sample size needed to estimate a parameter or function of parameters with desired confidence interval precision or to perform a variety of hypothesis tests (directional two-sided, equivalence, superiority, noninferiority) with desired power. For details, see: [https://csass.ucsc.edu/self-study%20courses/statistical\\_methods.html](https://csass.ucsc.edu/self-study%20courses/statistical_methods.html).

**BugReports** <https://github.com/dgbonett/statpsych/issues>

**License** GPL-3

**Encoding** UTF-8

**Imports** utils, stats, mathjaxr, Rdpack

**RoxygenNote** 7.1.1

**RdMacros** Rdpack, mathjaxr

**Suggests** testthat (>= 3.0.0)

**Config/testthat/edition** 3

**NeedsCompilation** no

**Author** Douglas G. Bonett [aut, cre],  
Robert J. Calin-Jageman [ctb]

**Maintainer** Douglas G. Bonett <dgbonett@ucsc.edu>

**Repository** CRAN

**Date/Publication** 2021-09-09 09:20:02 UTC

**R topics documented:**

ci.agree	4
ci.biphi	5
ci.cod1	6
ci.condslope	7
ci.condslope.log	8
ci.cor	9
ci.cor.dep	10
ci.cor2	11
ci.cor2.gen	12
ci.cronbach	13
ci.fisher	14
ci.indirect	15
ci.kappa	16
ci.lc.mean.bs	17
ci.lc.median.bs	18
ci.lc.prop.bs	19
ci.lc.reg	20
ci.lc.stdmean.bs	21
ci.lc.stdmean.ws	22
ci.mad1	23
ci.mann	24
ci.mape	25
ci.mean.ps	26
ci.mean1	27
ci.mean2	28
ci.median.ps	29
ci.median1	30
ci.median2	31
ci.oddsratio	32
ci.pairs.prop.bs	33
ci.pairs.prop1	34
ci.pbcor	35
ci.phi	36
ci.popsiz	37
ci.prop.ps	38
ci.prop1	39
ci.prop2	40
ci.random.anova1	41
ci.ratio.mad.ps	42
ci.ratio.mad2	43
ci.ratio.mean.ps	44
ci.ratio.mean2	45
ci.ratio.median.ps	46
ci.ratio.median2	47
ci.ratio.prop.ps	48
ci.ratio.prop2	49

ci.slope.prop.bs . . . . .	50
ci.spcor . . . . .	51
ci.spear . . . . .	52
ci.spear2 . . . . .	53
ci.stdmean.ps . . . . .	54
ci.stdmean.strat . . . . .	55
ci.stdmean1 . . . . .	56
ci.stdmean2 . . . . .	57
ci.tetra . . . . .	58
ci.tukey . . . . .	59
ci.var.upper . . . . .	60
ci.yule . . . . .	61
etasqr.adj . . . . .	62
iqv . . . . .	63
pi.score1 . . . . .	63
pi.score2 . . . . .	64
random.sample . . . . .	65
random.y . . . . .	66
random.yx . . . . .	67
randomize . . . . .	68
size.ci.agree . . . . .	68
size.ci.condmean . . . . .	69
size.ci.cor . . . . .	70
size.ci.cronbach . . . . .	71
size.ci.lc.ancova . . . . .	72
size.ci.lc.mean.bs . . . . .	73
size.ci.lc.mean.ws . . . . .	73
size.ci.lc.prop.bs . . . . .	74
size.ci.lc.stdmean.bs . . . . .	75
size.ci.lc.stdmean.ws . . . . .	76
size.ci.mean.ps . . . . .	77
size.ci.mean1 . . . . .	78
size.ci.mean2 . . . . .	78
size.ci.prop.ps . . . . .	79
size.ci.prop1 . . . . .	80
size.ci.prop2 . . . . .	81
size.ci.ratio.mean.ps . . . . .	81
size.ci.ratio.mean2 . . . . .	82
size.ci.ratio.prop.ps . . . . .	83
size.ci.ratio.prop2 . . . . .	84
size.ci.rsqr . . . . .	85
size.ci.second . . . . .	85
size.ci.slope . . . . .	86
size.ci.stdmean.ps . . . . .	87
size.ci.stdmean2 . . . . .	88
size.equiv.mean.ps . . . . .	89
size.equiv.mean2 . . . . .	90
size.equiv.prop.ps . . . . .	91

size.equiv.prop2 . . . . .	92
size.interval.cor . . . . .	93
size.supinf.mean.ps . . . . .	93
size.supinf.mean2 . . . . .	94
size.supinf.prop.ps . . . . .	95
size.supinf.prop2 . . . . .	96
size.test.cor . . . . .	97
size.test.cronbach . . . . .	98
size.test.lc.ancova . . . . .	99
size.test.lc.mean.bs . . . . .	100
size.test.lc.mean.ws . . . . .	100
size.test.lc.prop.bs . . . . .	101
size.test.mann . . . . .	102
size.test.mean.ps . . . . .	103
size.test.mean1 . . . . .	104
size.test.mean2 . . . . .	105
size.test.prop.ps . . . . .	105
size.test.prop1 . . . . .	106
size.test.prop2 . . . . .	107
size.test.sign.ps . . . . .	108
size.test.sign1 . . . . .	108
size.test.slope . . . . .	109
slope.contrast . . . . .	110
test.anova1.bs . . . . .	111
test.prop.bs . . . . .	112
test.prop.ps . . . . .	113
test.prop1 . . . . .	114
test.prop2 . . . . .	115

**Index** **116**

---

ci.agree	<i>Confidence interval for a G-index of agreement</i>
----------	---

---

**Description**

Computes a confidence interval for a G-index of agreement between two polychotomous ratings. This function requires the number of objects that were given the same rating by both raters.

**Usage**

```
ci.agree(alpha, n, f, k)
```

**Arguments**

alpha	alpha level for 1-alpha confidence
n	sample size
f	number of objects rated in agreement
k	number of rating categories

**Value**

Returns a 1-row matrix. The columns are:

- Estimate - estimate of G-index of agreement
- SE - standard error
- LL - lower limit of the confidence interval
- UL - upper limit of the confidence interval

**Examples**

```
ci.agree(.05, 100, 80, 4)

# Should return:
#      Estimate      SE      LL      UL
# [1,] 0.7333333 0.05333333 0.6132949 0.8226025
```

---

ci.biphi	<i>Confidence interval for a biserial-phi coefficient</i>
----------	---

---

**Description**

Computes a confidence interval for a biserial-phi coefficient using a transformation of a confidence interval for an odds ratio with .5 added to each cell frequency. This measure of association assumes the group variable is naturally dichotomous and the response variable is artificially dichotomous.

**Usage**

```
ci.biphi(alpha, f1, f2, n1, n2)
```

**Arguments**

alpha	alpha level for 1-alpha confidence
f1	number of participants in group 1 who have the attribute
f2	number of participants in group 2 who have the attribute
n1	sample size of group 1
n2	sample size of group 2

**Value**

Returns a 1-row matrix. The columns are:

- Estimate - estimate of phi coefficient
- SE - standard error
- LL - lower limit of the confidence interval
- UL - upper limit of the confidence interval

## References

Ulrich R, Wirtz M (2004). "On the correlation of a naturally and an artificially dichotomized variable." *British Journal of Mathematical and Statistical Psychology*, **57**(2), 235–251. ISSN 00071102, doi: [10.1348/0007110042307203](https://doi.org/10.1348/0007110042307203), <https://doi.wiley.com/10.1348/0007110042307203>.

## Examples

```
ci.biphi(.05, 46, 15, 100, 100)

# Should return:
#   Estimate      SE      LL      UL
# [1,] 0.4145733 0.07551281 0.2508866 0.546141
```

---

ci.cod1

*Confidence interval for a single coefficient of dispersion*

---

## Description

Computes a confidence interval for a population coefficient of dispersion (COD) which is defined as MAD/median. The COD is a robust alternative to the coefficient of variation and assumes ratio-scale scores.

## Usage

```
ci.cod1(alpha, y)
```

## Arguments

alpha	alpha level for 1-alpha confidence
y	vector of scores

## Value

Returns a 1-row matrix. The columns are:

- COD - estimated coefficient of dispersion
- LL - lower limit of the confidence interval
- UL - upper limit of the confidence interval

## References

Bonett DG, Seier E (2006). "Confidence interval for a coefficient of dispersion in nonnormal distributions." *Biometrical Journal*, **48**(1), 144–148. ISSN 0323-3847, doi: [10.1002/bimj.200410148](https://doi.org/10.1002/bimj.200410148), <https://onlinelibrary.wiley.com/doi/10.1002/bimj.200410148>.

**Examples**

```

y <- c(30, 20, 15, 10, 10, 60, 20, 25, 20, 30, 10, 5, 50, 40,
      20, 10, 0, 20, 50)
ci.cod1(.05, y)

# Should return:
#           COD          LL          UL
# [1,] 0.5921053 0.3813259 1.092679

```

---

ci.condslope

*Confidence interval for conditional (simple) slopes in a linear model*


---

**Description**

Computes confidence intervals and test statistics for population conditional slopes (simple slopes) in a general linear model that includes a predictor variable that is the product of a moderator variable and a predictor variable. Conditional slopes are computed at low and high values of the moderator variable.

**Usage**

```
ci.condslope(alpha, b1, b2, se1, se2, cov, lo, hi, dfe)
```

**Arguments**

alpha	alpha level for 1-alpha confidence
b1	sample coefficient for predictor variable
b2	sample coefficient for product variable
se1	standard error for predictor coefficient
se2	standard error for product coefficient
cov	sample covariance between predictor and product coefficients
lo	low value of moderator variable
hi	high value of moderator variable
dfe	error degrees of freedom

**Value**

Returns a 2-row matrix. The columns are:

- Estimate - estimated condition slope
- t - t test statistic
- p - p-value
- LL - lower limit of the confidence interval
- UL - upper limit of the confidence interval

**Examples**

```
ci.condslope(.05, .132, .154, .031, .021, .015, 5.2, 10.6, 122)

# Should return:
#           Estimate      SE      t  df      p
# At low moderator    0.9328 0.4109570 2.269824 122 0.024973618
# At high moderator   1.7644 0.6070517 2.906507 122 0.004342076
#           LL      UL
# At low moderator    0.1192696 1.746330
# At high moderator   0.5626805 2.966119
```

---

```
ci.condslope.log      Confidence interval for conditional (simple) slopes in a logistic model
```

---

**Description**

Computes confidence intervals and test statistics for population conditional slopes (simple slopes) in a general linear model that includes a predictor variable that is the product of a moderator variable and a predictor variable. Conditional slopes are computed at low and high values of the moderator variable.

**Usage**

```
ci.condslope.log(alpha, b1, b2, se1, se2, cov, lo, hi)
```

**Arguments**

alpha	alpha level for 1-alpha confidence
b1	sample coefficient for predictor variable
b2	sample coefficient for product variable
se1	standard error for predictor coefficient
se2	standard error for product coefficient
cov	sample covariance between predictor and product coefficients
lo	low value of moderator variable
hi	high value of moderator variable

**Value**

Returns a 2-row matrix. The columns are:

- Estimate - estimated condition slope
- exp(Estimate) - estimated exponentiated condition slope
- z - z test statistic
- p - p-value
- LL - lower limit of the confidence interval
- UL - upper limit of the confidence interval



**Examples**

```

ci.condslope.log(.05, .132, .154, .031, .021, .015, 5.2, 10.6)

# Should return:
#           Estimate exp(Estimate)      z      p
# At low moderator  0.9328      2.541616 2.269824 0.023218266
# At high moderator 1.7644      5.838068 2.906507 0.003654887
#           LL      UL
# At low moderator  1.135802  5.687444
# At high moderator 1.776421 19.186357

```

---

ci.cor

*Confidence interval for a Pearson or partial correlation*


---

**Description**

Computes a Fisher confidence interval for a population Pearson correlation or partial correlation with  $s$  control variables. Set  $s = 0$  for a Pearson correlation. This function uses a sample correlation as input. Use the `cor.test` function for a Pearson correlation with raw data input.

**Usage**

```
ci.cor(alpha, cor, s, n)
```

**Arguments**

alpha	alpha level for 1-alpha confidence
cor	sample Pearson or partial correlation
s	number of control variables
n	sample size

**Value**

Returns a 1-row matrix. The columns are:

- Estimate - estimated correlation
- SE - standard error
- LL - lower limit of the confidence interval
- UL - upper limit of the confidence interval

**References**

Snedecor GW, Cochran WG (1989). *Statistical Methods*, 8th edition. ISU University Pres, Ames, Iowa.

**Examples**

```
ci.cor(.05, .536, 0, 50)

# Should return:
#   Estimate      SE      LL      UL
# [1,]  0.536 0.1018149 0.3028333 0.7086249
```

---

<code>ci.cor.dep</code>	<i>Confidence interval for a difference in dependent Pearson correlations</i>
-------------------------	---

---

**Description**

Computes a confidence interval for a difference in population Pearson correlations that are estimated from the same sample and have one variable in common.

**Usage**

```
ci.cor.dep(alpha, cor1, cor2, cor12, n)
```

**Arguments**

alpha	alpha level for 1-alpha confidence
cor1	sample Pearson correlation between y and x1
cor2	sample Pearson correlation between y and x2
cor12	sample Pearson correlation between x1 and x2
n	sample size

**Value**

Returns a 1-row matrix. The columns are:

- Estimate - estimated correlation difference
- LL - lower limit of the confidence interval
- UL - upper limit of the confidence interval

**References**

Zou GY (2007). "Toward using confidence intervals to compare correlations." *Psychological Methods*, **12**(4), 399–413. ISSN 1939-1463, doi: [10.1037/1082989X.12.4.399](https://doi.org/10.1037/1082989X.12.4.399), <https://doi.org/getdoi.cfm?doi=10.1037/1082-989X.12.4.399>.

**Examples**

```
ci.cor.dep(.05, .396, .179, .088, 166)

# Should return:
#   Estimate      LL      UL
# [1,]  0.217 0.01366766 0.41594
```

---

ci.cor2

*Confidence interval for a 2-group Pearson correlation difference*


---

**Description**

Computes a confidence interval for a difference in population Pearson correlations in a 2-group design.

**Usage**

```
ci.cor2(alpha, cor1, cor2, n1, n2)
```

**Arguments**

alpha	alpha level for 1-alpha confidence
cor1	sample Pearson correlation in group 1
cor2	sample Pearson correlation in group 2
n1	sample size in group 1
n2	sample size in group 2

**Value**

Returns a 1-row matrix. The columns are:

- Estimate - estimated correlation difference
- LL - lower limit of the confidence interval
- UL - upper limit of the confidence interval

**References**

Zou GY (2007). "Toward using confidence intervals to compare correlations." *Psychological Methods*, **12**(4), 399–413. ISSN 1939-1463, doi: [10.1037/1082989X.12.4.399](https://doi.org/10.1037/1082989X.12.4.399), <https://doi.apa.org/getdoi.cfm?doi=10.1037/1082-989X.12.4.399>.

**Examples**

```
ci.cor2(.05, .886, .802, 200, 200)

# Should return:
#   Estimate      LL      UL
# [1,]  0.084 0.02795506 0.1457103
```

---

ci.cor2.gen

*Confidence interval for a 2-group correlation difference*


---

**Description**

Computes a 100(1 - alpha)% confidence interval for a difference in population correlations in a 2-group design. The correlations can be Pearson, Spearman, partial, semipartial, or point-biserial correlations. The function requires 100(1 - alpha)% confidence intervals for each correlation as input. This function also can be used to compute a confidence interval for the difference of two Cronbach reliability coefficients.

**Usage**

```
ci.cor2.gen(cor1, ll1, ul1, cor2, ll2, ul2)
```

**Arguments**

cor1	sample correlation in group 1
ll1	lower limit for group 1 correlation
ul1	upper limit for group 1 correlation
cor2	sample correlation in group 2
ll2	lower limit for group 2 correlation
ul2	upper limit for group 2 correlation

**Value**

Returns a 1-row matrix. The columns are:

- Estimate - estimated correlation difference
- LL - lower limit of the confidence interval
- UL - upper limit of the confidence interval

**References**

Zou GY (2007). "Toward using confidence intervals to compare correlations." *Psychological Methods*, **12**(4), 399–413. ISSN 1939-1463, doi: [10.1037/1082989X.12.4.399](https://doi.org/10.1037/1082989X.12.4.399), <https://doi.org/getdoi.cfm?doi=10.1037/1082-989X.12.4.399>.

**Examples**

```
ci.cor2.gen(.4, .35, .47, .2, .1, .32)

# Should return:
#   Estimate  LL      UL
# [1,]      0.2 0.07 0.3220656
```

---

ci.cronbach	<i>Confidence interval for a Cronbach reliability</i>
-------------	---

---

**Description**

Computes a confidence interval for a population Cronbach reliability. The point estimate of Cronbach's reliability assumes essentially tau-equivalent measurements and this confidence interval assumes parallel measurements.

**Usage**

```
ci.cronbach(alpha, rel, a, n)
```

**Arguments**

alpha	alpha level for 1-alpha confidence
rel	sample value of Cronbach's reliability
a	number of measurements (items, raters, etc.)
n	sample size

**Value**

Returns a 1-row matrix. The columns are:

- LL - lower limit of the confidence interval
- UL - upper limit of the confidence interval

**References**

Feldt LS (1965). "The approximate sampling distribution of Kuder-Richardson reliability coefficient twenty." *Psychometrika*, **30**(3), 357–370. ISSN 0033-3123, doi: [10.1007/BF02289499](https://doi.org/10.1007/BF02289499), <https://link.springer.com/10.1007/BF02289499>.

**Examples**

```
ci.cronbach(.05, .85, 7, 89)

# Should return:
#           LL           UL
# [1,] 0.7971254 0.8931436
```

---

ci.fisher	<i>Fisher confidence interval</i>
-----------	-----------------------------------

---

**Description**

Computes a Fisher confidence interval for any type of correlation or any measure of association that has a -1 to 1 range.

**Usage**

```
ci.fisher(alpha, cor, se)
```

**Arguments**

alpha	alpha value for 1-alpha confidence
cor	sample correlation or association coefficient
se	standard error of estimate

**Value**

Returns a 1-row matrix containing the lower and upper confidence limits.

