

# Package ‘statpsych’

July 9, 2024

**Type** Package

**Title** Statistical Methods for Psychologists

**Version** 1.6.0

**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: Statistical Methods for Psychologists, Volumes 1 – 4, <<https://dgbonett.sites.ucsc.edu/>>.

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

**License** GPL-3

**Encoding** UTF-8

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

**RoxygenNote** 7.3.2

**RdMacros** Rdpack, mathjaxr

**Suggests** MASS, 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** 2024-07-09 02:40:02 UTC

## Contents

adj.se . . . . .	6
ci.2x2.mean.bs . . . . .	7
ci.2x2.mean.mixed . . . . .	8
ci.2x2.mean.ws . . . . .	10
ci.2x2.median.bs . . . . .	11
ci.2x2.median.mixed . . . . .	12
ci.2x2.median.ws . . . . .	13
ci.2x2.prop.bs . . . . .	15
ci.2x2.prop.mixed . . . . .	16
ci.2x2.stdmean.bs . . . . .	17
ci.2x2.stdmean.mixed . . . . .	18
ci.2x2.stdmean.ws . . . . .	20
ci.agree . . . . .	21
ci.agree.3rater . . . . .	22
ci.agree2 . . . . .	23
ci.bayes.normal . . . . .	24
ci.bayes.prop . . . . .	25
ci.biphi . . . . .	26
ci.bscor . . . . .	27
ci.cod . . . . .	29
ci.condslope . . . . .	30
ci.condslope.log . . . . .	31
ci.cor . . . . .	32
ci.cor.dep . . . . .	33
ci.cor2 . . . . .	34
ci.cor2.gen . . . . .	35
ci.cqv . . . . .	36
ci.cramer . . . . .	37
ci.cronbach . . . . .	38
ci.cronbach2 . . . . .	39
ci.cv . . . . .	40
ci.etasqr . . . . .	41
ci.fisher . . . . .	42
ci.indirect . . . . .	43
ci.kappa . . . . .	44
ci.lc.gen.bs . . . . .	45
ci.lc.glm . . . . .	46
ci.lc.mean.bs . . . . .	47
ci.lc.median.bs . . . . .	48
ci.lc.prop.bs . . . . .	49
ci.lc.reg . . . . .	50
ci.lc.stdmean.bs . . . . .	51
ci.lc.stdmean.ws . . . . .	53
ci.mad . . . . .	54
ci.mann . . . . .	55
ci.mape . . . . .	56

ci.mean	57
ci.mean.fpc	58
ci.mean.ps	59
ci.mean2	60
ci.median	61
ci.median.ps	62
ci.median2	63
ci.oddsratio	64
ci.pairs.mult	65
ci.pairs.prop.bs	66
ci.pbcor	67
ci.phi	68
ci.poisson	69
ci.popsiz	70
ci.prop	71
ci.prop.fpc	72
ci.prop.inv	73
ci.prop.ps	74
ci.prop2	75
ci.prop2.inv	76
ci.pv	77
ci.random.anova	78
ci.ratio.cod2	79
ci.ratio.cv2	80
ci.ratio.mad.ps	81
ci.ratio.mad2	82
ci.ratio.mape2	83
ci.ratio.mean.ps	84
ci.ratio.mean2	85
ci.ratio.median.ps	86
ci.ratio.median2	87
ci.ratio.poisson2	88
ci.ratio.prop.ps	89
ci.ratio.prop2	90
ci.ratio.sd2	91
ci.rel2	92
ci.reliability	93
ci.rsqr	94
ci.sign	95
ci.slope.mean.bs	96
ci.slope.prop.bs	97
ci.spcor	98
ci.spear	99
ci.spear2	100
ci.stdmean	101
ci.stdmean.ps	102
ci.stdmean.strat	103
ci.stdmean2	104

ci.tetra . . . . .	106
ci.theil . . . . .	107
ci.tukey . . . . .	108
ci.var.upper . . . . .	109
ci.yule . . . . .	110
etasqr.adj . . . . .	111
etasqr.gen.2way . . . . .	112
fitindices . . . . .	113
iqv . . . . .	113
pi.cor . . . . .	114
pi.prop . . . . .	115
pi.score . . . . .	116
pi.score.ps . . . . .	117
pi.score2 . . . . .	118
pi.var.upper . . . . .	119
power.cor . . . . .	120
power.cor2 . . . . .	121
power.lc.mean.bs . . . . .	122
power.mean . . . . .	123
power.mean.ps . . . . .	123
power.mean2 . . . . .	124
power.prop . . . . .	125
power.prop.ps . . . . .	126
power.prop2 . . . . .	127
random.sample . . . . .	128
random.y . . . . .	128
random.yx . . . . .	129
randomize . . . . .	130
sim.ci.cor . . . . .	131
sim.ci.mean . . . . .	132
sim.ci.mean.ps . . . . .	133
sim.ci.mean2 . . . . .	134
sim.ci.median . . . . .	135
sim.ci.median.ps . . . . .	136
sim.ci.median2 . . . . .	137
sim.ci.spear . . . . .	138
sim.ci.stdmean.ps . . . . .	139
sim.ci.stdmean2 . . . . .	141
size.ci.agree . . . . .	142
size.ci.condmean . . . . .	143
size.ci.cor . . . . .	144
size.ci.cor.prior . . . . .	145
size.ci.cor2 . . . . .	146
size.ci.cronbach . . . . .	147
size.ci.cronbach2 . . . . .	148
size.ci.etasqr . . . . .	149
size.ci.indirect . . . . .	149
size.ci.lc.ancova . . . . .	150

size.ci.lc.mean.bs . . . . .	151
size.ci.lc.mean.ws . . . . .	152
size.ci.lc.prop.bs . . . . .	153
size.ci.lc.stdmean.bs . . . . .	153
size.ci.lc.stdmean.ws . . . . .	154
size.ci.mape . . . . .	155
size.ci.mean . . . . .	156
size.ci.mean.prior . . . . .	157
size.ci.mean.ps . . . . .	158
size.ci.mean2 . . . . .	158
size.ci.pbcor . . . . .	159
size.ci.prop . . . . .	160
size.ci.prop.prior . . . . .	161
size.ci.prop.ps . . . . .	162
size.ci.prop2 . . . . .	163
size.ci.ratio.mean.ps . . . . .	163
size.ci.ratio.mean2 . . . . .	164
size.ci.ratio.prop.ps . . . . .	165
size.ci.ratio.prop2 . . . . .	166
size.ci.rsqr . . . . .	167
size.ci.second . . . . .	167
size.ci.slope . . . . .	168
size.ci.spear . . . . .	169
size.ci.spear2 . . . . .	170
size.ci.stdmean.ps . . . . .	171
size.ci.stdmean2 . . . . .	172
size.ci.tetra . . . . .	173
size.equiv.mean.ps . . . . .	174
size.equiv.mean2 . . . . .	175
size.equiv.prop.ps . . . . .	176
size.equiv.prop2 . . . . .	177
size.interval.cor . . . . .	178
size.supinf.mean.ps . . . . .	179
size.supinf.mean2 . . . . .	180
size.supinf.prop.ps . . . . .	181
size.supinf.prop2 . . . . .	182
size.test.cor . . . . .	183
size.test.cor2 . . . . .	183
size.test.cronbach . . . . .	184
size.test.cronbach2 . . . . .	185
size.test.lc.ancova . . . . .	186
size.test.lc.mean.bs . . . . .	187
size.test.lc.mean.ws . . . . .	188
size.test.lc.prop.bs . . . . .	189
size.test.mann . . . . .	190
size.test.mean . . . . .	191
size.test.mean.ps . . . . .	191
size.test.mean2 . . . . .	192

size.test.prop . . . . .	193
size.test.prop.ps . . . . .	194
size.test.prop2 . . . . .	195
size.test.sign . . . . .	196
size.test.sign.ps . . . . .	196
size.test.slope . . . . .	197
slope.contrast . . . . .	198
spearmanbrown . . . . .	199
test.anova.bs . . . . .	199
test.cor . . . . .	200
test.cor2 . . . . .	201
test.kurtosis . . . . .	202
test.mean . . . . .	203
test.mono.mean.bs . . . . .	204
test.mono.prop.bs . . . . .	205
test.prop . . . . .	206
test.prop.bs . . . . .	207
test.prop.ps . . . . .	208
test.prop2 . . . . .	209
test.skew . . . . .	210
test.spear . . . . .	210
test.spear2 . . . . .	211