**Examples**

```
ci.fisher(.05, .641, .052)

# Should return:
#           LL           UL
# [1,] 0.5276396 0.7319293
```

---

ci.indirect

*Confidence interval for an indirect effect*


---

### Description

Computes a Monte Carlo confidence interval (500,000 trials) for a population unstandardized indirect effect in a path model. This function is not recommended for a standardized indirect effect unless the standardized slope estimates for both paths are less than about .3 in absolute value. The Monte Carlo method is general in that the slope estimates and standard errors do not need to be OLS estimates with homoscedastic standard errors. For example, LAD slope estimates and their standard errors, OLS slope estimates and heteroscedastic standard errors, or distribution-free Theil-Sen slope estimates with McKean-Schrader standard errors also could be used.

### Usage

```
ci.indirect(alpha, b1, b2, se1, se2)
```

### Arguments

alpha	alpha level for 1-alpha confidence
b1	unstandardized slope estimate for first path
b2	unstandardized slope estimate for second path
se1	standard error for b1
se2	standard error for b2

### Value

Returns a 1-row matrix. The columns are:

- Estimate - estimated indirect effect
- SE - standard error
- LL - lower limit of the confidence interval
- UL - upper limit of the confidence interval

### Examples

```
ci.indirect (.05, 2.48, 1.92, .586, .379)

# Should return:
#   Estimate      SE      LL      UL
# [1,]  4.7616 1.625282 2.178812 7.972262
```

---

ci.kappa

*Confidence interval for a kappa reliability*


---

**Description**

Computes a confidence interval for the intraclass kappa coefficient and Cohen's kappa coefficient for two dichotomous ratings. Both measures are intraclass reliability coefficients.

**Usage**

```
ci.kappa(alpha, f00, f01, f10, f11)
```

**Arguments**

alpha	alpha level for 1-alpha confidence
f00	number of objects rated y = 0 and x = 0
f01	number of objects rated y = 0 and x = 1
f10	number of objects rated y = 1 and x = 0
f11	number of objects rated y = 1 and x = 1

**Value**

Returns a 2-row matrix. The results in row 1 are for the intraclass kappa. The results in row 2 are for Cohen's kappa. The columns are:

- Estimate - estimate of interrater reliability
- SE - standard error
- LL - lower limit of the confidence interval
- UL - upper limit of the confidence interval

**References**

Fleiss JL, Paik MC (2003). *Statistical Methods for Rates and Proportions*, 3rd edition. Wiley.

**Examples**

```
ci.kappa(.05, 31, 12, 4, 58)

# Should return:
#           Estimate      SE      LL      UL
# IC kappa:  0.6736597 0.07479965 0.5270551 0.8202643
# Cohen kappa: 0.6756757 0.07344761 0.5317210 0.8196303
```



---

ci.lc.mean.bs	<i>Confidence interval for a linear contrast of means in a between-subjects design</i>
---------------	--

---

## Description

Computes a test statistic and confidence interval for a linear contrast of means. This function computes both unequal variance (recommended) and equal variance confidence intervals and test statistics. A Satterthwaite adjustment to the degrees of freedom is used with the unequal variance method.

## Usage

```
ci.lc.mean.bs(alpha, m, sd, n, v)
```

## Arguments

alpha	alpha level for 1-alpha confidence
m	vector of group sample means
sd	vector of group sample standard deviations
n	vector of sample sizes
v	vector of between-subjects contrast coefficients

## Value

Returns a 2-row matrix. The columns are:

- Estimate - estimated linear contrast
- SE - standard error
- t - t test statistic
- df - degrees of freedom
- p - p-value
- LL - lower limit of the confidence interval
- UL - upper limit of the confidence interval

## References

Snedecor GW, Cochran WG (1989). *Statistical Methods*, 8th edition. ISU University Pres, Ames, Iowa.

**Examples**

```

m <- c(33.5, 37.9, 38.0, 44.1)
sd <- c(3.84, 3.84, 3.65, 4.98)
n <- c(10,10,10,10)
v <- c(.5, .5, -.5, -.5)
ci.lc.mean.bs(.05, m, sd, n, v)

# Should return:
#
#           Estimate      SE      t      df
# Equal Variances Assumed:    -5.35  1.300136  -4.114955  36.00000
# Equal Variances Not Assumed:  -5.35  1.300136  -4.114955  33.52169
#
#           p      LL      UL
# Equal Variances Assumed:    0.0002152581  -7.986797  -2.713203
# Equal Variances Not Assumed: 0.0002372436  -7.993583  -2.706417

```

---

ci.lc.median.bs	<i>Confidence interval for a linear contrast of medians in a between-subjects design</i>
-----------------	--

---

**Description**

Computes a confidence interval for a linear contrast of medians in a between-subjects design using sample medians and their standard errors. The sample median and standard error for each group can be computed using the ci.median1 function.

**Usage**

```
ci.lc.median.bs(alpha, m, se, v)
```

**Arguments**

alpha	alpha level for 1-alpha confidence
m	vector of group sample medians
se	vector of group standard errors
v	vector of between-subjects contrast coefficients

**Value**

Returns a 1-row matrix. The columns are:

- Estimate - estimated linear contrast of medians
- SE - standard error
- LL - lower limit of the confidence interval
- UL - upper limit of the confidence interval

## References

Bonett DG, Price RM (2002). "Statistical inference for a linear function of medians: Confidence intervals, hypothesis testing, and sample size requirements." *Psychological Methods*, 7(3), 370–383. ISSN 1939-1463, doi: [10.1037/1082989X.7.3.370](https://doi.org/10.1037/1082989X.7.3.370), <https://doi.org/10.1037/1082-989X.7.3.370>.

## Examples

```
m <- c(46.13, 29.19, 30.32, 49.15)
se <- c(6.361, 5.892, 4.887, 6.103)
v <- c(1, -1, -1, 1)
ci.lc.median.bs(.05, m, se, v)

# Should return:
#   Estimate      SE      LL      UL
# [1,]  35.77 11.67507 12.88727 58.65273
```

---

ci.lc.prop.bs	<i>Confidence interval for a linear contrast of proportions in a between-subjects design</i>
---------------	--

---

## Description

Computes an adjusted Wald confidence interval for a linear contrast of proportions in a between-subjects design.

## Usage

```
ci.lc.prop.bs(alpha, f, n, v)
```

## Arguments

alpha	alpha level for 1-alpha confidence
f	vector of frequency counts of participants with attribute
n	vector of sample sizes
v	vector of between-subjects contrast coefficients

## Value

Returns a 1-row matrix. The columns are:

- Estimate - adjusted estimate of proportion linear contrast
- SE - adjusted standard error
- z - z test statistic
- p - p-value
- LL - lower limit of the adjusted Wald confidence interval
- UL - upper limit of the adjusted Wald confidence interval

## References

Price RM, Bonett DG (2004). “An improved confidence interval for a linear function of binomial proportions.” *Computational Statistics & Data Analysis*, **45**(3), 449–456. ISSN 01679473, doi: [10.1016/S01679473\(03\)000070](https://doi.org/10.1016/S01679473(03)000070), <https://linkinghub.elsevier.com/retrieve/pii/S0167947303000070>.

## Examples

```
f <- c(26, 24, 38)
n <- c(60, 60, 60)
v <- c(-.5, -.5, 1)
ci.lc.prop.bs(.05, f, n, v)

# Should return:
#      Estimate      SE      z      p      LL      UL
# [1,] 0.2119565 0.07602892 2.787841 0.005306059 0.06294259 0.3609705
```

---

ci.lc.reg

*Confidence interval for a linear contrast of regression coefficients*

---

## Description

Compute a confidence interval and test statistic for a linear contrast of a population regression coefficient (y-intercept or slope) across groups in a multiple group regression model. Equality of error variances across groups is not assumed. A Satterthwaite adjustment to the degrees of freedom is used to improve the accuracy of the confidence interval.

## Usage

```
ci.lc.reg(alpha, est, se, n, s, v)
```

## Arguments

alpha	alpha level for 1-alpha confidence
est	vector of parameter estimates
se	vector of standard errors
n	vector of group sample sizes
s	number of predictor variables for within-group model
v	vector of between-subject contrast coefficients

**Value**

Returns a 1-row matrix. The columns are:

- Estimate - estimated linear contrast
- SE - standard error
- t - t test statistic
- df - degrees of freedom
- p - p-value
- LL - lower limit of the confidence interval
- UL - upper limit of the confidence interval

**Examples**

```
est <- c(1.74, 1.83, 0.482)
se <- c(.483, .421, .395)
n <- c(40, 40, 40)
v <- c(.5, .5, -1)
ci.lc.reg(.05, est, se, n, 4, v)

# Should return:
#      Estimate      SE      t      df      p      LL      UL
# [1,]  1.303 0.5085838 2.562016 78.8197 0.01231256 0.2906532 2.315347
```

---

ci.lc.stdmean.bs	<i>Confidence interval for a standardized linear contrast of means in a between-subjects design</i>
------------------	---

---

**Description**

Computes confidence intervals for a population standardized linear contrast of means in a between-subjects design. The square root weighted variance standardizer is recommended in 2-group nonexperimental designs with simple random sampling. The square root unweighted variance standardizer is recommended in 2-group experimental designs. Equality of variances is not assumed.

**Usage**

```
ci.lc.stdmean.bs(alpha, m, sd, n, v)
```

**Arguments**

alpha	alpha level for 1-alpha confidence
m	vector of group sample means
sd	vector of group sample standard deviation
n	vector of sample sizes
v	vector of between-subjects contrast coefficients

**Value**

Returns a 2-row matrix. The columns are:

- Estimate - bias adjusted standardized linear contrast
- SE - standard error
- LL - lower limit of the confidence interval
- UL - upper limit of the confidence interval

**References**

Bonett DG (2008). "Confidence intervals for standardized linear contrasts of means." *Psychological Methods*, **13**(2), 99–109. ISSN 1939-1463, doi: [10.1037/1082989X.13.2.99](https://doi.org/10.1037/1082989X.13.2.99), <https://doi.org/10.1037/1082-989X.13.2.99>.

**Examples**

```
m <- c(33.5, 37.9, 38.0, 44.1)
sd <- c(3.84, 3.84, 3.65, 4.98)
n <- c(10,10,10,10)
v <- c(.5, .5, -.5, -.5)
ci.lc.stdmean.bs(.05, m, sd, n, v)

# Should return:
#               Estimate      SE      LL      UL
# Unweighted standardizer: -1.273964 0.3692800 -2.025039 -0.5774878
# Weighted standardizer:   -1.273964 0.3514511 -1.990095 -0.6124317
```

---

ci.lc.stdmean.ws	<i>Confidence interval for a standardized linear contrast of means in a within-subjects design</i>
------------------	--

---

**Description**

Computes a confidence interval for a population standardized linear contrast of means in a within-subjects design. A square root unweighted variance standardizer is used. Equality of variances is not assumed but the correlations among the repeated measures are assumed to be approximately equal.

**Usage**

```
ci.lc.stdmean.ws(alpha, m, sd, cor, n, q)
```

**Arguments**

alpha	alpha level for 1-alpha confidence
m	vector of within-subjects sample means
sd	vector of within-subjects sample standard deviations
cor	average correlation of measurement pairs
n	sample size
q	vector of within-subjects contrast coefficients

**Value**

Returns a 1-row matrix. The columns are:

- Estimate - bias adjusted standardized linear contrast
- SE - standard error
- LL - lower limit of the confidence interval
- UL - upper limit of the confidence interval

**References**

Bonett DG (2008). "Confidence intervals for standardized linear contrasts of means." *Psychological Methods*, **13**(2), 99–109. ISSN 1939-1463, doi: [10.1037/1082989X.13.2.99](https://doi.org/10.1037/1082989X.13.2.99), <https://doi.org/10.1037/1082-989X.13.2.99>.

**Examples**

```
m <- c(33.5, 37.9, 38.0, 44.1)
sd <- c(3.84, 3.84, 3.65, 4.98)
q <- c(.5, .5, -.5, -.5)
ci.lc.stdmean.ws(.05, m, sd, .672, 20, q)

# Should return:
#      Estimate      SE      LL      UL
# [1,] -1.266557  0.1860897 -1.665992 -0.936534
```

---

ci.mad1

*Confidence interval for a single MAD*


---

**Description**

Computes a confidence interval for a population mean absolute deviation from the median (MAD).

**Usage**

```
ci.mad1(alpha, y)
```

**Arguments**

alpha            alpha level for 1-alpha confidence  
y                 vector of scores

**Value**

Returns a 1-row matrix. The columns are:

- MAD - estimated mean absolute deviation
- LL - lower limit of the confidence interval
- UL - upper limit of the confidence interval

**References**

Bonett DG, Seier E (2003). "Confidence intervals for mean absolute deviations." *The American Statistician*, **57**(4), 233–236. ISSN 0003-1305, doi: [10.1198/0003130032323](https://doi.org/10.1198/0003130032323), <https://www.tandfonline.com/doi/abs/10.1198/0003130032323>.

**Examples**

```
y <- c(30, 20, 15, 10, 10, 60, 20, 25, 20, 30, 10, 5, 50, 40,
      20, 10, 0, 20, 50)
ci.mad1(.05, y)

# Should return:
#      MAD      LL      UL
# [1,] 12.5  7.962667 19.62282
```

---

ci.mann

*Confidence interval for a Mann-Whitney probability*


---

**Description**

Computes a distribution-free confidence interval for the Mann-Whitney parameter. In a 2-group experiment, this parameter is the proportion of members in the population with scores that would be higher under treatment 1 than treatment 2. In a 2-group nonexperiment where participants are sampled from two subpopulations of sizes  $N_1$  and  $N_2$ , the parameter is the proportion of all  $N_1 \times N_2$  pairs in which a member from subpopulation 1 has a larger score than a member from subpopulation 2.

**Usage**

```
ci.mann(alpha, y1, y2)
```



**Arguments**

alpha	alpha level for 1-alpha confidence
y1	vector of scores for group 1
y2	vector of scores for group 2

**Value**

Returns a 1-row matrix. The columns are:

- Estimate - estimate of probability
- SE - standard error
- LL - lower limit of the confidence interval
- UL - upper limit of the confidence interval

**References**

Sen PK (1967). "A note on asymptotically distribution-free confidence bounds for  $PX < Y$ , based on two independent samples." *The Indian Journal of Statistics, Series A*, **29**(1), 95–102. <https://www.jstor.org/stable/25049448>.

**Examples**

```
y1 <- c(32, 39, 26, 35, 43, 27, 40, 37, 34, 29)
y2 <- c(36, 44, 47, 42, 49, 39, 46, 31, 33, 48)
ci.mann(.05, y1, y2)

# Should return:
#   Estimate      SE      LL UL
# [1,]  0.795 0.1401834 0.5202456 1
```

---

ci.mape

*Confidence interval for a mean absolute prediction error*


---

**Description**

Computes a confidence interval for a population mean absolute prediction error (MAPE) in a general linear model. The MAPE is a robust alternative to the residual standard deviation. This function requires a vector of estimated residuals from a general linear model.

**Usage**

```
ci.mape(alpha, r, s)
```

**Arguments**

alpha	alpha level for 1-alpha confidence
r	vector of residuals
s	number of predictor variables in model

**Value**

Returns a 1-row matrix. The columns are:

- MAPE - estimated mean absolute prediction error
- LL - lower limit of the confidence interval
- UL - upper limit of the confidence interval

**Examples**

```
r <- c(-2.70, -2.69, -1.32, 1.02, 1.23, -1.46, 2.21, -2.10, 2.56,
      -3.02, -1.55, 1.46, 4.02, 2.34)
ci.mape(.05, r, 1)

# Should return:
#      MAPE      LL      UL
# [1,] 2.3744 1.751678 3.218499
```

---

ci.mean.ps

*Confidence interval for a paired-samples mean difference*


---

**Description**

Computes a confidence interval for a population paired-samples mean difference using the sample means, sample standard deviations, sample correlation, and sample size. Use the t.test function for raw data input.

**Usage**

```
ci.mean.ps(alpha, m1, m2, sd1, sd2, cor, n)
```

**Arguments**

alpha	alpha level for 1-alpha confidence
m1	sample mean for measurement 1
m2	sample mean for measurement 2
sd1	sample standard deviation for measurement 1
sd2	sample standard deviation for measurement 2
cor	sample correlation between measurements
n	sample size

**Value**

Returns a 1-row matrix. The columns are:

- Estimate - estimated mean difference
- SE - standard error
- t - t test statistic
- df - degrees of freedom
- p - p-value
- LL - lower limit of the confidence interval
- UL - upper limit of the confidence interval

**Examples**

```
ci.mean.ps(.05, 58.2, 51.4, 7.43, 8.92, .537, 30)

# Should return:
#   Estimate      SE      t df      p      LL      UL
# [1,]      6.8 1.455922 4.670578 29 6.33208e-05 3.822304 9.777696
```

---

ci.mean1

*Confidence interval for a single mean*

---

**Description**

Computes a confidence interval for a population mean using the sample mean, sample standard deviation, and sample size. Use the t.test function for raw data input.

**Usage**

```
ci.mean1(alpha, m, sd, n)
```

**Arguments**

alpha	alpha level for 1-alpha confidence
m	sample mean
sd	sample standard deviation
n	sample size

**Value**

Returns a 1-row matrix. The columns are:

- Estimate - estimated mean
- SE - standard error
- LL - lower limit of the confidence interval
- UL - upper limit of the confidence interval

**References**

Snedecor GW, Cochran WG (1989). *Statistical Methods*, 8th edition. ISU University Pres, Ames, Iowa.

**Examples**

```
ci.mean1(.05, 24.5, 3.65, 40)

# Should return:
#      Estimate      SE      LL      UL
# [1,]      24.5 0.5771157 23.33267 25.66733
```

---

<code>ci.mean2</code>	<i>Confidence interval for a 2-group mean difference</i>
-----------------------	--

---

**Description**

Computes a confidence interval for a population 2-group mean difference using the sample means, sample standard deviations, and sample sizes. Use the `t.test` function for raw data input.

**Usage**

```
ci.mean2(alpha, m1, m2, sd1, sd2, n1, n2)
```

**Arguments**

<code>alpha</code>	alpha level for 1-alpha confidence
<code>m1</code>	sample mean for group 1
<code>m2</code>	sample mean for group 2
<code>sd1</code>	sample standard deviation for group 1
<code>sd2</code>	sample standard deviation for group 2
<code>n1</code>	sample size for group 1
<code>n2</code>	sample size for group 2

**Value**

Returns a 2-row matrix. The columns are:

- Estimate - estimated mean difference
- SE - standard error
- t - t test statistic
- df - degrees of freedom
- p - p-value
- LL - lower limit of the confidence interval
- UL - upper limit of the confidence interval

## References

Snedecor GW, Cochran WG (1989). *Statistical Methods*, 8th edition. ISU University Pres, Ames, Iowa.

## Examples

```
ci.mean2(.05, 15.4, 10.3, 2.67, 2.15, 30, 20)

# Should return:
#
#           Estimate      SE      t      df
# Equal Variances Assumed:      5.1 1.602248 3.183029 48.0000
# Equal Variances Not Assumed:  5.1 1.406801 3.625247 44.1137
#
#           p      LL      UL
# Equal Variances Assumed:  0.0025578586 1.878465 8.321535
# Equal Variances Not Assumed: 0.0007438065 2.264986 7.935014
```

---

ci.median.ps

*Confidence interval for a paired-samples median difference*

---

## Description

Computes a confidence interval for a difference of population medians in a paired-samples design. This function also computes the standard errors for each median and the covariance between the two medians.

## Usage

```
ci.median.ps(alpha, y1, y2)
```

## Arguments

alpha	alpha level for 1-alpha confidence
y1	vector of scores for measurement 1
y2	vector of scores for measurement 2

## Value

Returns a 1-row matrix. The columns are:

- Median1 - estimated median for measurement 1
- Median2 - estimated median for measurement 2
- Median1-Median2 - estimated difference of medians
- SE1 - standard error of median 1
- SE2 - standard error of median 2

- cov - covariance of medians
- LL - lower limit of the confidence interval
- UL - upper limit of the confidence interval

## References

Bonett DG, Price RM (2020). "Interval estimation for linear functions of medians in within-subjects and mixed designs." *British Journal of Mathematical and Statistical Psychology*, **73**(2), 333–346. ISSN 0007-1102, doi: [10.1111/bmsp.12171](https://doi.org/10.1111/bmsp.12171), <https://onlinelibrary.wiley.com/doi/10.1111/bmsp.12171>.

## Examples

```
y1 <- c(21, 4, 9, 12, 35, 18, 10, 22, 24, 1, 6, 8, 13, 16, 19)
y2 <- c(67, 28, 30, 28, 52, 40, 25, 37, 44, 10, 14, 20, 28, 40, 51)
ci.median.ps(.05, y1, y2)

# Should return:
#   Median1 Median2 Median1-Median2      SE      LL      UL
# [1,]    13     30          -17 3.362289 -23.58996 -10.41004
#           SE1     SE2      cov
#   3.085608 4.509735 9.276849
```

---

<code>ci.median1</code>	<i>Confidence interval for a single median</i>
-------------------------	--

---

## Description

Computes a confidence interval for a single population median.

## Usage

```
ci.median1(alpha, y)
```

## Arguments

<code>alpha</code>	alpha level for 1-alpha confidence
<code>y</code>	vector of scores

## Value

Returns a 1-row matrix. The columns are:

- Median - estimated median
- SE - standard error
- LL - lower limit of the confidence interval
- UL - upper limit of the confidence interval

**References**

Snedecor GW, Cochran WG (1989). *Statistical Methods*, 8th edition. ISU University Pres, Ames, Iowa.

**Examples**

```
y <- c(30, 20, 15, 10, 10, 60, 20, 25, 20, 30, 10, 5, 50, 40,
      20, 10, 0, 20, 50)
ci.median1(.05, y)

# Should return:
#   Median      SE LL UL
# [1,]    20 5.390263 10 30
```

---

ci.median2

*Confidence interval for a 2-group median difference*


---

**Description**

Computes a confidence interval for a 2-group difference of population medians.

**Usage**

```
ci.median2(alpha, y1, y2)
```

**Arguments**

alpha	alpha level for 1-alpha confidence
y1	vector of scores for group 1
y2	vector of scores for group 2

**Value**

Returns a 1-row matrix. The columns are:

- Median1 - estimated median from group 1
- Median2 - estimated median from group 2
- Median1-Median2 - estimated difference in medians
- SE - standard error of the difference
- LL - lower limit of the confidence interval
- UL - upper limit of the confidence interval

## References

Bonett DG, Price RM (2002). "Statistical inference for a linear function of medians: Confidence intervals, hypothesis testing, and sample size requirements." *Psychological Methods*, 7(3), 370–383. ISSN 1939-1463, doi: [10.1037/1082989X.7.3.370](https://doi.org/10.1037/1082989X.7.3.370), <https://doi.org/10.1037/1082-989X.7.3.370>.

## Examples

```
y1 <- c(32, 39, 26, 35, 43, 27, 40, 37, 34, 29)
y2 <- c(36, 44, 47, 42, 49, 39, 46, 31, 33, 48)
ci.median2(.05, y1, y2)

# Should return:
#      Median1 Median2 Median1-Median2      SE      LL      UL
# [1,]    34.5     43          -8.5 4.316291 -16.95977 -0.04022524
```

---

ci.oddsratio	<i>Confidence interval for an odds ratio</i>
--------------	--

---

## Description

Computes a confidence interval for an odds ratio with .5 added to each cell frequency. This function requires the frequency counts from a 2 x 2 contingency table for two dichotomous variables.

## Usage

```
ci.oddsratio(alpha, f00, f01, f10, f11)
```

## Arguments

alpha	alpha level for 1-alpha confidence
f00	number of participants with y = 0 and x = 0
f01	number of participants with y = 0 and x = 1
f10	number of participants with y = 1 and x = 0
f11	number of participants with y = 1 and x = 1

## Value

Returns a 1-row matrix. The columns are:

- Estimate - estimate of proportion ratio
- LL - lower limit of the confidence interval
- UL - upper limit of the confidence interval



**References**

Fleiss JL, Paik MC (2003). *Statistical Methods for Rates and Proportions*, 3rd edition. Wiley.

**Examples**

```
ci.oddsratio(.05, 229, 28, 96, 24)

# Should return:
#   Estimate      LL      UL
# [1,] 2.044451 1.133267 3.688254
```

---

ci.pairs.prop.bs	<i>Bonferroni confidence intervals for all pairwise proportion differences in a between-subjects design</i>
------------------	---

---

**Description**

Computes adjusted Wald confidence intervals for all pairwise differences of proportions in a between-subjects design with a Bonferroni adjusted alpha level.

**Usage**

```
ci.pairs.prop.bs(alpha, f, n)
```

**Arguments**

alpha	alpha level for simultaneous 1-alpha confidence
f	vector of frequency counts of participants with attribute
n	vector of sample sizes

**Value**

Returns a 1-row matrix. The columns are:

- Estimate - adjusted estimate of proportion difference
- SE - adjusted standard error
- z - z test statistic
- p - p-value
- LL - lower limit of the adjusted Wald confidence interval
- UL - upper limit of the adjusted Wald confidence interval

## References

Agresti A, Caffo B (2000). "Simple and effective confidence intervals for proportions and differences of proportions result from adding two successes and two failures." *The American Statistician*, 54(4), 280. ISSN 00031305, doi: [10.2307/2685779](https://doi.org/10.2307/2685779), <https://www.jstor.org/stable/2685779?origin=crossref>.

## Examples

```
f <- c(111, 161, 132)
n <- c(200, 200, 200)
ci.pairs.prop.bs(.05, f, n)

# Should return:
#      Estimate      SE      z      p      LL      UL
# 1 2 -0.2475248 0.04482323 -5.522243 3.346989e-08 -0.35483065 -0.14021885
# 1 3 -0.1039604 0.04833562 -2.150803 3.149174e-02 -0.21967489  0.011175409
# 2 3  0.1435644 0.04358401  3.293968 9.878366e-04  0.03922511  0.24790360
```

---

<code>ci.pairs.prop1</code>	<i>Confidence intervals for pairwise proportion differences of a polychotomous variable</i>
-----------------------------	---

---

## Description

Computes adjusted Wald confidence intervals for pairwise proportion differences of a polychotomous variable. These adjusted Wald confidence intervals use the same method that is used to compare proportions in a paired-samples design.

## Usage

```
ci.pairs.prop1(alpha, f)
```

## Arguments

<code>alpha</code>	alpha level for 1-alpha confidence
<code>f</code>	vector of multinomial frequency counts

## Value

Returns a 1-row matrix. The columns are:

- Estimate - adjusted difference of proportions
- SE - adjusted standard error
- LL - lower limit of the adjusted Wald confidence interval
- UL - upper limit of the adjusted Wald confidence interval

## References

Bonett DG, Price RM (2012). “Adjusted wald confidence interval for a difference of binomial proportions based on paired data.” *Journal of Educational and Behavioral Statistics*, **37**(4), 479–488. ISSN 1076-9986, doi: [10.3102/1076998611411915](https://doi.org/10.3102/1076998611411915), <https://journals.sagepub.com/doi/10.3102/1076998611411915>.

## Examples

```
f <- c(125, 82, 92)
ci.pairs.prop1(.05, f)

# Should return:
#      Estimate      SE      LL      UL
# 1 2  0.14285714 0.04731825 0.05011508 0.23559920
# 1 3  0.10963455 0.04875715 0.01407230 0.20519680
# 2 3 -0.03322259 0.04403313 -0.11952594 0.05308076
```

---

ci.pbcor

*Confidence interval for a point-biserial correlation*

---

## Description

Computes confidence intervals for two types of population point-biserial correlations. One type uses a weighted average of the group variances and is appropriate for nonexperimental designs with simple random sampling (rather than stratified random sampling). The other type uses an unweighted average of the group variances and is appropriate for experimental designs. Equality of variances is not assumed for either type.

## Usage

```
ci.pbcor(alpha, m1, m2, sd1, sd2, n1, n2)
```

## Arguments

alpha	alpha level for 1-alpha confidence
m1	sample mean for group 1
m2	sample mean for group 2
sd1	sample standard deviation for group 1
sd2	sample standard deviation for group 2
n1	sample size for group 1
n2	sample size for group 2

**Value**

Returns a 2-row matrix. The columns are:

- Estimate - estimated point-biserial correlation
- LL - lower limit of the confidence interval
- UL - upper limit of the confidence interval

**References**

Bonett DG (2020). "Point-biserial correlation: Interval estimation, hypothesis testing, meta-analysis, and sample size determination." *British Journal of Mathematical and Statistical Psychology*, **73**(S1), 113–144. ISSN 0007-1102, doi: [10.1111/bmsp.12189](https://doi.org/10.1111/bmsp.12189), <https://onlinelibrary.wiley.com/doi/10.1111/bmsp.12189>.

**Examples**

```
ci.pbcor(.05, 28.32, 21.48, 3.81, 3.09, 40, 40)

# Should return:
#           Estimate      LL      UL
# Weighted:  0.7065799 0.5885458 0.7854471
# Unweighted: 0.7020871 0.5808366 0.7828948
```

---

 ci.phi

---

*Confidence interval for a phi coefficient*


---

**Description**

Computes a confidence interval for a phi coefficient. This function requires the frequency counts from a 2 x 2 contingency table for two dichotomous variables. This measure of association assumes that both dichotomous variables are naturally dichotomous.

**Usage**

```
ci.phi(alpha, f00, f01, f10, f11)
```

**Arguments**

alpha	alpha level for 1-alpha confidence
f00	number of participants with y = 0 and x = 0
f01	number of participants with y = 0 and x = 1
f10	number of participants with y = 1 and x = 0
f11	number of participants with y = 1 and x = 1

**Value**

Returns a 1-row matrix. The columns are:

- Estimate - estimate of phi coefficient
- SE - standard error
- LL - lower limit of the confidence interval
- UL - upper limit of the confidence interval

**References**

Bishop YMM, Fienberg SE, Holland PW (1975). *Discrete Multivariate Analysis*. MIT Press.

**Examples**

```
ci.phi(.05, 229, 28, 96, 24)

# Should return:
#   Estimate      SE      LL      UL
# [1,] 0.1229976 0.05746271 0.01037273 0.2356224
```

---

 ci.popsiz

*Confidence interval for an unknown population size*

---

**Description**

Computes a Wald confidence interval for an unknown population size using mark-recapture sampling. This method assumes independence of the two samples. This function requires the frequency counts from an incomplete 2 x 2 contingency table for the two samples (f11 is the unknown number of people who were not observed in either sample). This method sets the estimated odds ratio (with .5 added to each cell) to 1 and solves for unobserved cell frequency.

**Usage**

```
ci.popsiz(alpha, f00, f01, f10)
```

**Arguments**

alpha	alpha level for 1-alpha confidence
f00	number of people observed in both samples
f01	number of people observed in first sample but not second sample
f10	number of people observed in second sample but not first sample

**Value**

Returns a 1-row matrix. The columns are:

- Estimate - estimate of the unknown population size
- LL - lower limit of the confidence interval
- UL - upper limit of the confidence interval

**Examples**

```
ci.popsiize(.05, 794, 710, 741)

# Should return:
#   Estimate  LL  UL
# [1,]    2908 2818 3012
```

---

ci.prop.ps

*Confidence interval for a paired-samples proportion difference*

---

**Description**

Computes an adjusted Wald confidence interval for a difference of proportions in a paired-samples design. This function requires the frequency counts from a 2 x 2 contingency table for two repeated dichotomous measurements.

**Usage**

```
ci.prop.ps(alpha, f00, f01, f10, f11)
```

**Arguments**

alpha	alpha level for 1-alpha confidence
f00	number of participants with y = 0 and x = 0
f01	number of participants with y = 0 and x = 1
f10	number of participants with y = 1 and x = 0
f11	number of participants with y = 1 and x = 1

**Value**

Returns a 1-row matrix. The columns are:

- Estimate - adjusted estimate of proportion difference
- SE - adjusted standard error
- LL - lower limit of the adjusted Wald confidence interval
- UL - upper limit of the adjusted Wald confidence interval

## References

Bonett DG, Price RM (2012). “Adjusted wald confidence interval for a difference of binomial proportions based on paired data.” *Journal of Educational and Behavioral Statistics*, **37**(4), 479–488. ISSN 1076-9986, doi: [10.3102/1076998611411915](https://journals.sagepub.com/doi/10.3102/1076998611411915), <https://journals.sagepub.com/doi/10.3102/1076998611411915>.

## Examples

```
ci.prop.ps(.05, 12, 26, 4, 6)

# Should return:
#   Estimate      SE      LL      UL
# [1,] 0.4583333 0.09448809 0.2548067 0.6251933
```

---

ci.prop1	<i>Confidence interval for a single proportion</i>
----------	--

---

## Description

Computes adjusted Wald and Wilson confidence intervals for a single population proportion. The Wilson confidence interval uses a continuity correction.

## Usage

```
ci.prop1(alpha, f, n)
```

## Arguments

alpha	alpha level for 1-alpha confidence
f	number of participants who have the attribute
n	sample size

## Value

Returns a 2-row matrix. The columns of row 1 are:

- Estimate - adjusted estimate of proportion
- SE - adjusted standard error
- LL - lower limit of the adjusted Wald confidence interval
- UL - upper limit of the adjusted Wald confidence interval

The columns of row 2 are:

- Estimate - ML estimate of proportion
- SE - standard error
- LL - lower limit of the Wilson confidence interval
- UL - upper limit of the Wilson confidence interval

## References

Agresti A, Coull BA (1998). “Approximate is better than ‘exact’ for interval estimation of binomial proportions.” *The American Statistician*, **52**(2), 119–126. ISSN 0003-1305, doi: [10.1080/00031305.1998.10480550](https://doi.org/10.1080/00031305.1998.10480550), <https://www.tandfonline.com/doi/abs/10.1080/00031305.1998.10480550>.

## Examples

```
ci.prop1(.05, 12, 100)

# Should return:
#           Estimate      SE      LL      UL
# Adjusted Wald 0.1346154 0.03346842 0.06901848 0.2002123
# Wilson with cc 0.1200000 0.03249615 0.06625153 0.2039772
```

---

<code>ci.prop2</code>	<i>Confidence interval for a 2-group proportion difference</i>
-----------------------	--

---

## Description

Computes an adjusted Wald confidence interval for a proportion difference in a 2-group design.

## Usage

```
ci.prop2(alpha, f1, f2, n1, n2)
```

## Arguments

alpha	alpha level for 1-alpha confidence
f1	number of participants in group 1 who have the attribute
f2	number of participants in group 2 who have the attribute
n1	sample size of group 1
n2	sample size of group 2

## Value

Returns a 1-row matrix. The columns are:

- Estimate - adjusted estimate of proportion difference
- SE - adjusted standard error
- LL - lower limit of the adjusted Wald confidence interval
- UL - upper limit of the adjusted Wald confidence interval



## References

Agresti A, Caffo B (2000). "Simple and effective confidence intervals for proportions and differences of proportions result from adding two successes and two failures." *The American Statistician*, 54(4), 280. ISSN 00031305, doi: [10.2307/2685779](https://doi.org/10.2307/2685779), <https://www.jstor.org/stable/2685779?origin=crossref>.

## Examples

```
ci.prop2(.05, 35, 21, 150, 150)

# Should return:
#      Estimate      SE      LL      UL
# [1,] 0.09210526 0.04476077 0.004375769 0.1798348
```

---

ci.random.anova1	<i>Confidence intervals for parameters of one-way random effects ANOVA</i>
------------------	--

---

## Description

Computes estimates and confidence intervals for four parameters of the one-way random effects ANOVA: 1) the superpopulation grand mean, 2) the square-root within-group variance component, 3) the square-root between-group variance component, and 4) the omega-squared coefficient. This function assumes equal sample sizes.