**Index****213**


---

adj.se	<i>Adjusted standard errors for slope coefficients in an exploratory analysis</i>
--------	---

---

**Description**

Computes an adjusted standard error in a general linear model after one or more predictor variables with nonsignificant slopes have been dropped from the model. The adjusted standard errors are then used to compute adjusted t-values, p-values, and confidence intervals. The mean square error and error degrees of freedom from the full model are used to compute the adjusted standard errors. These adjusted results are less susceptible to the negative effects of an exploratory model selection.

**Usage**

```
adj.se(alpha, mse1, mse2, dfe1, se, b)
```

**Arguments**

alpha	alpha level for 1-alpha confidence
mse1	mean squared error in full model
mse2	mean squared error in selected model
dfe1	error df in full model

se                    vector of slope standard errors in selected model  
 b                     vector of estimated slopes in selected model

**Value**

Returns adjusted standard error, t-statistic, p-value, and confidence interval for each slope coefficient

**Examples**

```
se <- c(1.57, 3.15, 0.982)
b <- c(3.78, 8.21, 2.99)
adj.se(.05, 10.26, 8.37, 114, se, b)

# Should return:
#      Estimate  adj SE      t  df      p      LL      UL
# [1,]    3.78  1.738243  2.174609 114 0.031725582 0.3365531  7.223447
# [2,]    8.21  3.487559  2.354082 114 0.020279958 1.3011734 15.118827
# [3,]    2.99  1.087233  2.750102 114 0.006930554 0.8362007  5.143799
```

---

ci.2x2.mean.bs                    *Computes tests and confidence intervals of effects in a 2x2 between-subjects design for means*

---

**Description**

Computes confidence intervals and tests for the AB interaction effect, main effect of A, main effect of B, simple main effects of A, and simple main effects of B in a 2x2 between-subjects design with a quantitative response variable. A Satterthwaite adjustment to the degrees of freedom is used and equality of population variances is not assumed.

**Usage**

```
ci.2x2.mean.bs(alpha, y11, y12, y21, y22)
```

**Arguments**

alpha                alpha level for 1-alpha confidence  
 y11                  vector of scores at level 1 of A and level 1 of B  
 y12                  vector of scores at level 1 of A and level 2 of B  
 y21                  vector of scores at level 2 of A and level 1 of B  
 y22                  vector of scores at level 2 of A and level 2 of B

**Value**

Returns a 7-row matrix (one row per effect). The columns are:

- Estimate - estimate of effect
- SE - standard error
- t - t test statistic
- df - degrees of freedom
- p - two-sided p-value
- LL - lower limit of the confidence interval
- UL - upper limit of the confidence interval

**Examples**

```

y11 <- c(14, 15, 11, 7, 16, 12, 15, 16, 10, 9)
y12 <- c(18, 24, 14, 18, 22, 21, 16, 17, 14, 13)
y21 <- c(16, 11, 10, 17, 13, 18, 12, 16, 6, 15)
y22 <- c(18, 17, 11, 9, 9, 13, 18, 15, 14, 11)
ci.2x2.mean.bs(.05, y11, y12, y21, y22)

# Should return:
#      Estimate      SE      t      df      p      LL      UL
# AB:      -5.10 2.224860 -2.29227953 35.47894 0.027931810 -9.6145264 -0.5854736
# A:         1.65 1.112430  1.48323970 35.47894 0.146840430 -0.6072632  3.9072632
# B:        -2.65 1.112430 -2.38217285 35.47894 0.022698654 -4.9072632 -0.3927368
# A at b1:  -0.90 1.545244 -0.58243244 17.56296 0.567678242 -4.1522367  2.3522367
# A at b2:   4.20 1.600694  2.62386142 17.93761 0.017246053  0.8362274  7.5637726
# B at a1:  -5.20 1.536952 -3.38331916 17.61093 0.003393857 -8.4341379 -1.9658621
# B at a2:  -0.10 1.608657 -0.06216365 17.91650 0.951120753 -3.4807927  3.2807927

```

---

ci.2x2.mean.mixed	<i>Computes tests and confidence intervals of effects in a 2x2 mixed design for means</i>
-------------------	---

---

**Description**

Computes confidence intervals and tests for the AB interaction effect, main effect of A, main effect of B, simple main effects of A, and simple main effects of B in a 2x2 mixed factorial design with a quantitative response variable where Factor A is a within-subjects factor, and Factor B is a between-subjects factor. A Satterthwaite adjustment to the degrees of freedom is used and equality of population variances is not assumed.

**Usage**

```
ci.2x2.mean.mixed(alpha, y11, y12, y21, y22)
```



**Arguments**

alpha	alpha level for 1-alpha confidence
y11	vector of scores at level 1 of A and level 1 of B
y12	vector of scores at level 1 of A and level 2 of B
y21	vector of scores at level 2 of A and level 1 of B
y22	vector of scores at level 2 of A and level 2 of B

**Value**

Returns a 7-row matrix (one row per effect). The columns are:

- Estimate - estimate of effect
- SE - standard error
- t - t test statistic
- df - degrees of freedom
- p - two-sided p-value
- LL - lower limit of the confidence interval
- UL - upper limit of the confidence interval

**Examples**

```

y11 <- c(18, 19, 20, 17, 20, 16)
y12 <- c(19, 18, 19, 20, 17, 16)
y21 <- c(19, 16, 16, 14, 16, 18)
y22 <- c(16, 10, 12, 9, 13, 15)
ci.2x2.mean.mixed(.05, y11, y12, y21, y22)

# Should return:
#      Estimate      SE      t      df      p      LL      UL
# AB:    -3.833333 0.9803627 -3.910117 8.346534 0.0041247610 -6.0778198 -1.588847
# A:      2.083333 0.4901814  4.250128 8.346534 0.0025414549  0.9610901  3.205577
# B:      3.750000 1.0226599  3.666908 7.601289 0.0069250119  1.3700362  6.129964
# A at b1: 0.1666667 0.8333333  0.200000 5.000000 0.8493605140 -1.9754849  2.308818
# A at b2: 4.0000000 0.5163978  7.745967 5.000000 0.0005732451  2.6725572  5.327443
# B at a1: 1.8333333 0.9803627  1.870056 9.943850 0.0911668588 -0.3527241  4.019391
# B at a2: 5.6666667 1.2692955  4.464419 7.666363 0.0023323966  2.7173445  8.615989

```

---

ci.2x2.mean.ws      *Computes tests and confidence intervals of effects in a 2x2 within-subjects design for means*

---

### Description

Computes confidence intervals and tests for the AB interaction effect, main effect of A, main effect of B, simple main effects of A, and simple main effects of B in a 2x2 within-subjects design with a quantitative response variable.

### Usage

```
ci.2x2.mean.ws(alpha, y11, y12, y21, y22)
```

### Arguments

alpha	alpha level for 1-alpha confidence
y11	vector of scores at level 1 of A and level 1 of B
y12	vector of scores at level 1 of A and level 2 of B
y21	vector of scores at level 2 of A and level 1 of B
y22	vector of scores at level 2 of A and level 2 of B

### Value

Returns a 7-row matrix (one row per effect). The columns are:

- Estimate - estimate of effect
- SE - standard error
- t - t test statistic
- df - degrees of freedom
- p - two-sided p-value
- LL - lower limit of the confidence interval
- UL - upper limit of the confidence interval

### Examples

```
y11 <- c(1,2,3,4,5,7,7)
y12 <- c(1,0,2,4,3,8,7)
y21 <- c(4,5,6,7,8,9,8)
y22 <- c(5,6,8,7,8,9,9)
ci.2x2.mean.ws(.05, y11, y12, y21, y22)

# Should return:
#           Estimate      SE      t df      p      LL      UL
# AB:      1.28571429 0.5654449  2.2738102  6 0.0633355395 -0.09787945  2.66930802
```

```

# A:      -3.21428571  0.4862042 -6.6109784  6 0.0005765210 -4.40398462 -2.02458681
# B:      -0.07142857  0.2296107 -0.3110855  6 0.7662600658 -0.63326579  0.49040865
# A at b1: -2.57142857  0.2973809 -8.6469203  6 0.0001318413 -3.29909331 -1.84376383
# A at b2: -3.85714286  0.7377111 -5.2285275  6 0.0019599725 -5.66225692 -2.05202879
# B at a1:  0.57142857  0.4285714  1.3333333  6 0.2308094088 -0.47724794  1.62010508
# B at a2: -0.71428571  0.2857143 -2.5000000  6 0.0465282323 -1.41340339 -0.01516804

```

---

ci.2x2.median.bs	<i>Computes tests and confidence intervals of effects in a 2x2 between-subjects design for medians</i>
------------------	--

---

### Description

Computes distribution-free confidence intervals for the AB interaction effect, main effect of A, main effect of B, simple main effects of A, and simple main effects of B in a 2x2 between-subjects design with a quantitative response variable. The effects are defined in terms of medians rather than means. Tied scores are assumed to be rare.

### Usage

```
ci.2x2.median.bs(alpha, y11, y12, y21, y22)
```

### Arguments

alpha	alpha level for 1-alpha confidence
y11	vector of scores at level 1 of A and level 1 of B
y12	vector of scores at level 1 of A and level 2 of B
y21	vector of scores at level 2 of A and level 1 of B
y22	vector of scores at level 2 of A and level 2 of B

### Value

Returns a 7-row matrix (one row per effect). The columns are:

- Estimate - estimate of effect
- 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.

**Examples**

```

y11 <- c(19.2, 21.1, 14.4, 13.3, 19.8, 15.9, 18.0, 19.1, 16.2, 14.6)
y12 <- c(21.3, 27.0, 19.1, 21.5, 25.2, 24.1, 19.8, 19.7, 17.5, 16.0)
y21 <- c(16.5, 11.3, 10.3, 17.7, 13.8, 18.2, 12.8, 16.2, 6.1, 15.2)
y22 <- c(18.7, 17.3, 11.4, 12.4, 13.6, 13.8, 18.3, 15.0, 14.4, 11.9)
ci.2x2.median.bs(.05, y11, y12, y21, y22)

```

```

# Should return:
#           Estimate      SE      LL      UL
# AB:      -3.850 2.951019 -9.633891  1.9338914
# A:        4.525 1.475510  1.633054  7.4169457
# B:       -1.525 1.475510 -4.416946  1.3669457
# A at b1:  2.600 1.992028 -1.304302  6.5043022
# A at b2:  6.450 2.177232  2.182703 10.7172971
# B at a1: -3.450 2.045086 -7.458294  0.5582944
# B at a2:  0.400 2.127472 -3.769769  4.5697694

```

---

ci.2x2.median.mixed     *Computes confidence intervals of effects in a 2x2 mixed design for medians*

---

**Description**

Computes distribution-free confidence intervals for the AB interaction effect, main effect of A, main effect of B, simple main effects of A, and simple main effects of B in a 2x2 mixed design where Factor A is the within-subjects factor and Factor B is the between-subjects factor. Effects are defined in terms of medians rather than means. Tied scores are assumed to be rare.

**Usage**

```
ci.2x2.median.mixed(alpha, y11, y12, y21, y22)
```

**Arguments**

alpha	alpha level for 1-alpha confidence
y11	vector of scores at level 1 of A in group 1
y12	vector of scores at level 2 of A in group 1
y21	vector of scores at level 1 of A in group 2
y22	vector of scores at level 2 of A in group 2

**Value**

Returns a 7-row matrix (one row per effect). The columns are:

- Estimate - estimate of effect

- SE - standard error
- 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.

## Examples

```

y11 <- c(18.3, 19.5, 20.1, 17.4, 20.5, 16.1)
y12 <- c(19.1, 18.4, 19.8, 20.0, 17.2, 16.8)
y21 <- c(19.2, 16.4, 16.5, 14.0, 16.9, 18.3)
y22 <- c(16.5, 10.2, 12.7, 9.9, 13.5, 15.0)
ci.2x2.median.mixed(.05, y11, y12, y21, y22)

# Should return:
#           Estimate      SE      LL      UL
# AB:      -3.450 1.6317863 -6.6482423 -0.2517577
# A:         1.875 0.8158931  0.2758788  3.4741212
# B:         3.925 1.4262367  1.1296274  6.7203726
# A at b1:   0.150 1.4243192 -2.6416144  2.9416144
# A at b2:   3.600 0.7962670  2.0393454  5.1606546
# B at a1:   2.200 1.5812792 -0.8992503  5.2992503
# B at a2:   5.650 1.7027101  2.3127496  8.9872504

```

---

ci.2x2.median.ws

*Computes confidence intervals of effects in a 2x2 within-subjects design for medians*

---

## Description

Computes distribution-free confidence intervals for the AB interaction effect, main effect of A, main effect of B, simple main effects of A, and simple main effects of B in a 2x2 within-subjects design. The effects are defined in terms of medians rather than means. Tied scores are assumed to be rare.

## Usage

```
ci.2x2.median.ws(alpha, y11, y12, y21, y22)
```

**Arguments**

alpha	alpha level for 1-alpha confidence
y11	vector of scores at level 1 of A and level 1 of B
y12	vector of scores at level 1 of A and level 2 of B
y21	vector of scores at level 2 of A and level 1 of B
y22	vector of scores at level 2 of A and level 2 of B

**Value**

Returns a 7-row matrix (one row per effect). The columns are:

- Estimate - estimate of effect
- SE - standard error
- 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).

**Examples**

```

y11 <- c(221, 402, 333, 301, 284, 182, 281, 230, 290, 182, 133, 278)
y12 <- c(221, 371, 340, 288, 293, 150, 317, 211, 286, 161, 126, 234)
y21 <- c(219, 371, 314, 279, 284, 155, 278, 185, 296, 169, 118, 229)
y22 <- c(170, 332, 280, 273, 272, 160, 260, 204, 252, 153, 137, 221)
ci.2x2.median.ws(.05, y11, y12, y21, y22)

# Should return:
#      Estimate      SE      LL      UL
# AB:      2.50 21.050122 -38.757482 43.75748
# A:      24.75  9.603490  5.927505 43.57250
# B:      18.25  9.101881  0.410641 36.08936
# A at b1: 26.00 11.813742  2.845491 49.15451
# A at b2: 23.50 16.323093 -8.492675 55.49267
# B at a1: 19.50 15.710347 -11.291715 50.29171
# B at a2: 17.00 11.850202 -6.225970 40.22597

```

---

ci.2x2.prop.bs	<i>Computes tests and confidence intervals of effects in a 2x2 between-subjects design for proportions</i>
----------------	--

---

### Description

Computes adjusted Wald confidence intervals and tests for the AB interaction effect, main effect of A, main effect of B, simple main effects of A, and simple main effects of B in a 2x2 between-subjects factorial design with a dichotomous response variable. The input vector of frequency counts is  $f = [f_{11}, f_{12}, f_{21}, f_{22}]$ , and the input vector of sample sizes is  $n = [n_{11}, n_{12}, n_{21}, n_{22}]$  where the first subscript represents the levels of Factor A and the second subscript represents the levels of Factor B.