## Usage

```
ci.random.anova1(alpha, m, sd, n)
```

## Arguments

alpha	1 - alpha confidence
m	vector of group sample means
sd	vector of group standard deviations
n	sample size per group

## Value

Returns a 4-row matrix. The rows are:

- Grand mean - the mean of the superpopulation of means
- Within SD - the square-root within-group variance component
- Between SD - the square-root between-group variance component
- Omega-squared - the omega-squared coefficient

The columns are:

- Estimate - estimate of parameter
- LL - lower limit of the confidence interval
- UL - upper limit of the confidence interval

### Examples

```
m <- c(56.1, 51.2, 60.3, 68.2, 48.9, 70.5)
sd <- c(9.45, 8.79, 9.71, 8.90, 8.31, 9.75)
ci.random.anova1(.05, m, sd, 20)

# Should return:
#           Estimate      LL      UL
# Grand mean 59.200000 49.9363896 68.4636104
# Within SD:  9.166782  8.0509046 10.4373219
# Between SD:  8.585948  8.3239359  8.8562078
# Omega-squared: 0.467317 0.2284142 0.8480383
```

---

ci.ratio.mad.ps

*Confidence interval for a paired-sample MAD ratio*

---

### Description

Computes a confidence interval for a ratio of population MADs (mean absolute deviation from median) in a paired-samples design.

### Usage

```
ci.ratio.mad.ps(alpha, y1, y2)
```

### Arguments

alpha	alpha level for 1-alpha confidence
y1	vector of measurement 1 scores
y2	vector of measurement 2 scores

### Value

Returns a 1-row matrix. The columns are:

- MAD1 - estimated MAD for measurement 1
- MAD2 - estimated MAD for measurement 2
- MAD1/MAD2 - estimate of MAD ratio
- LL - lower limit of the confidence interval
- UL - upper limit of the confidence interval

## References

Bonett DG, Seier E (2003). "Statistical inference for a ratio of dispersions using paired samples." *Journal of Educational and Behavioral Statistics*, **28**(1), 21–30. ISSN 1076-9986, doi: [10.3102/10769986028001021](https://doi.org/10.3102/10769986028001021), <https://journals.sagepub.com/doi/10.3102/10769986028001021>.

## Examples

```
y2 <- c(21, 4, 9, 12, 35, 18, 10, 22, 24, 1, 6, 8, 13, 16, 19)
y1 <- c(67, 28, 30, 28, 52, 40, 25, 37, 44, 10, 14, 20, 28, 40, 51)
ci.ratio.mad.ps(.05, y1, y2)

# Should return:
#           MAD1  MAD2  MAD1/MAD2      LL      UL
# [1,] 12.71429   7.5   1.695238 1.109176 2.590961
```

---

ci.ratio.mad2	<i>Confidence interval for a 2-group MAD ratio</i>
---------------	--

---

## Description

Computes a confidence interval for a ratio of population MADs (mean absolute deviation from median) in a 2-group design.

## Usage

```
ci.ratio.mad2(alpha, y1, y2)
```

## Arguments

alpha	alpha level for 1-alpha confidence
y1	vector of scores for group 1
y2	vector of scores for group 2

## Value

Returns a 1-row matrix. The columns are:

- MAD1 - estimated MAD from group 1
- MAD2 - estimated MAD from group 2
- MAD1/MAD2 - estimate of MAD ratio
- LL - lower limit of the confidence interval
- UL - upper limit of the confidence interval

## References

Bonett DG, Seier E (2003). “Confidence intervals for mean absolute deviations.” *The American Statistician*, **57**(4), 233–236. ISSN 0003-1305, doi: [10.1198/0003130032323](https://doi.org/10.1198/0003130032323), <https://www.tandfonline.com/doi/abs/10.1198/0003130032323>.

## Examples

```
y1 <- c(32, 39, 26, 35, 43, 27, 40, 37, 34, 29)
y2 <- c(36, 44, 47, 42, 49, 39, 46, 31, 33, 48)
ci.ratio.mad2(.05, y1, y2)

# Should return:
#           MAD1      MAD2  MAD1/MAD2      LL      UL
# [1,] 5.111111 5.888889  0.8679245 0.4520879 1.666253
```

---

ci.ratio.mean.ps      *Confidence interval for a paired-samples mean ratio*

---

## Description

Compute a confidence interval for a ratio of population means of ratio-scale measurements in a paired-samples design. Equality of variances is not assumed.

## Usage

```
ci.ratio.mean.ps(alpha, y1, y2)
```

## Arguments

alpha	alpha level for 1-alpha confidence
y1	vector of measurement 1 scores
y2	vector of measurement 2 scores

## Value

Returns a 1-row matrix. The columns are:

- Mean1 - estimated measurement 1 mean
- Mean2 - estimated measurement 2 mean
- Mean1/Mean2 - estimate of mean ratio
- LL - lower limit of the confidence interval
- UL - upper limit of the confidence interval

## References

Bonett DG, Price RM (2020). “Confidence intervals for ratios of means and medians.” *Journal of Educational and Behavioral Statistics*, **45**(6), 750–770. ISSN 1076-9986, doi: [10.3102/1076998620934125](https://doi.org/10.3102/1076998620934125), <https://journals.sagepub.com/doi/10.3102/1076998620934125>.

## Examples

```
y1 <- c(3.3, 3.6, 3.0, 3.1, 3.9, 4.2, 3.5, 3.3)
y2 <- c(3.0, 3.1, 2.7, 2.6, 3.2, 3.8, 3.2, 3.0)
ci.ratio.mean.ps(.05, y1, y2)

# Should return:
#      Mean1 Mean2 Mean1/Mean2      LL      UL
# [1,] 3.4875 3.075    1.134146 1.09417 1.175583
```

---

<code>ci.ratio.mean2</code>	<i>Confidence interval for a 2-group mean ratio</i>
-----------------------------	---

---

## Description

Computes a confidence interval for a ratio of population means of ratio-scale measurements in a 2-group design. Equality of variances is not assumed.

## Usage

```
ci.ratio.mean2(alpha, y1, y2)
```

## Arguments

<code>alpha</code>	alpha level for 1-alpha confidence
<code>y1</code>	vector of scores for group 1
<code>y2</code>	vector of scores for group 2

## Value

Returns a 1-row matrix. The columns are:

- Mean1 - estimated mean from group 1
- Mean2 - estimated mean from group 2
- Mean1/Mean2- estimated mean ratio
- LL - lower limit of the confidence interval
- UL - upper limit of the confidence interval

## References

Bonett DG, Price RM (2020). “Confidence intervals for ratios of means and medians.” *Journal of Educational and Behavioral Statistics*, **45**(6), 750–770. ISSN 1076-9986, doi: [10.3102/1076998620934125](https://doi.org/10.3102/1076998620934125), <https://journals.sagepub.com/doi/10.3102/1076998620934125>.

## Examples

```
y2 <- c(32, 39, 26, 35, 43, 27, 40, 37, 34, 29, 49, 42, 40)
y1 <- c(36, 44, 47, 42, 49, 39, 46, 31, 33, 48)
ci.ratio.mean2(.05, y1, y2)

# Should return:
#
#      Mean1   Mean2 Mean1/Mean2      LL      UL
# [1,]  41.5  36.38462    1.140592 0.9897482 1.314425
```

---

ci.ratio.median.ps      *Confidence interval for a paired-samples median ratio*

---

## Description

Computes a confidence interval for a ratio of population medians in a paired-samples design.

## Usage

```
ci.ratio.median.ps(alpha, y1, y2)
```

## Arguments

alpha	alpha level for 1-alpha confidence
y1	vector of scores for measurement 1
y2	vector of scores for measurement 2

## Value

Returns a 1-row matrix. The columns are:

- Median1 - estimated median from measurement 1
- Median2 - estimated median from measurement 2
- Median1/Median2 - estimated ratio of medians
- LL - lower limit of the confidence interval
- UL - upper limit of the confidence interval

## References

Bonett DG, Price RM (2020). “Confidence intervals for ratios of means and medians.” *Journal of Educational and Behavioral Statistics*, **45**(6), 750–770. ISSN 1076-9986, doi: [10.3102/1076998620934125](https://doi.org/10.3102/1076998620934125), <https://journals.sagepub.com/doi/10.3102/1076998620934125>.

## Examples

```
y1 <- c(21, 4, 9, 12, 35, 18, 10, 22, 24, 1, 6, 8, 13, 16, 19)
y2 <- c(67, 28, 30, 28, 52, 40, 25, 37, 44, 10, 14, 20, 28, 40, 51)
ci.ratio.median.ps(.05, y1, y2)

# Should return:
#           Median1 Median2 Median1/Median2      LL      UL
# [1,]           13      30      0.4333333 0.3094838 0.6067451
```

---

<code>ci.ratio.median2</code>	<i>Confidence interval for a 2-group median ratio</i>
-------------------------------	---

---

## Description

Computes a confidence interval for a ratio of population medians of ratio-scale measurements in a 2-group design.

## Usage

```
ci.ratio.median2(alpha, y1, y2)
```

## Arguments

<code>alpha</code>	alpha level for 1-alpha confidence
<code>y1</code>	vector of scores for group 1
<code>y2</code>	vector of scores for group 2

## Value

Returns a 1-row matrix. The columns are:

- Median1 - estimated median from group 1
- Median2 - estimated median from group 2
- Median1/Median2 - estimated ratio of medians
- LL - lower limit of the confidence interval
- UL - upper limit of the confidence interval

## References

Bonett DG, Price RM (2020). “Confidence intervals for ratios of means and medians.” *Journal of Educational and Behavioral Statistics*, **45**(6), 750–770. ISSN 1076-9986, doi: [10.3102/1076998620934125](https://doi.org/10.3102/1076998620934125), <https://journals.sagepub.com/doi/10.3102/1076998620934125>.

## Examples

```
y2 <- c(32, 39, 26, 35, 43, 27, 40, 37, 34, 29, 49, 42, 40)
y1 <- c(36, 44, 47, 42, 49, 39, 46, 31, 33, 48)
ci.ratio.median2(.05, y1, y2)

# Should return:
#   Median1 Median2 Median1/Median2      LL      UL
# [1,]      43      37      1.162162 0.927667 1.455933
```

---

ci.ratio.prop.ps      *Confidence interval for a paired-samples proportion ratio*

---

## Description

Computes a confidence interval for a ratio of proportions in a paired-samples design. This function requires the frequency counts from a 2 x 2 contingency table for two repeated dichotomous measurements.

## Usage

```
ci.ratio.prop.ps(alpha, f00, f10, f01, f11)
```

## Arguments

alpha	alpha level for 1-alpha confidence
f00	number of participants with y = 0 and x = 0
f10	number of participants with y = 1 and x = 0
f01	number of participants with y = 0 and x = 1
f11	number of participants with y = 1 and x = 1

## Value

Returns a 1-row matrix. The columns are:

- Estimate - estimate of proportion ratio
- LL - lower limit of the confidence interval
- UL - upper limit of the confidence interval



## References

Bonett DG, Price RM (2006). “Confidence intervals for a ratio of binomial proportions based on paired data.” *Statistics in Medicine*, **25**(17), 3039–3047. ISSN 0277-6715, doi: [10.1002/sim.2440](https://onlinelibrary.wiley.com/doi/10.1002/sim.2440), <https://onlinelibrary.wiley.com/doi/10.1002/sim.2440>.

## Examples

```
ci.ratio.prop.ps(.05, 12, 26, 4, 6)

# Should return:
#   Estimate      LL      UL
# [1,]  2.375 1.537157 3.669518
```

---

ci.ratio.prop2	<i>Confidence interval for a 2-group proportion ratio</i>
----------------	---

---

## Description

Computes an adjusted Wald confidence interval for a proportion ratio in a 2-group design.

## Usage

```
ci.ratio.prop2(alpha, f1, f2, n1, n2)
```

## Arguments

alpha	alpha level for 1-alpha confidence
f1	number of participants in group 1 who have the attribute
f2	number of participants in group 2 who have the attribute
n1	sample size of group 1
n2	sample size of group 2

## Value

Returns a 1-row matrix. The columns are:

- Estimate - adjusted ratio of proportion ratio
- LL - lower limit of the adjusted Wald confidence interval
- UL - upper limit of the adjusted Wald confidence interval

## References

Price RM, Bonett DG (2008). “Confidence intervals for a ratio of two independent binomial proportions.” *Statistics in Medicine*, **27**(26), 5497–5508. ISSN 02776715, doi: [10.1002/sim.3376](https://onlinelibrary.wiley.com/doi/10.1002/sim.3376), <https://onlinelibrary.wiley.com/doi/10.1002/sim.3376>.

**Examples**

```
ci.ratio.prop2(.05, 35, 21, 150, 150)

# Should return:
#   Estimate      LL      UL
# [1,] 1.666667 1.017253 2.705025
```

---

ci.slope.prop.bs	<i>Confidence interval for a slope of a quantitative between-subjects factor</i>
------------------	--

---

**Description**

Computes an adjusted Wald confidence interval for the slope of a quantitative between-subjects factor using contrast coefficients.

**Usage**

```
ci.slope.prop.bs(alpha, f, n, x)
```

**Arguments**

alpha	alpha level for 1-alpha confidence
f	vector of frequency counts of participants with attribute
n	vector of sample sizes
x	vector of quantitative factor values

**Value**

Returns a 1-row matrix. The columns are:

- Estimate - adjusted slope estimate
- SE - adjusted standard error
- z - z test statistic
- p - p-value
- LL - lower limit of the adjusted Wald confidence interval
- UL - upper limit of the adjusted Wald confidence interval

**References**

Price RM, Bonett DG (2004). "An improved confidence interval for a linear function of binomial proportions." *Computational Statistics & Data Analysis*, **45**(3), 449–456. ISSN 01679473, doi: [10.1016/S01679473\(03\)000070](https://doi.org/10.1016/S01679473(03)000070), <https://linkinghub.elsevier.com/retrieve/pii/S0167947303000070>.

**Examples**

```
f <- c(14, 27, 38)
n <- c(100, 100, 100)
x <- c(10, 20, 40)
ci.slope.prop.bs(.05, f, n, x)

# Should return:
#      Estimate      SE      z      p      LL      UL
# [1,] 0.007542293 0.002016793 3.739746 0.000184206 0.003589452 0.01149513
```

---

ci.spcor	<i>Confidence interval for a semipartial correlation</i>
----------	--

---

**Description**

Computes a Fisher confidence interval for a population semipartial correlation. This function requires an (unadjusted) estimate of the squared multiple correlation in the full model that contains the predictor variable of interest plus all control variables. This function computes a modified Aloe-Becker confidence interval that uses  $n - 3$  rather than  $n$  in the standard error and also uses a Fisher transformation of the semipartial correlation.

**Usage**

```
ci.spcor(alpha, cor, r2, n)
```

**Arguments**

alpha	alpha level for 1-alpha confidence
cor	sample semipartial correlation
r2	sample squared multiple correlation in full model
n	sample size

**Value**

Returns a 1-row matrix. The columns are:

- Estimate - estimated semipartial correlation
- SE - standard error
- LL - lower limit of the confidence interval
- UL - upper limit of the confidence interval

**References**

Aloe AM, Becker BJ (2012). "An effect size for regression predictors in meta-analysis." *Journal of Educational and Behavioral Statistics*, **37**(2), 278–297. ISSN 1076-9986, doi: [10.3102/1076998610396901](https://doi.org/10.3102/1076998610396901), <https://journals.sagepub.com/doi/10.3102/1076998610396901>.

### Examples

```
ci.spcor(.05, .582, .699, 20)

# Should return:
#   Estimate      SE      LL      UL
# [1,]  0.582 0.1374298 0.2525662 0.7905182
```

---

ci.spear	<i>Confidence interval for a Spearman correlation</i>
----------	---

---

### Description

Computes a Fisher confidence interval for a population Spearman correlation.

### Usage

```
ci.spear(alpha, y, x)
```

### Arguments

alpha	alpha level for 1-alpha confidence
y	vector of y scores
x	vector of x scores

### Value

Returns a 1-row matrix. The columns are:

- Estimate - estimated correlation
- SE - standard error
- LL - lower limit of the confidence interval
- UL - upper limit of the confidence interval

### References

Bonett DG, Wright TA (2000). "Sample size requirements for estimating Pearson, Kendall and Spearman correlations." *Psychometrika*, **65**(1), 23–28. ISSN 0033-3123, doi: [10.1007/BF02294183](https://doi.org/10.1007/BF02294183), <https://link.springer.com/10.1007/BF02294183>.

**Examples**

```

y <- c(21, 4, 9, 12, 35, 18, 10, 22, 24, 1, 6, 8, 13, 16, 19)
x <- c(67, 28, 30, 28, 52, 40, 25, 37, 44, 10, 14, 20, 28, 40, 51)
ci.spear(.05, y, x)

# Should return:
#      Estimate      SE      LL      UL
# [1,] 0.8699639 0.08241326 0.5840951 0.9638297

```

---

ci.spear2

*Confidence interval for a 2-group Spearman correlation difference*


---

**Description**

Computes a confidence interval for a difference in population Spearman correlations in a 2-group design.

**Usage**

```
ci.spear2(alpha, cor1, cor2, n1, n2)
```

**Arguments**

alpha	alpha level for 1-alpha confidence
cor1	sample Spearman correlation in group 1
cor2	sample Spearman correlation in group 2
n1	sample size in group 1
n2	sample size in group 2

**Value**

Returns a 1-row matrix. The columns are:

- Estimate - estimated correlation difference
- LL - lower limit of the confidence interval
- UL - upper limit of the confidence interval

**References**

Bonett DG, Wright TA (2000). "Sample size requirements for estimating Pearson, Kendall and Spearman correlations." *Psychometrika*, **65**(1), 23–28. ISSN 0033-3123, doi: [10.1007/BF02294183](https://doi.org/10.1007/BF02294183), <https://link.springer.com/10.1007/BF02294183>.

Zou GY (2007). "Toward using confidence intervals to compare correlations." *Psychological Methods*, **12**(4), 399–413. ISSN 1939-1463, doi: [10.1037/1082989X.12.4.399](https://doi.org/10.1037/1082989X.12.4.399), [https://doi.org/getdoi.cfm?doi=10.1037/1082-989X.12.4.399](https://doi.org/10.1037/1082-989X.12.4.399).

**Examples**

```
ci.spear2(.05, .54, .48, 180, 200)

# Should return:
#   Estimate      LL      UL
# [1,]    0.06 -0.1003977 0.2185085
```

---

ci.stdmean.ps	<i>Confidence interval for a paired-samples standardized mean difference</i>
---------------	--

---

**Description**

Computes confidence intervals for a population standardized mean difference in a paired-samples design. A square root unweighted variance standardizer and single measurement standard deviation standardizers are used. Equality of variances is not assumed.

**Usage**

```
ci.stdmean.ps(alpha, m1, m2, sd1, sd2, cor, n)
```

**Arguments**

alpha	alpha level for 1-alpha confidence
m1	sample mean of measurement 1
m2	sample mean of measurement 2
sd1	sample standard deviation in measurement 1
sd2	sample standard deviation in measurement 2
cor	sample correlation between measurements
n	sample size

**Value**

Returns a 3-row matrix. The columns are:

- Estimate - bias adjusted standardized mean difference
- SE - standard error
- LL - lower limit of the confidence interval
- UL - upper limit of the confidence interval

**References**

Bonett DG (2008). "Confidence intervals for standardized linear contrasts of means." *Psychological Methods*, **13**(2), 99–109. ISSN 1939-1463, doi: [10.1037/1082989X.13.2.99](https://doi.org/10.1037/1082989X.13.2.99), <https://doi.org/10.1037/1082-989X.13.2.99>.

**Examples**

```
ci.stdmean.ps(.05, 110.4, 102.1, 15.3, 14.6, .75, 25)

# Should return:
#
#           Estimate      SE      LL      UL
# Unweighted standardizer:  0.5433457 0.1609934 0.2394905 0.8705732
# Measurement 1 standardizer: 0.5253526 0.1615500 0.2258515 0.8591158
# Measurement 2 standardizer: 0.5505407 0.1692955 0.2366800 0.9003063
```

---

```
ci.stdmean.strat      Confidence interval for a 2-group standardized mean difference with
                        stratified sampling
```

---

**Description**

Computes confidence intervals for a population standardized mean difference in a 2-group nonexperimental design with stratified random sampling using a square root weighted variance standardizer or single group standard deviation standardizer. Equality of variances is not assumed.

**Usage**

```
ci.stdmean.strat(alpha, m1, m2, sd1, sd2, n1, n2, p1)
```

**Arguments**

alpha	alpha level for 1-alpha confidence
m1	sample mean in group 1
m2	sample mean in group 2
sd1	sample standard deviation in group 1
sd2	sample standard deviation in group 2
n1	sample size in group 1
n2	sample size in group 2
p1	proportion of total population in subpopulation 1

**Value**

Returns a 3-row matrix. The columns are:

- Estimate - bias adjusted standardized mean difference
- SE - standard error
- LL - lower limit of the confidence interval
- UL - upper limit of the confidence interval

## References

Bonett DG (2020). “Point-biserial correlation: Interval estimation, hypothesis testing, meta-analysis, and sample size determination.” *British Journal of Mathematical and Statistical Psychology*, **73**(S1), 113–144. ISSN 0007-1102, doi: [10.1111/bmsp.12189](https://doi.org/10.1111/bmsp.12189), <https://onlinelibrary.wiley.com/doi/10.1111/bmsp.12189>.

## Examples

```
ci.stdmean.strat(.05, 30.2, 30.8, 10.5, 11.2, 200, 200, .533)

# Should return:
#               Estimate      SE      LL      UL
# Weighted standardizer: -0.05528428 0.10023259 -0.2518410 0.1410636
# Group 1 standardizer:  -0.05692722 0.10368609 -0.2603639 0.1460782
# Group 2 standardizer:  -0.05692722 0.09720571 -0.2440911 0.1369483
```

---

ci.stdmean1	<i>Confidence interval for a single standardized mean</i>
-------------	---

---

## Description

Computes a confidence interval for a population standardized mean difference from a hypothesized value. If the hypothesized value is set to 0, the reciprocals of the confidence interval endpoints gives a confidence interval for the coefficient of variation.

## Usage

```
ci.stdmean1(alpha, m, sd, n, h)
```

## Arguments

alpha	alpha level for 1-alpha confidence
m	sample mean
sd	sample standard deviation
n	sample size
h	hypothesized value

## Value

Returns a 1-row matrix. The columns are:

- Estimate - bias adjusted standardized mean difference
- SE - standard error
- LL - lower limit of the confidence interval
- UL - upper limit of the confidence interval



## References

Bonett DG (2008). "Confidence intervals for standardized linear contrasts of means." *Psychological Methods*, **13**(2), 99–109. ISSN 1939-1463, doi: [10.1037/1082989X.13.2.99](https://doi.org/10.1037/1082989X.13.2.99), [https://doi.apa.org/getdoi.cfm?doi=10.1037/1082-989X.13.2.99](https://doi.org/10.1037/1082-989X.13.2.99).

## Examples

```
ci.stdmean1(.05, 24.5, 3.65, 40, 20)

# Should return:
#   Estimate      SE      LL      UL
# [1,] 1.209015  0.2124335  0.8165146  1.649239
```

---

ci.stdmean2	<i>Confidence interval for a 2-group standardized mean difference</i>
-------------	---

---

## Description

Computes confidence intervals for a population standardized mean difference. Unweighted, weighted, and single group variance standardizers are used. The square root weighted variance standardizer is recommended in 2-group nonexperimental designs with simple random sampling. The square root unweighted variance standardizer is recommended in 2-group experimental designs. The single group standard deviation standardizer can be used with experimental or nonexperimental designs. Equality of variances is not assumed.

## Usage

```
ci.stdmean2(alpha, m1, m2, sd1, sd2, n1, n2)
```

## Arguments

alpha	alpha level for 1-alpha confidence
m1	sample mean in group 1
m2	sample mean in group 2
sd1	sample standard deviation in group 1
sd2	sample standard deviation in group 2
n1	sample size in group 1
n2	sample size in group 2

**Value**

Returns a 4-row matrix. The columns are:

- Estimate - bias adjusted standardized mean difference
- SE - standard error
- LL - lower limit of the confidence interval
- UL - upper limit of the confidence interval

**References**

Bonett DG (2008). "Confidence intervals for standardized linear contrasts of means." *Psychological Methods*, **13**(2), 99–109. ISSN 1939-1463, doi: [10.1037/1082989X.13.2.99](https://doi.org/10.1037/1082989X.13.2.99), <https://doi.org/10.1037/1082-989X.13.2.99>.

**Examples**

```
ci.stdmean2(.05, 35.1, 26.7, 7.32, 6.98, 30, 30)

# Should return:
#
# Unweighted standardizer:  Estimate      SE      LL      UL
# Weighted standardizer:   1.159240  0.2844012  0.6170771  1.731909
# Group 1 standardizer:    1.159240  0.2802826  0.6251494  1.723837
# Group 2 standardizer:    1.117605  0.2975582  0.5643375  1.730744
# Group 2 standardizer:    1.172044  0.3120525  0.5918268  1.815050
```

---

ci.tetra

*Confidence interval for a tetrachoric correlation*

---

**Description**

Computes a confidence interval for an approximation to the tetrachoric correlation. This function requires the frequency counts from a 2 x 2 contingency table for two dichotomous variables. This measure of association assumes both of the dichotomous variables are artificially dichotomous.

**Usage**

```
ci.tetra(alpha, f00, f01, f10, f11)
```

**Arguments**

alpha	alpha level for 1-alpha confidence
f00	number of participants with y = 0 and x = 0
f01	number of participants with y = 0 and x = 1
f10	number of participants with y = 1 and x = 0
f11	number of participants with y = 1 and x = 1

**Value**

Returns a 1-row matrix. The columns are:

- Estimate - estimate of tetrachoric approximation
- LL - lower limit of the confidence interval
- UL - upper limit of the confidence interval

**References**

Bonett DG, Price RM (2005). "Inferential methods for the tetrachoric correlation coefficient." *Journal of Educational and Behavioral Statistics*, **30**(2), 213–225. ISSN 1076-9986, doi: [10.3102/10769986030002213](https://journals.sagepub.com/doi/10.3102/10769986030002213), <https://journals.sagepub.com/doi/10.3102/10769986030002213>.

**Examples**

```
ci.tetra(.05, 46, 15, 54, 85)

# Should return:
#   Estimate      LL      UL
# [1,] 0.5135167 0.3102345 0.6748546
```

---

ci.tukey	<i>Tukey-Kramer confidence intervals for all pairwise mean differences in a betwen-subjects design</i>
----------	--

---

**Description**

Computes heteroscedastic Tukey-Kramer (also known as Games-Howell) confidence intervals for all pairwise comparisons of population means using sample means, sample standard deviations, and samples sizes as input. A Satterthwaite adjustment to the degrees of freedom is used to improve the accuracy of the confidence intervals.

**Usage**

```
ci.tukey(alpha, m, sd, n)
```

**Arguments**

alpha	alpha level for simultaneous 1-alpha confidence
m	vector of group sample means
sd	vector of group sample standard deviations
n	vector of sample sizes

**Value**

Returns a matrix with the number of rows equal to the number of pairwise comparisons. The columns are:

- Estimate - estimated mean difference
- SE - standard error
- t - t test statistic
- df - degrees of freedom
- p - p-value
- LL - lower limit of the confidence interval
- UL - upper limit of the confidence interval

**References**

Games PA, Howell JF (1976). "Pairwise multiple comparison procedures with unequal N's and/or variances: A Monte Carlo study." *Journal of Educational Statistics*, 1(2), 113. ISSN 03629791, doi: [10.2307/1164979](https://doi.org/10.2307/1164979), <https://www.jstor.org/stable/1164979?origin=crossref>.

**Examples**

```
m <- c(12.86, 17.57, 26.29)
sd <- c(3.185, 3.995, 3.773)
n <- c(20, 20, 20)
ci.tukey(.05, m, sd, n)

# Should return:
#   Estimate      SE      t      p      df      LL      UL
# 1 2      -4.71  1.142459 -4.122686 0.001 36.20303 -7.501842 -1.918158
# 1 3     -13.43  1.104078 -12.163998 0.000 36.95915 -16.125713 -10.734287
# 2 3      -8.72  1.228730 -7.096758 0.000 37.87646 -11.717053 -5.722947
```

---

ci.var.upper

*Upper confidence limit of variance*

---

**Description**

Computes an upper limit for a population variance using a sample variance from a sample of size n in a prior study. The upper limit can be used as a variance planning value in sample size functions that require a variance planning value.

**Usage**

```
ci.var.upper(alpha, var, n)
```

**Arguments**

alpha	alpha value for 1-alpha confidence (one-sided)
var	sample variance
n	sample size

**Value**

Returns an upper limit (UL) variance planning value

**Examples**

```
ci.var.upper(.25, 15, 60)

# Should return:
#           UL
# [1,] 17.23264
```

---

 ci.yule

---

*Confidence interval for Yule's Q*


---

**Description**

Computes a confidence interval for Yule's Q measure of association using a transformation of a confidence interval for an odds ratio with .5 added to each cell frequency. This function requires the frequency counts from a 2 x 2 contingency table for two dichotomous variables.

**Usage**

```
ci.yule(alpha, f00, f01, f10, f11)
```

**Arguments**

alpha	alpha level for 1-alpha confidence
f00	number of participants with y = 0 and x = 0
f01	number of participants with y = 0 and x = 1
f10	number of participants with y = 1 and x = 0
f11	number of participants with y = 1 and x = 1

**Value**

Returns a 1-row matrix. The columns are:

- Estimate - estimate of Yule's Q
- LL - lower limit of the confidence interval
- UL - upper limit of the confidence interval

**Examples**

```
ci.yule(.05, 229, 28, 96, 24)

# Should return:
#   Estimate      LL      UL
# [1,] 0.343067 0.06247099 0.573402
```

---

etasqr.adj	<i>Bias adjusts an eta-squared estimate</i>
------------	---

---

**Description**

Computes an approximate bias adjustment for eta-squared. This adjustment can be applied to eta-squared, partial-eta squared, and generalized eta-squared estimates.

**Usage**

```
etasqr.adj(etasqr, dfeffect, dferror)
```

**Arguments**

etasqr	unadjusted eta-square estimate
dfeffect	degrees of freedom for the effect
dferror	error degrees of freedom

**Value**

Returns a bias adjusted eta-squared estimate

**Examples**

```
etasqr.adj(.315, 2, 42)

# Should return:
#   Adjusted eta-squared
# [1,]                0.282381
```

---

iqv *Indices of qualitative variation*

---

**Description**

Computes the Shannon, Berger, and Simpson indices of qualitative variation.

**Usage**

```
iqv(f)
```

**Arguments**

f                    vector of multinomial frequency counts

**Value**

Returns estimates of the Shannon, Berger, and Simpson qualitative indices

**Examples**

```
f <- c(10, 46, 15, 3)
iqv(f)
# Should return:
#        Simpson    Berger    Shannon
# [1,] 0.7367908 0.5045045        0.7
```

---

pi.score1 *Prediction interval for one score*

---

**Description**

Computes a prediction interval for the response variable score of one randomly selected member from the study population.

**Usage**

```
pi.score1(alpha, m, sd, n)
```

**Arguments**

alpha                alpha level for 1-alpha confidence  
m                    sample mean  
sd                    sample standard deviation  
n                    sample size

**Value**

Returns a 1-row matrix. The columns are:

- Predicted - predicted score
- df - degrees of freedom
- LL - lower limit of the prediction interval
- UL - upper limit of the prediction interval

**Examples**

```
pi.score1(.05, 24.5, 3.65, 40)

# Should return:
#   Predicted df      LL      UL
# [1,]      24.5 39 17.02546 31.97454
```

---

pi.score2

*Prediction interval for a difference of scores*

---

**Description**

For a 2-group experimental design, this function computes a prediction interval for how the response variable score for one randomly selected person from the study population would differ under the two treatment conditions. Both equal variance and unequal variance prediction intervals are computed.

**Usage**

```
pi.score2(alpha, m1, m2, sd1, sd2, n1, n2)
```

**Arguments**

alpha	alpha level for 1-alpha confidence
m1	sample mean for group 1
m2	sample mean for group 1
sd1	sample standard deviation for group 1
sd2	sample standard deviation for group 2
n1	sample size for group 1
n2	sample size for group 2



**Value**

Returns a 2-row matrix. The columns are:

- Predicted - predicted difference in scores
- df - degrees of freedom
- LL - lower limit of the prediction interval
- UL - upper limit of the prediction interval

**References**

Hahn GJ (1977). "A prediction interval on the difference between two future sample means and its application to a claim of product superiority." *Technometrics*, **19**(2), 131–134. ISSN 0040-1706, doi: [10.1080/00401706.1977.10489520](https://doi.org/10.1080/00401706.1977.10489520), <https://www.tandfonline.com/doi/abs/10.1080/00401706.1977.10489520>.

**Examples**

```
pi.score2(.05, 29.57, 18.35, 2.68, 1.92, 40, 45)

# Should return:
#               Predicted      df      LL      UL
# Equal Variances Assumed:    11.22 83.00000 4.650454 17.78955
# Equal Variances Not Assumed: 11.22 72.34319 4.603642 17.83636
```

---

random.sample	<i>Generate a random sample</i>
---------------	---------------------------------

---

**Description**

Generates a random sample of participant IDs.

**Usage**

```
random.sample(popsiz, samsiz)
```

**Arguments**

popsiz	study population size
samsiz	sample size

**Value**

Returns a vector of randomly generated participant IDs

**Examples**

```
random.sample(3000, 25)

# Should return:
# [1] 37 94 134 186 212 408 485 697 722 781 998 1055
# [13] 1182 1224 1273 1335 1452 1552 1783 1817 2149 2188 2437 2850 2936
```

---

random.y	<i>Generate random sample of scores</i>
----------	---

---

**Description**

Generates a random sample of scores from a normal distribution with a specified population mean and standard deviation.

**Usage**

```
random.y(n, m, sd, min, max, dec)
```

**Arguments**

n	sample size
m	population mean of scores
sd	population standard deviation of scores
min	minimum allowable value
max	maximum allowable value
dec	number of decimal points

**Value**

Returns a vector of randomly generated scores.

**Examples**

```
random.y(10, 3.6, 2.8, 1, 7, 0)

# Should return:
# [1] 2 7 7 1 6 3 1 3 2 1
```

---

random.yx	<i>Generates random bivariate scores</i>
-----------	--

---

**Description**

Generates a random sample of y scores and x scores from a bivariate normal distributions with specified population means, standard deviations, and correlation.

**Usage**

```
random.yx(n, my, mx, sdy, sdx, cor, dec)
```

**Arguments**

n	sample size
my	population mean of y scores
mx	population mean of x scores
sdy	population standard deviation of y scores
sdx	population standard deviation of x scores
cor	population correlation between x and y
dec	number of decimal points

**Value**

Returns n pairs of y and x scores

**Examples**

```
random.yx(10, 50, 20, 4, 2, .5, 1)

# Should return:
#      y    x
# 1  50.3 21.6
# 2  52.0 21.6
# 3  53.0 22.7
# 4  46.9 21.3
# 5  56.3 23.8
# 6  50.4 20.3
# 7  44.6 19.9
# 8  49.9 18.3
# 9  49.4 18.5
# 10 42.3 20.2
```

---

randomize	<i>Randomize a sample into groups</i>
-----------	---------------------------------------

---

**Description**

Randomly assigns a sample of participants into k groups.

**Usage**

```
randomize(n)
```

**Arguments**

n                      vector of sample sizes per group

**Value**

Returns a vector of randomly generated group assignments.

**Examples**

```
n <- c(10, 10, 5)
randomize(n)

# Should return:
# [1] 2 3 2 1 1 2 3 3 2 1 2 1 3 1 1 2 3 1 1 2 2 1 1 2 2
```

---

size.ci.agree	<i>Sample size for a G-index confidence interval</i>
---------------	--

---

**Description**

Computes the sample size required to estimate a G-index of agreement for two dichotomous ratings with desired confidence interval precision. Set the G-index planning value to the smallest value within a plausible range for a conservatively large sample size.