### Usage

```
ci.2x2.prop.bs(alpha, f, n)
```

### Arguments

alpha	alpha level for 1-alpha confidence
f	vector of frequency counts of participants who have the attribute
n	vector of sample sizes

### Value

Returns a 7-row matrix (one row per effect). The columns are:

- Estimate - adjusted estimate of effect
- SE - standard error
- z - z test statistic for test of null hypothesis
- p - two-sided 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).

**Examples**

```
f <- c(15, 24, 28, 23)
n <- c(50, 50, 50, 50)
ci.2x2.prop.bs(.05, f, n)

# Should return:
#           Estimate      SE      z      p      LL      UL
# AB:      -0.27450980  0.13692496 -2.0048193 0.044982370 -0.54287780 -0.00614181
# A:       -0.11764706  0.06846248 -1.7184165 0.085720668 -0.25183106  0.01653694
# B:       -0.03921569  0.06846248 -0.5728055 0.566776388 -0.17339968  0.09496831
# A at b1: -0.25000000  0.09402223 -2.6589456 0.007838561 -0.43428019 -0.06571981
# A at b2:  0.01923077  0.09787658  0.1964798 0.844234654 -0.17260380  0.21106534
# B at a1: -0.17307692  0.09432431 -1.8349132 0.066518551 -0.35794917  0.01179533
# B at a2:  0.09615385  0.09758550  0.9853293 0.324462356 -0.09511021  0.28741790
```

---

ci.2x2.prop.mixed	<i>Computes tests and confidence intervals of effects in a 2x2 mixed factorial design for proportions</i>
-------------------	---

---

**Description**

Computes adjusted Wald confidence intervals and tests for the AB interaction effect, main effect of A, main effect of B, simple main effects of A, and simple main effects of B in a 2x2 mixed factorial design with a dichotomous response variable where Factor A is a within-subjects factor and Factor B is a between-subjects factor. The 4x1 vector of frequency counts for Factor A within each group is f00, f01, f10, f11 where  $f_{ij}$  is the number of participants with a response of  $i = 0$  or 1 at level 1 of Factor A and a response of  $j = 0$  or 1 at level 2 of Factor A.

**Usage**

```
ci.2x2.prop.mixed(alpha, group1, group2)
```

**Arguments**

alpha	alpha level for 1-alpha confidence
group1	vector of frequency counts from 2x2 contingency table in group 1
group2	vector of frequency counts from 2x2 contingency table in group 2

**Value**

Returns a 7-row matrix (one row per effect). The columns are:

- Estimate - adjusted estimate of effect
- SE - standard error of estimate
- z - z test statistic



- p - two-sided p-value
- LL - lower limit of the adjusted Wald confidence interval
- UL - upper limit of the adjusted Wald confidence interval

### Examples

```
group1 <- c(125, 14, 10, 254)
group2 <- c(100, 16, 9, 275)
ci.2x2.prop.mixed (.05, group1, group2)

# Should return:
#           Estimate      SE      z      p      LL      UL
# AB:      0.007555369 0.017716073 0.4264697 0.66976559 -0.02716750 0.042278234
# A:      -0.013678675 0.008858036 -1.5442107 0.12253730 -0.03104011 0.003682758
# B:      -0.058393219 0.023032656 -2.5352360 0.01123716 -0.10353640 -0.013250043
# A at b1: -0.009876543 0.012580603 -0.7850612 0.43241768 -0.03453407 0.014780985
# A at b2: -0.017412935 0.012896543 -1.3502018 0.17695126 -0.04268969 0.007863824
# B at a1: -0.054634236 0.032737738 -1.6688458 0.09514794 -0.11879902 0.009530550
# B at a2: -0.062170628 0.032328556 -1.9230871 0.05446912 -0.12553343 0.001192177
```

---

ci.2x2.stdmean.bs      *Computes confidence intervals of standardized effects in a 2x2 between-subjects design*

---

### Description

Computes confidence intervals for standardized linear contrasts of means (AB interaction, main effect of A, main effect of B, simple main effects of A, and simple main effects of B) in a 2x2 between-subjects design with a quantitative response variable. Equality of population variances is not assumed. An unweighted variance standardizer is used, which is the recommended standardizer when both factors are treatment factors.

### Usage

```
ci.2x2.stdmean.bs(alpha, y11, y12, y21, y22)
```

### Arguments

alpha	alpha level for 1-alpha confidence
y11	vector of scores at level 1 of A and level 1 of B
y12	vector of scores at level 1 of A and level 2 of B
y21	vector of scores at level 2 of A and level 1 of B
y22	vector of scores at level 2 of A and level 2 of B

**Value**

Returns a 7-row matrix (one row per effect). The columns are:

- Estimate - estimate of standardized effect
- adj Estimate - bias adjusted estimate of standardized effect
- 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.

**Examples**

```

y11 <- c(14, 15, 11, 7, 16, 12, 15, 16, 10, 9)
y12 <- c(18, 24, 14, 18, 22, 21, 16, 17, 14, 13)
y21 <- c(16, 11, 10, 17, 13, 18, 12, 16, 6, 15)
y22 <- c(18, 17, 11, 9, 9, 13, 18, 15, 14, 11)
ci.2x2.stdmean.bs(.05, y11, y12, y21, y22)

# Should return:
#           Estimate adj Estimate      SE      LL      UL
# AB:      -1.44976487 -1.4193502 0.6885238 -2.7992468 -0.1002829
# A:         0.46904158  0.4592015 0.3379520 -0.1933321  1.1314153
# B:        -0.75330920 -0.7375055 0.3451209 -1.4297338 -0.0768846
# A at b1: -0.25584086 -0.2504736 0.4640186 -1.1653006  0.6536189
# A at b2:  1.19392401  1.1688767 0.5001423  0.2136630  2.1741850
# B at a1: -1.47819163 -1.4471806 0.4928386 -2.4441376 -0.5122457
# B at a2: -0.02842676 -0.0278304 0.4820369 -0.9732017  0.9163482

```

---

ci.2x2.stdmean.mixed *Computes confidence intervals of standardized effects in a 2x2 mixed design*

---

**Description**

Computes confidence intervals for the standardized AB interaction effect, main effect of A, main effect of B, simple main effects of A, and simple main effects of B in a 2x2 mixed factorial design where Factor A is a within-subjects factor, and Factor B is a between-subjects factor. Equality of population variances is not assumed.

**Usage**

```
ci.2x2.stdmean.mixed(alpha, y11, y12, y21, y22)
```

**Arguments**

alpha	alpha level for 1-alpha confidence
y11	vector of scores at level 1 of A in group 1
y12	vector of scores at level 2 of A in group 1
y21	vector of scores at level 1 of A in group 2
y22	vector of scores at level 2 of A in group 2

**Value**

Returns a 7-row matrix (one row per effect). The columns are:

- Estimate - estimated standardized effect
- adj Estimate - bias adjusted standardized effect estimate
- 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).