**Usage**

```
size.ci.agree(alpha, G, w)
```

**Arguments**

alpha                  alpha level for 1-alpha confidence  
 G                      planning value of G-index  
 w                      desired confidence interval width

**Value**

Returns the required sample size

**Examples**

```
size.ci.agree(.05, .8, .2)

# Should return:
#   Sample size
# [1,]        139
```

---

size.ci.condmean	<i>Sample size for a conditional mean confidence interval</i>
------------------	---

---

**Description**

Computes the total sample size required to estimate a conditional mean of  $y$  at  $x = x^*$  in a fixed- $x$  simple linear regression model with desired confidence interval precision. In an experimental design, the total sample size would be allocated to the levels of the quantitative factor and it might be necessary to increase the total sample size to achieve equal sample sizes. Set the error variance planning value to the largest value within a plausible range for a conservatively large sample size.

**Usage**

```
size.ci.condmean(alpha, evar, xvar, diff, w)
```

**Arguments**

alpha	alpha level for 1-alpha confidence
evar	planning value of within group (error) variance
xvar	variance of fixed predictor variables
diff	difference between $x^*$ and mean of $x$
w	desired confidence interval width

**Value**

Returns the required total sample size

**Examples**

```
size.ci.condmean(.05, 120, 125, 15, 5)

# Should return:
#   Total sample size
# [1,]            210
```

---

`size.ci.cor`*Sample size for a Pearson or partial correlation confidence interval*

---

### Description

Computes the sample size required to estimate a Pearson or partial correlation with desired confidence interval precision. Set  $s = 0$  for a Pearson correlation. Set the correlation planning value to the smallest value within a plausible range for a conservatively large sample size.

### Usage

```
size.ci.cor(alpha, cor, s, w)
```

### Arguments

<code>alpha</code>	alpha level for 1-alpha confidence
<code>cor</code>	planning value of correlation
<code>s</code>	number of control variables
<code>w</code>	desired confidence interval width

### Value

Returns the required sample size

### References

Bonett DG, Wright TA (2000). "Sample size requirements for estimating Pearson, Kendall and Spearman correlations." *Psychometrika*, **65**(1), 23–28. ISSN 0033-3123, doi: [10.1007/BF02294183](https://doi.org/10.1007/BF02294183), <https://link.springer.com/10.1007/BF02294183>.

### Examples

```
size.ci.cor(.05, .362, 0, .25)

# Should return:
#   Sample size
# [1,]         188
```

---

size.ci.cronbach	<i>Sample size for a Cronbach reliability confidence interval</i>
------------------	---

---

**Description**

Computes the sample size required to estimate a Cronbach reliability with desired confidence interval precision.

**Usage**

```
size.ci.cronbach(alpha, rel, a, w)
```

**Arguments**

alpha	alpha value for 1-alpha confidence
rel	reliability planning value
a	number of measurements
w	desired confidence interval width

**Value**

Returns the required sample size

**References**

Bonett DG, Wright TA (2015). "Cronbach's alpha reliability: Interval estimation, hypothesis testing, and sample size planning." *Journal of Organizational Behavior*, **36**(1), 3–15. ISSN 08943796, doi: [10.1002/job.1960](https://onlinelibrary.wiley.com/doi/10.1002/job.1960), <https://onlinelibrary.wiley.com/doi/10.1002/job.1960>.

**Examples**

```
size.ci.cronbach(.05, .85, 5, .1)

# Should return:
#   Sample size
# [1,]         89
```

---

size.ci.lc.ancova	<i>Sample size for a mean linear contrast confidence interval in an AN-COVA</i>
-------------------	---

---

### Description

Computes the sample size for each group required to estimate a linear contrast of means in an AN-COVA model with desired confidence interval precision. In a nonexperimental design, the sample size is affected by the magnitude of covariate mean differences across groups. The covariate mean differences can be approximated by specifying the largest standardized covariate mean difference across all pairwise group differences and for all covariates. In an experiment, this standardized mean difference should be set to 0. Set the error variance planning value to the largest value within a plausible range for a conservatively large sample size.

### Usage

```
size.ci.lc.ancova(alpha, evar, s, d, w, v)
```

### Arguments

alpha	alpha level for 1-alpha confidence
evar	planning value of within group (error) variance
s	number of covariates
d	largest standardized mean difference for all covariates
w	desired confidence interval width
v	vector of between-subjects contrast coefficients

### Value

Returns the required sample size per group

### Examples

```
v <- c(1, -1)
size.ci.lc.ancova(.05, 1.37, 1, 0, 1.5, v)

# Should return:
#   Sample size per group
# [1,]                21
```



---

size.ci.lc.mean.bs	<i>Sample size for a between-subjects mean linear contrast confidence interval</i>
--------------------	--

---

**Description**

Computes the sample size in each group (assuming equal sample sizes) required to estimate a linear contrast of population means with desired confidence interval precision in a between-subjects design. Set the variance planning value to the largest value within a plausible range for a conservatively large sample size.

**Usage**

```
size.ci.lc.mean.bs(alpha, var, w, v)
```

**Arguments**

alpha	alpha level for 1-alpha confidence
var	planning value of average within-group variance
w	desired confidence interval width
v	vector of between-subjects contrast coefficients

**Value**

Returns the required sample size for each group

**Examples**

```
v <- c(.5, .5, -1)
size.ci.lc.mean.bs(.05, 5.62, 2.0, v)

# Should return:
#   Sample size per group
# [1,]                34
```

---

size.ci.lc.mean.ws	<i>Sample size for a within-subjects mean linear contrast confidence interval</i>
--------------------	---

---

**Description**

Computes the sample size required to estimate a linear contrast of population means with desired confidence interval precision in a within-subjects design. Set the variance planning value to the largest value within a plausible range for a conservatively large sample size. Set the correlation planning value to the smallest value within a plausible range for a conservatively large sample size.

**Usage**

```
size.ci.lc.mean.ws(alpha, var, cor, w, q)
```

**Arguments**

alpha	alpha level for 1-alpha confidence
var	planning value of average variance of the measurements
cor	planning value of average correlation between measurements
w	desired confidence interval width
q	vector of within-subjects contrast coefficients

**Value**

Returns the required sample size

**Examples**

```
q <- c(.5, .5, -.5, -.5)
size.ci.lc.mean.ws(.05, 265, .8, 10, q)

# Should return:
#   Sample size
# [1,]       11
```

---

size.ci.lc.prop.bs	<i>Sample size for a between-subjects proportion linear contrast confidence interval</i>
--------------------	--

---

**Description**

Computes the sample size in each group required to estimate a linear contrast of proportions with desired confidence interval precision in a between-subjects design. Set the proportion planning values to .5 for a conservatively large sample size.

**Usage**

```
size.ci.lc.prop.bs(alpha, p, w, v)
```

**Arguments**

alpha	alpha level for 1-alpha confidence
p	vector of proportion planning values
w	desired confidence interval width
v	vector of between-subjects contrast coefficients

**Value**

Returns the required sample size per group

**Examples**

```
p <- c(.25, .30, .50, .50)
v <- c(.5, .5, -.5, -.5)
size.ci.lc.prop.bs(.05, p, .2, v)

# Should return:
#   Sample size per group
# [1,]                87
```

---

size.ci.lc.stdmean.bs *Sample size for a between-subjects standardized mean linear contrast confidence interval*

---

**Description**

Computes the sample size in each group (assuming equal sample sizes) required to estimate a standardized linear contrast of population means with desired confidence interval precision in a between-subjects design. Set the standardized mean contrast planning value to the largest value within a plausible range for a conservatively large sample size.

**Usage**

```
size.ci.lc.stdmean.bs(alpha, d, w, v)
```

**Arguments**

alpha	alpha level for 1-alpha confidence
d	planning value of standardized linear contrast
w	desired confidence interval width
v	vector of between-subjects contrast coefficients

**Value**

Returns the required sample size for each group

**References**

Bonett DG (2009). "Estimating standardized linear contrasts of means with desired precision." *Psychological Methods*, **14**(1), 1–5. ISSN 1939-1463, doi: [10.1037/a0014270](https://doi.org/10.1037/a0014270), <https://doi.org/10.1037/a0014270>.

**Examples**

```
v <- c(.5, .5, -.5, -.5)
size.ci.lc.stdmean.bs(.05, 1, .6, v)

# Should return:
#   Sample size per group
# [1,]                   49
```

---

size.ci.lc.stdmean.ws *Sample size for a within-subjects standardized mean linear contrast confidence interval*

---

**Description**

Computes the sample size required to estimate a standardized linear contrast of population means with desired confidence interval precision in a within-subjects design. Set the standardized mean difference planning value to the largest value within a plausible range for a conservatively large sample size. Set the correlation planning value to the smallest value within a plausible range for a conservatively large sample size.

**Usage**

```
size.ci.lc.stdmean.ws(alpha, d, cor, w, q)
```

**Arguments**

alpha	alpha level for 1-alpha confidence
d	planning value of standardized linear contrast
cor	planning value of average correlation between measurements
w	desired confidence interval width
q	vector of within-subjects contrast coefficients

**Value**

Returns the required sample size

**References**

Bonett DG (2009). "Estimating standardized linear contrasts of means with desired precision." *Psychological Methods*, **14**(1), 1–5. ISSN 1939-1463, doi: [10.1037/a0014270](https://doi.org/10.1037/a0014270), <https://doi.org/10.1037/a0014270>.

**Examples**

```
q <- c(.5, .5, -.5, -.5)
size.ci.lc.stdmean.ws(.05, 1, .7, .6, q)

# Should return:
#      Sample size
# [1,]          26
```

---

size.ci.mean.ps	<i>Sample size for a paired-samples mean difference confidence interval</i>
-----------------	---

---

**Description**

Computes the sample size required to estimate a difference in population means with desired confidence interval precision in a paired-samples design. This function requires a planning value for the average of the variances for the two measurements. Set the correlation planning value to the smallest value within a plausible range for a conservatively large sample size. Set the variance planning value to the largest value within a plausible range for a conservatively large sample size.

**Usage**

```
size.ci.mean.ps(alpha, var, cor, w)
```

**Arguments**

alpha	alpha level for 1-alpha confidence
var	planning value of average variance of the two measurements
cor	planning value of correlation between measurements
w	desired confidence interval width

**Value**

Returns the required sample size

**Examples**

```
size.ci.mean.ps(.05, 265, .8, 10)

# Should return:
#      Sample size
# [1,]          19
```

---

size.ci.mean1	<i>Sample size for a single mean confidence interval</i>
---------------	--

---

**Description**

Computes the sample size required to estimate a single population mean with desired confidence interval precision. Set the variance planning value to the largest value within a plausible range for a conservatively large sample size.

**Usage**

```
size.ci.mean1(alpha, var, w)
```

**Arguments**

alpha	alpha level for 1-alpha confidence
var	planning value of response variable variance
w	desired confidence interval width

**Value**

Returns the required sample size

**Examples**

```
size.ci.mean1(.05, 264.4, 10)

# Should return:
#   Sample size
# [1,]         43
```

---

size.ci.mean2	<i>Sample size for a 2-group mean difference confidence interval</i>
---------------	--

---

**Description**

Computes the sample size for each group required to estimate a population mean difference with desired confidence interval precision in a 2-group design. Set  $R = 1$  for equal sample sizes. Set the variance planning value to the largest value within a plausible range for a conservatively large sample size.

**Usage**

```
size.ci.mean2(alpha, var, w, R)
```

**Arguments**

alpha	alpha level for 1-alpha confidence
var	planning value of average within-group variance
w	desired confidence interval width
R	n2/n1 ratio

**Value**

Returns the required sample size for each group

**Examples**

```
size.ci.mean2(.05, 37.1, 5, 1)

# Should return:
#      n1  n2
# [1,] 47  47
```

---

size.ci.prop.ps	<i>Sample size for a paired-sample proportion difference confidence interval</i>
-----------------	--

---

**Description**

Computes the sample size required to estimate a proportion difference with desired confidence interval precision in a paired-samples design. Set the proportion planning values to .5 for a conservatively large sample size. Set the phi planning value to the smallest value within a plausible range for a conservatively large sample size.

**Usage**

```
size.ci.prop.ps(alpha, p1, p2, phi, w)
```

**Arguments**

alpha	alpha level for 1-alpha confidence
p1	planning value of proportion for group 1
p2	planning value of proportion for group 2
phi	planning value of phi coefficient
w	desired confidence interval width

**Value**

Returns the required sample size

**Examples**

```
size.ci.prop.ps(.05, .2, .3, .8, .1)

# Should return:
#   Sample size
# [1,]        118
```

---

size.ci.prop1	<i>Sample size for a single proportion confidence interval</i>
---------------	--

---

**Description**

Computes the sample size required to estimate a single proportion with desired confidence interval precision. Set the proportion planning value to .5 for a conservatively large sample size.

**Usage**

```
size.ci.prop1(alpha, p, w)
```

**Arguments**

alpha	alpha level for 1-alpha confidence
p	planning value of proportion
w	desired confidence interval width

**Value**

Returns the required sample size

**Examples**

```
size.ci.prop1(.05, .4, .2)

# Should return:
#   Sample size
# [1,]        93
```



---

size.ci.prop2      *Sample size for a 2-group proportion difference confidence interval*

---

**Description**

Computes the sample size in each group required to estimate a difference of proportions with desired confidence interval precision in a 2-group design. Set the proportion planning values to .5 for a conservatively large sample size.

**Usage**

```
size.ci.prop2(alpha, p1, p2, w)
```

**Arguments**

alpha	alpha level for 1-alpha confidence
p1	planning value of proportion for group 1
p2	planning value of proportion for group 2
w	desired confidence interval width

**Value**

Returns the required sample size per group

**Examples**

```
size.ci.prop2(.05, .4, .2, .15)

# Should return:
#   Sample size per group
# [1,]                274
```

---

size.ci.ratio.mean.ps      *Sample size for a paired-samples mean ratio confidence interval*

---

**Description**

Computes the sample size required to estimate a ratio of population means with desired confidence interval precision in a paired-samples design. Set the correlation planning value to the smallest value within a plausible range for a conservatively large sample size. This function requires planning values for each mean and the sample size requirement is very sensitive to these planning values. Set the variance planning value to the largest value within a plausible range for a conservatively large sample size.

**Usage**

```
size.ci.ratio.mean.ps(alpha, var, m1, m2, cor, r)
```

**Arguments**

alpha	alpha level for 1-alpha confidence
var	planning value of average variance of the two measurements
m1	planning value of mean for measurement 1
m2	planning value of mean for measurement 2
cor	planning value for correlation between measurements
r	desired upper to lower confidence interval endpoint ratio

**Value**

Returns the required sample size

**Examples**

```
size.ci.ratio.mean.ps(.05, 400, 150, 100, .7, 1.2)

# Should return:
#   Sample size
# [1,]        21
```

---

size.ci.ratio.mean2    *Sample size for a 2-group mean ratio confidence interval*

---

**Description**

Computes the sample size in each group required to estimate a ratio of population means with desired confidence interval precision in a 2-group design. This function requires planning values for each mean and the sample size requirement is very sensitive to these planning values. Set the variance planning value to the largest value within a plausible range for a conservatively large sample size.

**Usage**

```
size.ci.ratio.mean2(alpha, var, m1, m2, r, R)
```

**Arguments**

alpha	alpha level for 1-alpha confidence
var	planning value of average within-group variance
m1	planning value of mean for group 1
m2	planning value of mean for group 2
r	desired upper to lower confidence interval endpoint ratio
R	n2/n1 ratio

**Value**

Returns the required sample size for each group

**Examples**

```
size.ci.ratio.mean2(.05, .4, 3.5, 3.1, 1.2, 2)

# Should return:
#      n1  n2
# [1,] 53 106
```

---

size.ci.ratio.prop.ps *Sample size for a paired-samples proportion ratio confidence interval*

---

**Description**

Computes the sample size required to estimate a ratio of proportions with desired confidence interval precision in a paired-samples design. Set the phi planning value to the smallest value within a plausible range for a conservatively large sample size.

**Usage**

```
size.ci.ratio.prop.ps(alpha, p1, p2, phi, r)
```

**Arguments**

alpha	alpha level for 1-alpha confidence
p1	planning value of proportion for measurement 1
p2	planning value of proportion for measurement 2
phi	planning value of phi coefficient
r	desired upper to lower confidence interval endpoint ratio

**Value**

Returns the required sample size

**Examples**

```
size.ci.ratio.prop.ps(.05, .4, .2, .7, 2)

# Should return:
#   Sample size
# [1,]         67
```

---

size.ci.ratio.prop2    *Sample size for a 2-group proportion ratio confidence interval*

---

**Description**

Computes the sample size in each group required to estimate a ratio of proportions with desired confidence interval precision in a 2-group design.

**Usage**

```
size.ci.ratio.prop2(alpha, p1, p2, r)
```

**Arguments**

alpha	alpha level for 1-alpha confidence
p1	planning value of proportion for group 1
p2	planning value of proportion for group 2
r	desired upper to lower confidence interval endpoint ratio

**Value**

Returns the required sample size per group

**Examples**

```
size.ci.ratio.prop2(.05, .2, .1, 2)

# Should return:
#   Sample size per group
# [1,]                 416
```

---

<code>size.ci.rsqr</code>	<i>Sample size for a squared multiple correlation confidence interval</i>
---------------------------	---

---

**Description**

Computes the sample size required to estimate a squared multiple correlation in a random-x regression model with desired confidence interval precision. Set the planning value of squared multiple correlation to 1/3 for a conservatively large sample size. This function uses an approximation to the standard error of the squared multiple correlation.

**Usage**

```
size.ci.rsqr(alpha, r2, s, w)
```

**Arguments**

<code>alpha</code>	alpha level for 1-alpha confidence
<code>r2</code>	planning value of squared multiple correlation
<code>s</code>	number of predictor variables in model
<code>w</code>	desired confidence interval width

**Value**

Returns the required sample size

**Examples**

```
size.ci.rsqr(.05, .333, 2, .2)

# Should return:
#   Sample size
# [1,]        232
```

---

<code>size.ci.second</code>	<i>Sample size for a second-stage confidence interval</i>
-----------------------------	---

---

**Description**

Computes the second-stage sample size required to obtain desired confidence interval precision. This function can use either the total sample size for all groups in the first stage sample or a single group sample size in the first stage sample. If the total first-stage sample size is given, then the function computes the total sample size required in the second-stage sample. If a single group first-stage sample size is given, then the function computes the single-group sample size required in the second-stage sample. The second-stage sample is combined with the first-stage sample to obtain the desired confidence interval width.

**Usage**

```
size.ci.second(n0, w0, w)
```

**Arguments**

n0	first-stage sample size
w0	confidence interval width in first-stage sample
w	desired confidence interval width

**Value**

Returns the required sample size for the second-stage sample

**Examples**

```
size.ci.second(20, 5.3, 2.5)

# Should return:
#   Second-stage sample size
# [1,]                    70
```

---

size.ci.slope	<i>Sample size for a slope confidence interval</i>
---------------	--

---

**Description**

Computes the total sample size required to estimate a slope with desired confidence interval precision in a between-subjects design with a quantitative factor. In an experimental design, the total sample size would be allocated to the levels of the quantitative factor and it might be necessary to increase the total sample size to achieve equal sample sizes. Set the error variance planning value to the largest value within a plausible range for a conservatively large sample size.

**Usage**

```
size.ci.slope(alpha, evar, x, w)
```

**Arguments**

alpha	alpha level for 1-alpha confidence
evar	planning value of within group (error) variance
x	vector of x values of the quantitative factor
w	desired confidence interval width

**Value**

Returns the required total sample size

**Examples**

```
x <- c(2, 5, 8)
size.ci.slope(.05, 31.1, x, 1)

# Should return:
#   Sample size
# [1,]         83
```

---

size.ci.stdmean.ps	<i>Sample size for a paired-samples standardized mean difference confidence interval</i>
--------------------	--

---

**Description**

Computes the sample size required to estimate a population standardized mean difference with desired confidence interval precision in a paired-samples design. Set the standardized mean difference planning value to the largest value within a plausible range for a conservatively large sample size. Set the correlation planning value to the smallest value within a plausible range for a conservatively large sample size.

**Usage**

```
size.ci.stdmean.ps(alpha, d, cor, w)
```

**Arguments**

alpha	alpha level for 1-alpha confidence
d	planning value of standardized mean difference
cor	planning value of correlation between measurements
w	desired confidence interval width

**Value**

Returns the required sample size

**References**

Bonett DG (2009). "Estimating standardized linear contrasts of means with desired precision." *Psychological Methods*, **14**(1), 1–5. ISSN 1939-1463, doi: [10.1037/a0014270](https://doi.org/10.1037/a0014270), [https://doi.apa.org/getdoi.cfm?doi=10.1037/a0014270.0](https://doi.org/10.1037/a0014270)

**Examples**

```
size.ci.stdmean.ps(.05, 1, .65, .6)

# Should return:
#   Sample size
# [1,]         46
```

---

size.ci.stdmean2	<i>Sample size for a 2-group standardized mean difference confidence interval</i>
------------------	---

---

**Description**

Computes the sample size in each group required to estimate a population standardized mean difference with desired confidence interval precision in a 2-group design. Set the standardized mean difference planning value to the largest value within a plausible range for a conservatively large sample size. Set  $R = 1$  for equal sample sizes.

**Usage**

```
size.ci.stdmean2(alpha, d, w, R)
```

**Arguments**

alpha	alpha level for 1-alpha confidence
d	planning value of standardized mean difference
w	desired confidence interval width
R	$n_2/n_1$ ratio

**Value**

Returns the required sample size for each group

**References**

Bonett DG (2009). "Estimating standardized linear contrasts of means with desired precision." *Psychological Methods*, **14**(1), 1–5. ISSN 1939-1463, doi: [10.1037/a0014270](https://doi.org/10.1037/a0014270), <https://doi.org/10.1037/a0014270>.



## Examples

```
size.ci.stdmean2(.05, .75, .5, 1)

# Should return:
#      n1  n2
# [1,] 132 132
```

---

size.equiv.mean.ps      *Sample size for a paired-samples mean equivalence test*

---

## Description

Computes the sample size required to perform an equivalence test for the difference in population means with desired power in a paired-samples design. The value of  $h$  specifies a range of practical equivalence,  $-h$  to  $h$ , for the difference in population means. The planning value for the absolute mean difference must be less than  $h$ . Equivalence tests often require a very large sample size. Equivalence tests usually use  $2 \times \alpha$  rather than  $\alpha$  (e.g., use  $\alpha = .10$  rather than  $\alpha = .05$ ). Set the correlation value to the smallest value within a plausible range for a conservatively large sample size. Set the variance planning value to the largest value within a plausible range for a conservatively large sample size.

## Usage

```
size.equiv.mean.ps(alpha, pow, var, es, cor, h)
```

## Arguments

alpha	alpha level for hypothesis test
pow	desired power
var	planning value of average within-group variance
es	planning value of mean difference
cor	planning value of the correlation between measurements
h	upper limit for range of practical equivalence

## Value

Returns the required sample size

**Examples**

```
size.equiv.mean.ps(.10, .85, 15, .5, .7, 1.5)

# Should return:
#   Sample size
# [1,]         68
```

---

```
size.equiv.mean2      Sample size for a 2-group mean equivalence test
```

---

**Description**

Computes the sample size in each group (assuming equal sample sizes) required to perform an equivalence test for the difference in population means with desired power in a 2-group design. The value of  $h$  specifies a range of practical equivalence,  $-h$  to  $h$ , for the difference in population means. The planning value for the absolute mean difference must be less than  $h$ . Equivalence tests often require a very large sample size. Equivalence tests usually use  $2 \times \alpha$  rather than  $\alpha$  (e.g., use  $\alpha = .10$  rather than  $\alpha = .05$ ). Set the variance planning value to the largest value within a plausible range for a conservatively large sample size.

**Usage**

```
size.equiv.mean2(alpha, pow, var, es, h)
```

**Arguments**

alpha	alpha level for hypothesis test
pow	desired power
var	planning value of average within-group variance
es	planning value of mean difference
h	upper limit for range of practical equivalence

**Value**

Returns the required sample size per group

**Examples**

```
size.equiv.mean2(.10, .80, 15, 2, 4)

# Should return:
#   Sample size per group
# [1,]                 50
```

---

size.equiv.prop.ps      *Sample size for a paired-samples proportion equivalence test*

---

### Description

Computes the sample size in each group required to perform an equivalence test for the difference in population proportions with desired power in a paired-samples design. The value of *h* specifies a range of practical equivalence,  $-h$  to  $h$ , for the difference in population proportions. The absolute difference in the proportion planning values must be less than *h*. Equivalence tests often require a very large sample size. Equivalence tests usually use  $2 \times \alpha$  rather than  $\alpha$  (e.g., use  $\alpha = .10$  rather than  $\alpha = .05$ ). This function sets the effect size equal to the difference in proportion planning values. Set the *phi* planning value to the smallest value within a plausible range for a conservatively large sample size.

### Usage

```
size.equiv.prop.ps(alpha, pow, p1, p2, phi, h)
```

### Arguments

<i>alpha</i>	alpha level for hypothesis test
<i>pow</i>	desired power
<i>p1</i>	planning value of proportion for group 1
<i>p2</i>	planning value of proportion for group 2
<i>phi</i>	planning value of phi coefficient
<i>h</i>	upper limit for range of practical equivalence

### Value

Returns the required sample size

### Examples

```
size.equiv.prop.ps(.1, .8, .30, .35, .40, .15)

# Should return:
#   Sample size
# [1,]      173
```

---

size.equiv.prop2	<i>Sample size for a 2-group proportion equivalence test</i>
------------------	--

---

### Description

Computes the sample size in each group required to perform an equivalence test for the difference in population proportions with desired power in a 2-group design. The value of  $h$  specifies a range of practical equivalence,  $-h$  to  $h$ , for the difference in population proportions. The absolute difference in the proportion planning values must be less than  $h$ . Equivalence tests often require a very large sample size. Equivalence tests usually use  $2 \times \alpha$  rather than  $\alpha$  (e.g., use  $\alpha = .10$  rather than  $\alpha = .05$ ).

### Usage

```
size.equiv.prop2(alpha, pow, p1, p2, h)
```

### Arguments

alpha	alpha level for hypothesis test
pow	desired power
p1	planning value of proportion for group 1
p2	planning value of proportion for group 2
h	upper limit for range of practical equivalence

### Value

Returns the required sample size per group

### Examples

```
size.equiv.prop2(.1, .8, .30, .35, .15)

# Should return:
#   Sample size per group
# [1,]                288
```

---

size.interval.cor      *Sample size for a finite interval test of a Pearson or partial correlation*

---

### Description

Computes the sample size required to perform a finite interval test for a Pearson or a partial correlation with desired power. Set  $s = 0$  for a Pearson correlation. The correlation planning value must be a value within the hypothesized finite interval.

### Usage

```
size.interval.cor(alpha, pow, cor, s, h)
```

### Arguments

alpha	alpha level for hypothesis test
pow	desired power
cor	planning value of correlation
s	number of control variables
h	upper limit of hypothesized interval

### Value

Returns the required sample size

### Examples

```
size.interval.cor(.05, .8, .1, 0, .25)

# Should return:
#   Sample size
# [1,]       360
```

---

size.supinf.mean.ps      *Sample size for a paired-samples mean superiority or noninferiority test*

---

### Description

Computes the sample size required to perform a superiority or noninferiority test for the difference in population means with desired power in a paired-samples design. For a superiority test, specify the upper limit ( $h$ ) for the range of practical equivalence and specify an effect size ( $es$ ) such that  $es > h$ . For a noninferiority test, specify the lower limit ( $-h$ ) for the range of practical equivalence and specify an effect size such that  $es > -h$ . Set the correlation planning value to the smallest value within a plausible range for a conservatively large sample size. Set the variance planning value to the largest value within a plausible range for a conservatively large sample size.

**Usage**

```
size.supinf.mean.ps(alpha, pow, var, es, cor, h)
```

**Arguments**

alpha	alpha level for hypothesis test
pow	desired power
var	planning value of average within-group variance
es	planning value of mean difference
cor	planning value of the correlation between measurements
h	upper or lower limit for range of practical equivalence

**Value**

Returns the required sample size

**Examples**

```
size.supinf.mean.ps(.05, .80, 225, 9, .75, 4)

# Should return:
#   Sample size
# [1,]        38
```

---

size.supinf.mean2      *Sample size for a 2-group mean superiority or noninferiority test*

---

**Description**

Computes the sample size in each group (assuming equal sample sizes) required to perform a superiority or noninferiority test for the difference in population means with desired power in a 2-group design. For a superiority test, specify the upper limit (h) for the range of practical equivalence and specify an effect size (es) such that  $es > h$ . For a noninferiority test, specify the lower limit (-h) for the range of practical equivalence and specify an effect size such that  $es > -h$ . Set the variance planning value to the largest value within a plausible range for a conservatively large sample size.

**Usage**

```
size.supinf.mean2(alpha, pow, var, es, h)
```

**Arguments**

alpha	alpha level for hypothesis test
pow	desired power
var	planning value of average within-group variance
es	planning value of mean difference
h	upper or lower limit for range of practical equivalence

**Value**

Returns the required sample size per group

**Examples**

```
size.supinf.mean2(.05, .80, 225, 9, 4)

# Should return:
#   Sample size per group
# [1,]                143
```

---

size.supinf.prop.ps     *Sample size for a paired-samples superiority or inferiority test of proportions*

---

**Description**

Computes the sample size required to perform a superiority or inferiority test for the difference in population proportions with desired power in a paired-samples design. For a superiority test, specify the upper limit (h) for the range of practical equivalence and specify values of p1 and p2 such that  $p1 - p2 > h$ . For an inferiority test, specify the lower limit (-h) for the range of practical equivalence and specify values of p1 and p2 such that  $p1 - p2 > -h$ . This function sets the effect size equal to the difference in proportion planning values. Set the phi planning value to the smallest value within a plausible range for a conservatively large sample size.

**Usage**

```
size.supinf.prop.ps(alpha, pow, p1, p2, phi, h)
```

**Arguments**

alpha	alpha level for hypothesis test
pow	desired power
p1	planning value of proportion for measurement 1
p2	planning value of proportion for measurement 2
phi	planning value of phi coefficient
h	lower or upper limit for range of practical equivalence

**Value**

Returns the required sample size

**Examples**

```
size.supinf.prop.ps(.05, .9, .35, .20, .45, .05)

# Should return:
#   Sample size
# [1,]      227
```

---

size.supinf.prop2      *Sample size for a 2-group superiority or inferiority test of proportions*

---

**Description**

Computes the sample size in each group required to perform a superiority or inferiority test for the difference in population proportions with desired power in a 2-group design. For a superiority test, specify the upper limit (h) for the range of practical equivalence and specify values of p1 and p2 such that  $p1 - p2 > h$ . For an inferiority test, specify the lower limit (-h) for the range of practical equivalence and specify values of p1 and p2 such that  $p1 - p2 > -h$ . This function sets the effect size equal to  $p1 - p2$ .

**Usage**

```
size.supinf.prop2(alpha, pow, p1, p2, h)
```

**Arguments**

alpha	alpha level for hypothesis test
pow	desired power
p1	planning value of proportion for group 1
p2	planning value of proportion for group 2
h	lower or upper limit for range of practical equivalence

**Value**

Returns the required sample size per group



**Examples**

```
size.supinf.prop2(.05, .9, .35, .20, .05)

# Should return:
#   Sample size per group
# [1,]                408
```

---

size.test.cor

*Sample size for a test of a Pearson or partial correlation*

---

**Description**

Computes the sample size required to test a Pearson or a partial correlation with desired power. Set  $s = 0$  for a Pearson correlation. Set the correlation planning value to the smallest value within a plausible range for a conservatively large sample size.

**Usage**

```
size.test.cor(alpha, pow, cor, s, h)
```

**Arguments**

alpha	alpha level for hypothesis test
pow	desired power
cor	planning value of correlation
s	number of control variables
h	hypothesized value of slope

**Value**

Returns the required sample size

**Examples**

```
size.test.cor(.05, .9, .45, 0, 0)

# Should return:
#   Sample size
# [1,]         48
```

---

size.test.cronbach     *Sample size to test a Cronbach reliability*

---

### Description

Computes the sample size required to test a Cronbach reliability with desired power.

### Usage

```
size.test.cronbach(alpha, pow, rel, a, h)
```

### Arguments

alpha	alpha level for hypothesis test
pow	desired power
rel	reliability planning value
a	number of measurements
h	null hypothesis value of reliability

### Value

Returns the required sample size

### References

Bonett DG, Wright TA (2015). "Cronbach's alpha reliability: Interval estimation, hypothesis testing, and sample size planning." *Journal of Organizational Behavior*, **36**(1), 3–15. ISSN 08943796, doi: [10.1002/job.1960](https://doi.org/10.1002/job.1960), <https://onlinelibrary.wiley.com/doi/10.1002/job.1960>.

### Examples

```
size.test.cronbach(.05, .85, .80, 5, .7)

# Should return:
#   Sample size
# [1,]         139
```

---

size.test.lc.ancova    *Sample size for a mean linear contrast test in an ANCOVA*

---

### Description

Computes the sample size for each group required to test a linear contrast of means in an ANCOVA model with desired power. In a nonexperimental design, the sample size is affected by the magnitude of covariate mean differences across groups. The covariate mean differences can be approximated by specifying the largest standardized covariate mean difference across all pairwise comparisons and for all covariates. In an experiment, this standardized mean difference is set to 0. Set the error variance planning value to the largest value within a plausible range for a conservatively large sample size.

### Usage

```
size.test.lc.ancova(alpha, pow, evar, es, s, d, v)
```

### Arguments

alpha	alpha level for hypothesis test
pow	desired power
evar	planning value of within group (error) variance
es	planning value of linear contrast
s	number of covariates
d	largest standardized mean difference for all covariates
v	vector of between-subjects contrast coefficients

### Value

Returns the required sample size per group

### Examples

```
v <- c(.5, .5, -1)
size.test.lc.ancova(.05, .9, 1.37, .7, 1, 0, v)

# Should return:
#     Sample size per group
# [1,]                    47
```

---

size.test.lc.mean.bs *Sample size for a test of a between-subjects mean linear contrast*

---

### Description

Computes the sample size in each group (assuming equal sample sizes) required to test a linear contrast of population means with desired power in a between-subjects design. Set the variance planning value to the largest value within a plausible range for a conservatively large sample size.

### Usage

```
size.test.lc.mean.bs(alpha, pow, var, es, v)
```

### Arguments

alpha	alpha level for hypothesis test
pow	desired power
var	planning value of average within-group variance
es	planning value of linear contrast of means
v	vector of between-subjects contrast coefficients

### Value

Returns the required sample size per group

### Examples

```
v <- c(1, -1, -1, 1)
size.test.lc.mean.bs(.05, .90, 27.5, 5, v)

# Should return:
#   Sample size per group
# [1,]                  47
```

---

size.test.lc.mean.ws *Sample size for a test of a within-subjects mean linear contrast*

---

### Description

Computes the sample size required to test a linear contrast of population means with desired power in a within-subjects design. Set the variance planning value to the largest value within a plausible range for a conservatively large sample size. Set the correlation planning value to the smallest value within a plausible range for a conservatively large sample size.

**Usage**

```
size.test.lc.mean.ws(alpha, pow, var, es, cor, q)
```

**Arguments**

alpha	alpha level for hypothesis test
pow	desired power
var	planning value of average variance of measurements
es	planning value of linear contrast of means
cor	planning value of average correlation between measurements
q	vector of with-subjects contrast coefficients

**Value**

Returns the required sample size

**Examples**

```
q <- c(.5, .5, -.5, -.5)
size.test.lc.mean.ws(.05, .90, 50.7, 2, .8, q)

# Should return:
#   Sample size
# [1,]        29
```

---

size.test.lc.prop.bs *Sample size for a test of between-subjects proportion linear contrast*

---

**Description**

Computes the sample size in each group required to test a linear contrast of population proportions with desired power in a between-subjects design. This function requires planning values for all proportions. Set the proportion planning values to .5 for a conservatively large sample size. The planning value for the linear contrast of proportions could be set equal to the linear contrast of proportion planning values or it could be set equal to a minimally interesting effect size.