**Examples**

```

y11 <- c(18, 19, 20, 17, 20, 16)
y12 <- c(19, 18, 19, 20, 17, 16)
y21 <- c(19, 16, 16, 14, 16, 18)
y22 <- c(16, 10, 12, 9, 13, 15)
ci.2x2.stdmean.mixed(.05, y11, y12, y21, y22)

# Should return:
#           Estimate adj Estimate      SE      LL      UL
# AB:      -1.95153666 -1.80141845 0.5424100 -3.0146407 -0.8884326
# A:         1.06061775  1.01125934 0.2780119  0.5157244  1.6055111
# B:         1.90911195  1.76225718 0.5743510  0.7834047  3.0348192
# A at b1:  0.08484942  0.07589163 0.4649598 -0.8264549  0.9961538
# A at b2:  2.03638608  1.82139908 0.2964013  1.4554502  2.6173219
# B at a1:  0.93334362  0.86154796 0.5487927 -0.1422703  2.0089575
# B at a2:  2.88488027  2.66296641 0.7127726  1.4878717  4.2818889

```

---

ci.2x2.stdmean.ws      *Computes confidence intervals of standardized effects in a 2x2 within-subjects design*

---

### Description

Computes confidence intervals for standardized linear contrasts of means (AB interaction, main effect of A, main effect of B, simple main effects of A, and simple main effects of B) in a 2x2 within-subjects design. Equality of population variances is not assumed. An unweighted variance standardizer is used.

### Usage

```
ci.2x2.stdmean.ws(alpha, y11, y12, y21, y22)
```

### Arguments

alpha	alpha level for 1-alpha confidence
y11	vector of scores at level 1 of A and level 1 of B
y12	vector of scores at level 1 of A and level 2 of B
y21	vector of scores at level 2 of A and level 1 of B
y22	vector of scores at level 2 of A and level 2 of B

### Value

Returns a 7-row matrix (one row per effect). The columns are:

- Estimate - estimated standardized effect
- adj Estimate - bias adjusted standardized effect estimate
- 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).

### Examples

```
y11 <- c(21, 39, 32, 29, 27, 17, 27, 21, 28, 17, 12, 27)
y12 <- c(20, 36, 33, 27, 28, 14, 30, 20, 27, 15, 11, 22)
y21 <- c(21, 36, 30, 27, 28, 15, 27, 18, 29, 16, 11, 22)
y22 <- c(18, 34, 29, 28, 28, 17, 27, 21, 26, 16, 14, 23)
ci.2x2.stdmean.ws(.05, y11, y12, y21, y22)
```

```

# Should return:
#           Estimate adj Estimate      SE      LL      UL
# AB:      0.17248839  0.16446123 0.13654635 -0.095137544 0.4401143
# A:       0.10924265  0.10415878 0.05752822 -0.003510596 0.2219959
# B:       0.07474497  0.07126653 0.05920554 -0.041295751 0.1907857
# A at b1: 0.19548684  0.18638939 0.08460680  0.029660560 0.3613131
# A at b2: 0.02299845  0.02192816 0.09371838 -0.160686202 0.2066831
# B at a1: 0.16098916  0.15349715 0.09457347 -0.024371434 0.3463498
# B at a2: -0.01149923 -0.01096408 0.08595873 -0.179975237 0.1569768

```

---

ci.agree

*Confidence interval for a G-index of agreement*


---

### Description

Computes an adjusted Wald 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. The G-index corrects for chance agreement.

### 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 - maximum likelihood estimate of G-index
- SE - standard error
- LL - lower limit of the adjusted Wald confidence interval
- UL - upper limit of the adjusted Wald confidence interval

### References

Bonett DG (2022). “Statistical inference for G-indices of agreement.” *Journal of Educational and Behavioral Statistics*, **47**(4), 438–458. doi:10.3102/10769986221088561.

**Examples**

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

# Should return:
# Estimate      SE      LL      UL
# 0.7333333 0.05333333 0.6132949 0.8226025
```

---

ci.agree.3rater	<i>Computes confidence intervals for a 3-rater design with dichotomous ratings</i>
-----------------	--

---

**Description**

Computes adjusted Wald confidence intervals for a G-index of agreement for all pairs of raters in a 3-rater design with a dichotomous rating, and computes adjusted Wald confidence intervals for differences of all pairs of G agreement. An adjusted Wald confidence interval for unanimous G agreement among the three raters is also computed. In the three-rater design, unanimous G agreement is equal to the average of all pairs of G agreement. The G-index corrects for chance agreement.

**Usage**

```
ci.agree.3rater(alpha, f)
```

**Arguments**

alpha	alpha level for 1-alpha confidence
f	vector of frequency counts from 2x2x2 table where $f = [f_{111}, f_{112}, f_{121}, f_{122}, f_{211}, f_{212}, f_{221}, f_{222}]$ , first subscript represents the rating of rater 1, second subscript represents the rating of rater 2, and third subscript represents the rating of rater 3

**Value**

Returns a 7-row matrix. The rows are:

- G(1,2): G-index for raters 1 and 2
- G(1,3): G-index for raters 1 and 3
- G(2,3): G-index for raters 2 and 3
- G(1,2)-G(1,3): difference in G(1,2) and G(1,3)
- G(1,2)-G(2,3): difference in G(1,2) and G(2,3)
- G(2,3)-G(1,3): difference in G(2,3) and G(1,3)
- G(3): G-index of unanimous agreement for all three raters

The columns are:

- Estimate - estimate of G-index (two-rater, difference, or unanimous)
- LL - lower limit of adjusted Wald confidence interval
- UL - upper limit of adjusted Wald confidence interval

## References

Bonett DG (2022). “Statistical inference for G-indices of agreement.” *Journal of Educational and Behavioral Statistics*, **47**(4), 438–458. doi:10.3102/10769986221088561.

## Examples

```
f <- c(100, 6, 4, 40, 20, 1, 9, 120)
ci.agree.3rater(.05, f)

# Should return:
#           Estimate      LL      UL
# G(1,2)      0.56666667  0.46601839  0.6524027
# G(1,3)      0.50000000  0.39564646  0.5911956
# G(2,3)      0.86666667  0.79701213  0.9135142
# G(1,2)-G(1,3) 0.06666667  0.00580397  0.1266464
# G(1,2)-G(2,3) -0.30000000 -0.40683919 -0.1891873
# G(2,3)-G(1,3) -0.36666667 -0.46222023 -0.2662566
# G(3)        0.64444444  0.57382971  0.7068720
```

---

ci.agree2

*Confidence interval for G-index difference in a 2-group design*

---

## Description

Computes adjusted Wald confidence intervals for the G-index of agreement within each group and the difference of G-indices.

## Usage

```
ci.agree2(alpha, n1, f1, n2, f2, r)
```

## Arguments

alpha	alpha level for simultaneous 1-alpha confidence
n1	sample size (objects) in group 1
f1	number of objects rated in agreement in group 1
n2	sample size (objects) in group 2
f2	number of objects rated in agreement in group 2
r	number of rating categories

**Value**

Returns a 3-row matrix. The rows are:

- Row 1: G-index for group 1
- Row 2: G-index for group 2
- Row 3: G-index difference

The columns are:

- Estimate - maximum likelihood estimate of G-index and difference
- SE - standard error
- LL - lower limit of adjusted Wald confidence interval
- UL - upper limit of adjusted Wald confidence interval

**References**

Bonett DG (2022). “Statistical inference for G-indices of agreement.” *Journal of Educational and Behavioral Statistics*, **47**(4), 438–458. doi:10.3102/10769986221088561.

**Examples**

```
ci.agree2(.05, 75, 70, 60, 45, 2)

# Should return:
#      Estimate      SE      LL      UL
# G1      0.8666667 0.02880329 0.6974555 0.9481141
# G2      0.5000000 0.05590170 0.2523379 0.6851621
# G1 - G2 0.3666667 0.06288585 0.1117076 0.6088621
```

---

ci.bayes.normal      *Bayesian credible interval for a normal prior distribution*

---

**Description**

Computes an approximate Bayesian credible interval for a normal prior distribution. This function can be used with any parameter estimator (e.g., mean, mean difference, linear contrast of means, slope coefficient, standardized mean difference, standardized linear contrast of means, median, median difference, linear contrast of medians, etc.) that has an approximate normal sampling distribution. The mean and standard deviation of the posterior normal distribution are also reported.

**Usage**

```
ci.bayes.normal(alpha, prior.mean, prior.sd, est, se)
```



**Arguments**

alpha	alpha level for 1-alpha credibility interval
prior.mean	mean of prior Normal distribution
prior.sd	standard deviation of prior Normal distribution
est	sample estimate
se	standard error of sample estimate

**Value**

Returns a 1-row matrix. The columns are:

- Posterior mean - posterior mean of Normal distribution
- Posterior SD - posterior standard deviation of Normal distribution
- LL - lower limit of the credible interval
- UL - upper limit of the credible interval

**References**

Gelman A, B. CJ, Stern HS, Rubin DB (2004). *Bayesian Data Analysis*, 2nd edition. Chapman & Hall.

**Examples**

```
ci.bayes.normal(.05, 30, 2, 24.5, 0.577)

# Should return:
# Posterior mean Posterior SD      LL      UL
#      24.9226      0.5543895 23.83602 26.00919
```

---

ci.bayes.prop                      *Bayesian credible interval for a proportion*

---

**Description**

Computes a Bayesian credible interval for a population proportion using the mean and standard deviation of a prior Beta distribution along with sample information. The mean and standard deviation of the posterior Beta distribution are also reported. For a noninformative prior, set the prior mean to .5 and the prior standard deviation to  $1/\sqrt{12}$  (which corresponds to a Beta(1,1) distribution). The prior variance must be less than  $m(1 - m)$  where  $m$  is the prior mean.

**Usage**

```
ci.bayes.prop(alpha, prior.mean, prior.sd, f, n)
```

**Arguments**

alpha	alpha level for 1-alpha credibility interval
prior.mean	mean of prior Beta distribution
prior.sd	standard deviation of prior Beta distribution
f	number of participants who have the attribute
n	sample size

**Value**

Returns a 1-row matrix. The columns are:

- Posterior mean - posterior mean of Beta distribution
- Posterior SD - posterior standard deviation of Beta distribution
- LL - lower limit of the credible interval
- UL - upper limit of the credible interval

**References**

Gelman A, B. CJ, Stern HS, Rubin DB (2004). *Bayesian Data Analysis*, 2nd edition. Chapman & Hall.

**Examples**

```
ci.bayes.prop(.05, .4, .1, 12, 100)

# Should return:
# Posterior mean Posterior SD      LL      UL
#           0.15   0.03273268  0.09218  0.2188484
```

---

ci.biphi

*Confidence interval for a biserial-phi correlation*


---

**Description**

Computes a confidence interval for a population biserial-phi correlation 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 for group 1
n2	sample size for group 2

**Value**

Returns a 1-row matrix. The columns are:

- Estimate - estimate of biserial-phi correlation
- 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).

**Examples**

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

# Should return:
# Estimate      SE      LL      UL
# 0.4145733 0.07551281 0.2508866 0.546141
```

---

ci.bscor

*Confidence interval for a biserial correlation*


---

**Description**

Computes a confidence interval for a population biserial correlation. A biserial correlation can be used when one variable is quantitative and the other variable has been artificially dichotomized to create two groups. The biserial correlation estimates the correlation between the observed quantitative variable and the unobserved quantitative variable that has been measured on a dichotomous scale.

**Usage**

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

**Arguments**

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

**Details**

This function computes a point-biserial correlation and its standard error as a function of a standardized mean difference with a weighted variance standardizer. Then the point-biserial estimate is transformed into a biserial correlation using the traditional adjustment. The adjustment is also applied to the point-biserial standard error to obtain the standard error for the biserial correlation.

The biserial correlation assumes that the observed quantitative variable and the unobserved quantitative variable have a bivariate normal distribution. Bivariate normality is a crucial assumption underlying the transformation of a point-biserial correlation to a biserial correlation. Bivariate normality also implies equal variances of the observed quantitative variable at each level of the dichotomized variable, and this assumption is made in the computation of the standard error.

**Value**

Returns a 1-row matrix. The columns are:

- Estimate - estimated biserial correlation
- 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).

**Examples**

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

# Should return:
# Estimate      SE      LL      UL
# 0.8855666 0.06129908 0.7376327 0.984412
```

---

`ci.cod`*Confidence interval for a coefficient of dispersion*

---

**Description**

Computes a confidence interval for a population coefficient of dispersion which is defined as a mean absolute deviation from the median divided by a median. The coefficient of dispersion assumes ratio-scale scores and is a robust alternative to the coefficient of variation. An approximate standard error is recovered from the confidence interval.

**Usage**

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

**Arguments**

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

**Value**

Returns a 1-row matrix. The columns are:

- Estimate - estimated coefficient of dispersion
- SE - recovered standard error
- 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).

**Examples**

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

# Should return:
# Estimate      SE      LL      UL
# 0.5921053 0.1814708 0.3813259 1.092679
```

---

ci.condslope                      *Confidence intervals 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 (x1), a moderator variable (x2), and a product predictor variable (x1\*x2). Conditional slopes are computed at specified 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	estimated slope coefficient for predictor variable
b2	estimated slope coefficient for product variable
se1	standard error for predictor coefficient
se2	standard error for product coefficient
cov	estimated 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 conditional 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 intervals for conditional (simple) slopes in a logistic model*

---

### Description

Computes confidence intervals and test statistics for population conditional slopes (simple slopes) in a logistic model that includes a predictor variable (x1), a moderator variable (x2), and a product predictor variable (x1\*x2). 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	estimated slope coefficient for predictor variable
b2	estimated slope coefficient for product variable
se1	standard error for predictor coefficient
se2	standard error for product coefficient
cov	estimated 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 conditional slope
- exp(Estimate) - estimated exponentiated conditional slope
- z - z test statistic
- p - two-sided p-value
- LL - lower limit of the exponentiated confidence interval
- UL - upper limit of the exponentiated 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. A bias adjustment is used to reduce the bias of the Fisher transformed correlation. This function uses an estimated correlation as input. Use the `cor.test` function for raw data input.

**Usage**

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

**Arguments**

alpha	alpha level for 1-alpha confidence
cor	estimated 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
# 0.536 0.1018149 0.2978573 0.7058914
```

---

ci.cor.dep

*Confidence interval for a difference in dependent Pearson correlations*

---

## 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. A bias adjustment is used to reduce the bias of each Fisher transformed correlation. An approximate standard error is recovered from the confidence interval.

## Usage

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

## Arguments

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

## Value

Returns a 1-row matrix. The columns are:

- Estimate - estimated correlation difference
- SE - recovered standard error
- 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).

## Examples

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

# Should return:
# Estimate      SE      LL      UL
# 0.217 0.1026986 0.01323072 0.415802
```

---

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. A bias adjustment is used to reduce the bias of each Fisher transformed correlation.

## Usage

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

## Arguments

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

## Value

Returns a 1-row matrix. The columns are:

- Estimate - estimated correlation difference
- SE - standard error
- 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).