**Usage**

```
size.test.lc.prop.bs(alpha, pow, p, es, v)
```

**Arguments**

alpha	alpha level for hypothesis test
pow	desired power
p	vector of proportion planning values
es	planning value of proportion linear contrast
v	vector of between-subjects contrast coefficients

**Value**

Returns the required sample size per group

**Examples**

```
p <- c(.25, .30, .50, .50)
v <- c(.5, .5, -.5, -.5)
size.test.lc.prop.bs(.05, .9, p, .15, v)

# Should return:
#   Sample size per group
# [1,]                105
```

---

size.test.mann	<i>Sample size for a Mann-Whitney test</i>
----------------	--

---

**Description**

Computes the sample size in each group (assuming equal sample sizes) required for the Mann-Whitney test with desired power. A planning value of the Mann-Whitney parameter is required. In a 2-group experiment, this parameter is the proportion of members in the population with scores that would be larger under treatment 1 than treatment 2. In a 2-group nonexperiment where participants are sampled from two subpopulations of sizes N1 and N2, the parameter is the proportion of all N1 x N2 pairs in which a member from subpopulation 1 has a larger score than a member from subpopulation 2.

**Usage**

```
size.test.mann(alpha, pow, p)
```

**Arguments**

alpha	alpha level for hypothesis test
pow	desired power
p	planning value of Mann-Whitney parameter

**Value**

Returns the required sample size per group

**References**

Noether GE (1987). "Sample size determination for some common nonparametric tests." *Journal of the American Statistical Association*, **82**(398), 645–647. ISSN 0162-1459, doi: [10.1080/01621459.1987.10478478](https://doi.org/10.1080/01621459.1987.10478478), <https://www.tandfonline.com/doi/abs/10.1080/01621459.1987.10478478>.

**Examples**

```
size.test.mann(.05, .90, .3)

# Should return:
#   Sample size per group
# [1,]                44
```

---

size.test.mean.ps      *Sample size for a test of a paired-samples mean difference*

---

**Description**

Computes the sample size required to test a difference in population means with desired power in a paired-samples design. Set the correlation planning value to the smallest value within a plausible range for a conservatively large sample size. Set the variance planning value to the largest value within a plausible range for a conservatively large sample size.

**Usage**

```
size.test.mean.ps(alpha, pow, var, es, cor)
```

**Arguments**

alpha	alpha level for hypothesis test
pow	desired power
var	planning value of average variance of the two measurements
es	planning value of mean difference
cor	planning value of correlation

**Value**

Returns the required sample size

**Examples**

```
size.test.mean.ps(.05, .80, 1.25, .5, .75)

# Should return:
#   Sample size
# [1,]         22
```

---

size.test.mean1	<i>Sample size for a test of a single mean</i>
-----------------	--

---

**Description**

Computes the sample size required to test a single population mean with desired power in a 1-group design. Set the variance planning value to the largest value within a plausible range for a conservatively large sample size.

**Usage**

```
size.test.mean1(alpha, pow, var, es)
```

**Arguments**

alpha	alpha level for hypothesis test
pow	desired power
var	planning value of response variable variance
es	planning value of mean minus null hypothesis value

**Value**

Returns the required sample size

**Examples**

```
size.test.mean1(.05, .9, 80.5, 7)

# Should return:
#   Sample size
# [1,]         20
```



---

size.test.mean2      *Sample size for a test of a 2-group mean difference*

---

### Description

Computes the sample size in each group requires to test a difference in population means with desired power in a 2-group design. Set R =1 for equal sample sizes. Set the variance planning value to the largest value within a plausible range for a conservatively large sample size.

### Usage

```
size.test.mean2(alpha, pow, var, es, R)
```

### Arguments

alpha	alpha level for hypothesis test
pow	desired power
var	planning value of average within-group variance
es	planning value of mean difference
R	n2/n1 ratio

### Value

Returns the required sample size per group

### Examples

```
size.test.mean2(.05, .95, 100, 10, 1)

# Should return:
#   n1 n2
# [1,] 27 27
```

---

size.test.prop.ps      *Sample size for a test of a paired-samples proportion difference*

---

### Description

Computes the sample size required to test a difference in population proportions with desired power in a paired-samples design. This function requires planning values for both proportions and a phi coefficient that describes the correlation between the two dichotomous measurements. The proportion planning values can set to .5 for a conservatively large sample size. The planning value for the proportion difference could be set equal to the difference of the two proportion planning values or it could be set equal to a minimally interesting effect size. Set the phi planning value to the smallest value within a plausible range for a conservatively large sample size.

**Usage**

```
size.test.prop.ps(alpha, pow, p1, p2, phi, es)
```

**Arguments**

alpha	alpha level for hypothesis test
pow	desired power
p1	planning value of proportion for measurement 1
p2	planning value of proportion for measurement 2
phi	planning value of phi coefficient
es	planning value of proportion difference

**Value**

Returns the required sample size

**Examples**

```
size.test.prop.ps(.05, .80, .4, .3, .5, .1)

# Should return:
#   Sample size
# [1,]       177
```

---

size.test.prop1	<i>Sample size for a test of a single proportion</i>
-----------------	--

---

**Description**

Computes the sample size required to test a single population proportion with desired power in a 1-group design. Set the proportion planning value to .5 for a conservatively large sample size.

**Usage**

```
size.test.prop1(alpha, pow, p, es)
```

**Arguments**

alpha	alpha level for hypothesis test
pow	desired power
p	planning value of proportion
es	planning value of proportion minus null hypothesis value

**Value**

Returns the required sample size

**Examples**

```
size.test.prop1(.05, .9, .5, .2)

# Should return:
#   Sample size
# [1,]         66
```

---

size.test.prop2      *Sample size for a test of a 2-group proportion difference*

---

**Description**

Computes the sample size in each group required to test a difference in population proportions with desired power in a 2-group design. This function requires planning values for both proportions. Set the proportion planning values to .5 for a conservatively large sample size. The planning value for the proportion difference could be set equal to the difference of the two proportion planning values or it could be set equal to a minimally interesting effect size.

**Usage**

```
size.test.prop2(alpha, pow, p1, p2, es)
```

**Arguments**

alpha	alpha level for hypothesis test
pow	desired power
p1	planning value of proportion for group 1
p2	planning value of proportion for group 2
es	planning value of proportion difference

**Value**

Returns the required sample size per group

**Examples**

```
size.test.prop2(.05, .8, .2, .4, .2)

# Should return:
#   Sample size per group
# [1,]                 79
```

---

size.test.sign.ps      *Sample size for a paired-samples Sign test*

---

### Description

Computes sample size required for a Sign test with desired power in a paired-samples design. A planning value of the paired-samples Sign test parameter is required. In a paired-samples experiment, this parameter is the proportion of members in the population with scores that would be larger under treatment 1 than treatment 2. In a paired-samples nonexperiment, this parameter is the proportion of members in the population with measurement 1 scores that are larger than their measurement 2 scores.

### Usage

```
size.test.sign.ps(alpha, pow, p)
```

### Arguments

alpha	alpha level for hypothesis test
pow	desired power
p	planning value of proportion

### Value

Returns the required sample size

### Examples

```
size.test.sign.ps(.05, .90, .75)

# Should return:
#   Sample size
# [1,]        32
```

---

size.test.sign1      *Sample size for a 1-sample Sign test*

---

### Description

Computes the sample size required for a Sign test with desired power in a 1-sample design. A planning value of the 1-sample Sign test parameter value is required. This parameter is the proportion of members in the population with scores greater than the hypothesized value.

**Usage**

```
size.test.sign1(alpha, pow, p)
```

**Arguments**

alpha	alpha level for hypothesis test
pow	desired power
p	planning value of proportion

**Value**

Returns the required sample size

**Examples**

```
size.test.sign1(.05, .90, .3)

# Should return:
#   Sample size
# [1,]        56
```

---

size.test.slope	<i>Sample size for a test of a slope</i>
-----------------	--

---

**Description**

Computes the total sample size required to test a population slope with desired power in a between-subjects design with a quantitative factor. In an experimental design, the total sample size would be allocated to the levels of the quantitative factor and it might be necessary to use a larger total sample size to achieve equal sample sizes. Set the error variance planning value to the largest value within a plausible range for a conservatively large sample size.

**Usage**

```
size.test.slope(alpha, pow, evar, x, slope, h)
```

**Arguments**

alpha	alpha level for hypothesis test
pow	desired power
evar	planning value of within group (error) variance
x	vector of x values of the quantitative factor
slope	planning value of slope
h	hypothesized value of slope

**Value**

Returns the required total sample size

**Examples**

```
x <- c(2, 5, 8)
size.test.slope(.05, .9, 31.1, x, .75, 0)

# Should return:
#   Sample size
# [1,]       100
```

---

slope.contrast	<i>Contrast coefficients for the slope of a quantitative factor</i>
----------------	---

---

**Description**

Computes the contrast coefficients to estimate the slope of a line in a single factor design with a quantitative factor.

**Usage**

```
slope.contrast(x)
```

**Arguments**

x                    vector of numeric factor levels

**Value**

Returns the vector of contrast coefficients

**Examples**

```
x <- c(25, 50, 75, 100)
slope.contrast(x)

# Should return:
#   Coefficient
# [1,]   -0.012
# [2,]   -0.004
# [3,]    0.004
# [4,]    0.012
```

---

test.anova1.bs	<i>Between-subjects F statistic and eta-squared from summary information</i>
----------------	--

---

### Description

Computes the F statistic, p-value, eta-squared, and adjusted eta-squared for the main effect of Factor A in a one-way between-subjects ANOVA using the sample means, sample standard deviations, and sample sizes.

### Usage

```
test.anova1.bs(m, sd, n)
```

### Arguments

m	vector of sample means
sd	vector of sample standard deviations
n	vector of sample sizes

### Value

Returns a 1-row matrix. The columns are:

- F - F statistic for test of null hypothesis
- dfA - degrees of freedom for between-subjects factor
- dfE - error degrees of freedom
- dfA - degrees of freedom for between-subjects factor
- p - p-value for F-test
- eta-squared - estimate of eta-squared
- adj eta-squared - a bias adjusted estimate of eta-squared

### Examples

```
m <- c(12.4, 8.6, 10.5)
sd <- c(3.84, 3.12, 3.48)
n <- c(20, 20, 20)
test.anova1.bs(m, sd, n)

# Should return:
#           F dfA dfE           p eta-squared adj eta-squared
# [1,] 5.919585 2 57 0.004614428 0.1719831 0.1429298
```

---

`test.prop.bs`*Hypothesis test of equal proportions in a between-subjects design*

---

**Description**

Computes a Pearson chi-square test for equal population proportions for a dichotomous response variable in a one-factor between-subjects design.

**Usage**

```
test.prop.bs(f, n)
```

**Arguments**

f	vector of frequency counts of participants with the attribute
n	vector of sample sizes

**Value**

Returns a 1-row matrix. The columns are: Chi-square - chi-square test statistic df - degrees of freedom p - p-value

**References**

Fleiss JL, Paik MC (2003). *Statistical Methods for Rates and Proportions*, 3rd edition. Wiley.

**Examples**

```
f <- c(35, 30, 15)
n <- c(50, 50, 50)
test.prop.bs(f, n)

# Should return:
#   Chi-square df      p
# [1,] 17.41071 2 0.0001656958
```



---

`test.prop.ps`*Hypothesis test for a paired-samples proportion difference*

---

### Description

Computes a continuity-corrected McNemar test for equality of proportions in a paired-samples design. This function requires the frequency counts from a 2 x 2 contingency table for two paired dichotomous measurements.

### Usage

```
test.prop.ps(f00, f01, f10, f11)
```

### Arguments

<code>f00</code>	number participants with $y = 0$ and $x = 0$
<code>f01</code>	number participants with $y = 0$ and $x = 1$
<code>f10</code>	number participants with $y = 1$ and $x = 0$
<code>f11</code>	number participants with $y = 1$ and $x = 1$

### Value

Returns a 1-row matrix. The columns are: Estimate - ML estimate of proportion difference  $z$  -  $z$  test statistic  $p$  -  $p$ -value

### References

Snedecor GW, Cochran WG (1989). *Statistical Methods*, 8th edition. ISU University Pres, Ames, Iowa.

### Examples

```
test.prop.ps(156, 96, 68, 80)

# Should return:
#   Estimate      z      p
# [1,]    0.07 2.108346 0.03500109
```

---

`test.prop1`*Hypothesis test for a single proportion*

---

**Description**

Computes a continuity-corrected z test for a single proportion in a 1-group design.

**Usage**

```
test.prop1(f, n, h)
```

**Arguments**

f	number of participants who have the attribute
n	sample size
h	population proportion under null hypothesis

**Value**

Returns a 1-row matrix. The columns are:

- Estimate - ML estimate of proportion
- z - z test statistic
- p - p-value

**References**

Snedecor GW, Cochran WG (1989). *Statistical Methods*, 8th edition. ISU University Pres, Ames, Iowa.

**Examples**

```
test.prop1(9, 20, .2)

# Should return:
#   Estimate      z      p
# [1,]    0.45 2.515576 0.01188379
```

---

`test.prop2`*Hypothesis test for a 2-group proportion difference*

---

**Description**

Computes a continuity-corrected z test for a difference of proportions in a 2-group design.

**Usage**

```
test.prop2(f1, f2, n1, n2)
```

**Arguments**

f1	number of group 1 participants who have the attribute
f2	number of group 2 participants who have the attribute
n1	sample size in group 1
n2	sample size in group 2

**Value**

Returns a 1-row matrix. The columns are:

- Estimate - ML estimate of proportion difference
- z - z test statistic
- p - p-value

**References**

Snedecor GW, Cochran WG (1989). *Statistical Methods*, 8th edition. ISU University Pres, Ames, Iowa.

**Examples**

```
test.prop2(11, 26, 50, 50)

# Should return:
# Estimate      z      p
#      -0.3 2.899726 0.003734895
```

# Index

ci.agree, 4  
ci.biphi, 5  
ci.cod1, 6  
ci.condslope, 7  
ci.condslope.log, 8  
ci.cor, 9  
ci.cor.dep, 10  
ci.cor2, 11  
ci.cor2.gen, 12  
ci.cronbach, 13  
ci.fisher, 14  
ci.indirect, 15  
ci.kappa, 16  
ci.lc.mean.bs, 17  
ci.lc.median.bs, 18  
ci.lc.prop.bs, 19  
ci.lc.reg, 20  
ci.lc.stdmean.bs, 21  
ci.lc.stdmean.ws, 22  
ci.mad1, 23  
ci.mann, 24  
ci.mape, 25  
ci.mean.ps, 26  
ci.mean1, 27  
ci.mean2, 28  
ci.median.ps, 29  
ci.median1, 30  
ci.median2, 31  
ci.oddsratio, 32  
ci.pairs.prop.bs, 33  
ci.pairs.prop1, 34  
ci.pbcor, 35  
ci.phi, 36  
ci.popsiz, 37  
ci.prop.ps, 38  
ci.prop1, 39  
ci.prop2, 40  
ci.random.anova1, 41  
ci.ratio.mad.ps, 42  
ci.ratio.mad2, 43  
ci.ratio.mean.ps, 44  
ci.ratio.mean2, 45  
ci.ratio.median.ps, 46  
ci.ratio.median2, 47  
ci.ratio.prop.ps, 48  
ci.ratio.prop2, 49  
ci.slope.prop.bs, 50  
ci.spcor, 51  
ci.spear, 52  
ci.spear2, 53  
ci.stdmean.ps, 54  
ci.stdmean.strat, 55  
ci.stdmean1, 56  
ci.stdmean2, 57  
ci.tetra, 58  
ci.tukey, 59  
ci.var.upper, 60  
ci.yule, 61  
  
etasqr.adj, 62  
  
iqv, 63  
  
pi.score1, 63  
pi.score2, 64  
  
random.sample, 65  
random.y, 66  
random.yx, 67  
randomize, 68  
  
size.ci.agree, 68  
size.ci.condmean, 69  
size.ci.cor, 70  
size.ci.cronbach, 71  
size.ci.lc.ancova, 72  
size.ci.lc.mean.bs, 73  
size.ci.lc.mean.ws, 73  
size.ci.lc.prop.bs, 74  
size.ci.lc.stdmean.bs, 75

size.ci.lc.stdmean.ws, 76  
size.ci.mean.ps, 77  
size.ci.mean1, 78  
size.ci.mean2, 78  
size.ci.prop.ps, 79  
size.ci.prop1, 80  
size.ci.prop2, 81  
size.ci.ratio.mean.ps, 81  
size.ci.ratio.mean2, 82  
size.ci.ratio.prop.ps, 83  
size.ci.ratio.prop2, 84  
size.ci.rsqr, 85  
size.ci.second, 85  
size.ci.slope, 86  
size.ci.stdmean.ps, 87  
size.ci.stdmean2, 88  
size.equiv.mean.ps, 89  
size.equiv.mean2, 90  
size.equiv.prop.ps, 91  
size.equiv.prop2, 92  
size.interval.cor, 93  
size.supinf.mean.ps, 93  
size.supinf.mean2, 94  
size.supinf.prop.ps, 95  
size.supinf.prop2, 96  
size.test.cor, 97  
size.test.cronbach, 98  
size.test.lc.ancova, 99  
size.test.lc.mean.bs, 100  
size.test.lc.mean.ws, 100  
size.test.lc.prop.bs, 101  
size.test.mann, 102  
size.test.mean.ps, 103  
size.test.mean1, 104  
size.test.mean2, 105  
size.test.prop.ps, 105  
size.test.prop1, 106  
size.test.prop2, 107  
size.test.sign.ps, 108  
size.test.sign1, 108  
size.test.slope, 109  
slope.contrast, 110  
  
test.anova1.bs, 111  
test.prop.bs, 112  
test.prop.ps, 113  
test.prop1, 114  
test.prop2, 115