**Examples**

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

# Should return:
# Estimate      SE      LL      UL
# 0.084 0.02967934 0.02803246 0.1463609
```

---

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 correlations could also be correlations between two latent factors. The function requires a point estimate and a 100(1 - alpha)% confidence interval for each correlation as input. The confidence intervals can be obtained using the ci.fisher function.

**Usage**

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

**Arguments**

cor1	estimated correlation for group 1
ll1	lower limit for group 1 correlation
ul1	upper limit for group 1 correlation
cor2	estimated correlation for 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).

**Examples**

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

# Should return:
# Estimate  LL      UL
#      0.2 0.07 0.3220656
```

---

ci.cqv

---

*Confidence interval for a coefficient of quartile variation*


---

**Description**

Computes a distribution-free confidence interval for a population coefficient of quartile variation which is defined as  $(Q3 - Q1)/(Q3 + Q1)$  where  $Q1$  is the 25th percentile and  $Q3$  is the 75th percentile. The coefficient of quartile variation assumes ratio-scale scores and is a robust alternative to the coefficient of variation. The 25th and 75th percentiles are computed using the type = 2 method (SAS default).

**Usage**

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

**Arguments**

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

**Value**

Returns a 1-row matrix. The columns are:

- Estimate - estimated coefficient of quartile variation
- SE - standard error
- LL - lower limit of the confidence interval
- UL - upper limit of the confidence interval

**References**

Bonett DG (2006). "Confidence interval for a coefficient of quartile variation." *Computational Statistics and Data Analysis*, **50**(11), 2953–2957. doi:10.1016/j.csda.2005.05.007.

**Examples**

```

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

# Should return:
# Estimate      SE      LL      UL
#      0.5 0.1552485 0.2617885 0.8841821

```

---

ci.cramer

*Confidence interval for Cramer's V*


---

**Description**

Computes a confidence interval for a population Cramer's V coefficient of nominal association for an  $r \times s$  contingency table and its approximate standard error. The confidence interval is based on a noncentral chi-square distribution, and an approximate standard error is recovered from the confidence interval.

**Usage**

```
ci.cramer(alpha, chisqr, r, c, n)
```

**Arguments**

alpha	alpha value for 1-alpha confidence
chisqr	Pearson chi-square test statistic of independence
r	number of rows in contingency table
c	number of columns in contingency table
n	sample size

**Value**

Returns a 1-row matrix. The columns are:

- Estimate - estimate of Cramer's V
- SE - recovered standard error
- LL - lower limit of the confidence interval
- UL - upper limit of the confidence interval

**References**

Smithson M (2003). *Confidence Intervals*. Sage.

**Examples**

```
ci.cramer(.05, 19.21, 2, 3, 200)

# Should return:
# Estimate    SE    LL    UL
# 0.3099 0.0718 0.1601 0.4417
```

---

ci.cronbach

*Confidence interval for a Cronbach reliability*


---

**Description**

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

**Usage**

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

**Arguments**

alpha	alpha level for 1-alpha confidence
rel	estimated Cronbach reliability
r	number of measurements (items, raters, etc.)
n	sample size

**Value**

Returns a 1-row matrix. The columns are:

- Estimate - estimated Cronbach reliability (from input)
- 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).

**Examples**

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

# Should return:
# Estimate      SE      LL      UL
#      0.85  0.02456518 0.7971254 0.8931436
```

---

ci.cronbach2	<i>Confidence interval for a difference in Cronbach reliabilities in a 2-group design</i>
--------------	---

---

**Description**

Computes a confidence interval for a difference in population Cronbach reliability coefficients in a 2-group design. The number of measurements (e.g., items or raters) used in each group need not be equal.

**Usage**

```
ci.cronbach2(alpha, rel1, rel2, r1, r2, n1, n2)
```

**Arguments**

alpha	alpha level for 1-alpha confidence
rel1	estimated Cronbach reliability for group 1
rel2	estimated Cronbach reliability for group 2
r1	number of measurements used in group 1
r2	number of measurements used in group 2
n1	sample size for group 1
n2	sample size for group 2

**Value**

Returns a 1-row matrix. The columns are:

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

**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).

**Examples**

```
ci.cronbach2(.05, .88, .76, 8, 8, 200, 250)

# Should return:
# Estimate      LL      UL
#    0.12 0.06973411 0.173236
```

---

ci.cv

*Confidence interval for a coefficient of variation*


---

**Description**

Computes a confidence interval for a population coefficient of variation (standard deviation divided by mean). This confidence interval is the reciprocal of a confidence interval for a standardized mean (see [ci.stdmean](#)). An approximate standard error is recovered from the confidence interval. The coefficient of variation assumes ratio-scale scores.

**Usage**

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

**Arguments**

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

**Value**

Returns a 1-row matrix. The columns are:

- Estimate - estimated coefficient of variation
- SE - recovered standard error
- LL - lower limit of the confidence interval
- UL - upper limit of the confidence interval

**Examples**

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

# Should return:
# Estimate      SE      LL      UL
# 0.1489796 0.01817373 0.1214381 0.1926778
```



---

`ci.etasqr`*Confidence interval for eta-squared*

---

**Description**

Computes a confidence interval for a population eta-squared, partial eta-squared, or generalized eta-squared in a fixed-factor between-subjects design. An approximate bias adjusted estimate is computed, and an approximate standard error is recovered from the confidence interval.

**Usage**

```
ci.etasqr(alpha, etasqr, df1, df2)
```

**Arguments**

alpha	alpha value for 1-alpha confidence
etasqr	estimated eta-squared
df1	degrees of freedom for effect
df2	error degrees of freedom

**Value**

Returns a 1-row matrix. The columns are:

- Eta-squared - eta-squared (from input)
- adj Eta-squared - bias adjusted eta-squared estimate
- SE - recovered standard error
- LL - lower limit of the confidence interval
- UL - upper limit of the confidence interval

**Examples**

```
ci.etasqr(.05, .241, 3, 116)

# Should return:
# Eta-squared  adj Eta-squared      SE      LL      UL
#      0.241      0.2213707 0.06258283 0.1040229 0.3493431
```

---

`ci.fisher`*Fisher confidence interval*

---

### Description

Computes a Fisher confidence interval for any type of correlation (e.g., Pearson, Spearman, Kendall-tau, tetrachoric, phi, partial, semipartial, etc.) or ordinal association such as gamma, Somers' d, or tau-b. The correlation could also be between two latent factors obtained from a SEM analysis (the Fisher CI will be more accurate than the large-sample CI from a SEM analysis). The standard error can be a traditional standard error, a bootstrap standard error, or a robust standard error from a SEM analysis.

### Usage

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

### Arguments

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

### Value

Returns a 1-row matrix. The columns are:

- Estimate - correlation (from input)
- LL - lower limit of the confidence interval
- UL - upper limit of the confidence interval

### Examples

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

# Should return:
# Estimate      LL      UL
# 0.641 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 and a Sobel standard error. This function is not recommended for a standardized indirect effect. 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-consistent standard errors, and (in models with no direct effects) distribution-free Theil-Sen slope estimates with recovered standard errors also could be used.

**Usage**

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

**Arguments**

<code>alpha</code>	alpha level for 1-alpha confidence
<code>b1</code>	unstandardized slope estimate for first path
<code>b2</code>	unstandardized slope estimate for second path
<code>se1</code>	standard error for b1
<code>se2</code>	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 (within sampling error):
# Estimate      SE      LL      UL
# 4.7616 1.625282 2.178812 7.972262
```

---

 ci.kappa

*Confidence interval for two kappa reliability coefficients*


---

**Description**

Computes confidence intervals for the intraclass kappa coefficient and Cohen's kappa coefficient with two dichotomous ratings.

**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.gen.bs	<i>Confidence interval for a linear contrast of parameters in a between-subjects design</i>
--------------	---

---

### Description

Computes the estimate, standard error, and approximate confidence interval for a linear contrast of any type of parameter (e.g., quartile, logistic regression slope, path coefficient) where each parameter value has been estimated from a different sample. The parameter values are assumed to be of the same type (e.g., all unstandardized path coefficients) and their sampling distributions are assumed to be approximately normal.

### Usage

```
ci.lc.gen.bs(alpha, est, se, v)
```

### Arguments

alpha	alpha level for 1-alpha confidence
est	vector of parameter estimates
se	vector of standard errors
v	vector of contrast coefficients

### Value

Returns a 1-row matrix. The columns are:

- Estimate - estimate of linear contrast
- SE - standard error of linear contrast
- LL - lower limit of confidence interval
- UL - upper limit of confidence interval

### Examples

```
est <- c(3.86, 4.57, 2.29, 2.88)
se <- c(0.185, 0.365, 0.275, 0.148)
v <- c(.5, .5, -.5, -.5)
ci.lc.gen.bs(.05, est, se, v)

# Should return:
# Estimate      SE      LL      UL
#      1.63 0.2573806 1.125543 2.134457
```

---

ci.lc.glm	<i>Confidence interval for a linear contrast of general linear model parameters</i>
-----------	---

---

### Description

Computes the estimate, standard error, and confidence interval for a linear contrast of parameters in a general linear model using `coef(object)` and `vcov(object)` where "object" is a fitted model object from the `lm` function.

### Usage

```
ci.lc.glm(alpha, n, b, V, q)
```

### Arguments

alpha	alpha for 1 - alpha confidence
n	sample size
b	vector of parameter estimates from <code>coef(object)</code>
V	covariance matrix of parameter estimates from <code>vcov(object)</code>
q	vector of coefficients

### Value

Returns a 1-row matrix. The columns are:

- Estimate - estimate of linear function
- 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

```
y <- c(43, 62, 49, 60, 36, 79, 55, 42, 67, 50)
x1 <- c(3, 6, 4, 6, 2, 7, 4, 2, 7, 5)
x2 <- c(4, 6, 3, 7, 1, 9, 3, 3, 8, 4)
out <- lm(y ~ x1 + x2)
b <- coef(out)
V <- vcov(out)
n <- length(y)
q <- c(0, .5, .5)
b
ci.lc.glm(.05, n, b, V, q)
```

```

# Should return:
# (Intercept)      x1      x2
# 26.891111  3.648889  2.213333
#
# Estimate      SE      t df      p      LL      UL
# 2.931111  0.4462518  6.56829  7 0.000313428  1.875893  3.986329

```

---

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 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 estimated group means
sd	vector of estimated group 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 - two-sided 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 distribution-free confidence interval for a linear contrast of medians in a between-subjects design using estimated medians and their standard errors. The sample median and standard error for each group can be computed using the [ci.median](#) function.

## Usage

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

## Arguments

alpha	alpha level for 1-alpha confidence
m	vector of estimated group 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.

**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
# 35.77 11.67507 12.88727 58.65273
```

---

ci.lc.prop.bs

*Confidence interval for a linear contrast of proportions in a between-subjects design*

---

**Description**

Computes an adjusted Wald confidence interval for a linear contrast of population 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 who have the 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 - two-sided 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).

**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
# 0.2119565 0.07602892 2.787841 0.005306059 0.06294259 0.3609705
```

---

ci.lc.reg

*Confidence interval for a linear contrast of regression coefficients in multiple group regression model*

---

**Description**

Computes a confidence interval and test statistic for a linear contrast of population regression coefficients (e.g., a y-intercept or a slope coefficient) 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 each within-group model
v	vector of 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
# 1.303 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 unweighted standardizer is recommended in experimental designs. The weighted standardizer is recommended in nonexperimental designs with simple random sampling. The group 1 standardizer is useful in both experimental and nonexperimental 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 estimated group means
sd	vector of estimated group standard deviation
n	vector of sample sizes
v	vector of between-subjects contrast coefficients

**Value**

Returns a 3-row matrix. The columns are:

- Estimate - estimated standardized linear contrast
- adj Estimate - bias adjusted standardized linear contrast estimate
- 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).

**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  adj Estimate      SE      LL      UL
# Unweighted standardizer: -1.301263  -1.273964  0.3692800 -2.025039 -0.5774878
# Weighted standardizer:   -1.301263  -1.273964  0.3514511 -1.990095 -0.6124317
# Group 1 standardizer:    -1.393229  -1.273810  0.4849842 -2.343781 -0.4426775
```

---

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

---

## Description

Computes confidence intervals for two types of population standardized linear contrast of means (unweighted standardizer and level 1 standardizer) in a within-subjects design. 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 estimated means for levels of within-subjects factor
sd	vector of estimated standard deviations for levels of within-subjects factor
cor	average estimated correlation of all measurement pairs
n	sample size
q	vector of within-subjects contrast coefficients

## Value

Returns a 2-row matrix. The columns are:

- Estimate - estimated standardized linear contrast
- adj Estimate - bias adjusted standardized linear contrast estimate
- 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).

**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 adj Estimate      SE      LL      UL
# Unweighted standardizer: -1.301263  -1.266557  0.3147937 -1.918248 -0.6842788
# Level 1 standardizer:   -1.393229   -1.337500  0.3661824 -2.110934 -0.6755248

```

---

ci.mad

*Confidence interval for a mean absolute deviation*


---

**Description**

Computes a confidence interval for a population mean absolute deviation from the median (MAD). The MAD is a robust alternative to the standard deviation.

**Usage**

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

**Arguments**

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

**Value**

Returns a 1-row matrix. The columns are:

- Estimate - estimated mean absolute deviation
- SE - standard error
- 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).

**Examples**

```

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

# Should return:
# Estimate      SE      LL      UL
#      12.5 2.876103 7.962667 19.62282

```

---

ci.mann

*Confidence interval for a Mann-Whitney parameter*


---

**Description**

Computes a distribution-free confidence interval for the Mann-Whitney parameter (a "common language effect size"). 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 - estimated proportion
- 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  $P(X < Y)$ , based on two independent samples." *The Indian Journal of Statistics, Series A*, **29**(1), 95–102.

### Examples

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

# Should return:
# Estimate      SE      LL UL
# 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 more robust alternative to the residual standard deviation. This function requires a vector of estimated residuals from a general linear model. This confidence interval does not assume zero excess kurtosis but does assume symmetry of the population prediction errors.

### Usage

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

### Arguments

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

### Value

Returns a 1-row matrix. The columns are:

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



**Examples**

```
res <- 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, res, 1)

# Should return:
# Estimate      SE      LL      UL
#  2.3744 0.3314752 1.751678 3.218499
```

---

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

---

**Description**

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

`ci.mean1` is deprecated and will soon be removed from `statpsych`; please switch to `ci.mean`

**Usage**

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

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

**Arguments**

alpha	alpha level for 1-alpha confidence
m	estimated mean
sd	estimated 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.mean(.05, 24.5, 3.65, 40)

# Should return:
# Estimate      SE      LL      UL
#    24.5 0.5771157 23.33267 25.66733
```

---

ci.mean.fpc

*Confidence interval for a mean with a finite population correction*


---

**Description**

Computes a confidence interval for a population mean with a finite population correction (fpc) using the estimated mean, estimated standard deviation, sample size, and population size. This function is useful when the sample size is not a small fraction of the population size.

**Usage**

```
ci.mean.fpc(alpha, m, sd, n, N)
```

**Arguments**

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

**Value**

Returns a 1-row matrix. The columns are:

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

**Examples**

```
ci.mean.fpc(.05, 24.5, 3.65, 40, 300)

# Should return:
# Estimate      SE      LL      UL
#    24.5 0.5381631 23.41146 25.58854
```

---

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 estimated means, estimated standard deviations, estimated correlation, and sample size. Also computes a paired-samples t-test. 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	estimated mean for measurement 1
m2	estimated mean for measurement 2
sd1	estimated standard deviation for measurement 1
sd2	estimated standard deviation for measurement 2
cor	estimated 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 - two-sided 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
#      6.8 1.455922 4.670578 29 6.33208e-05 3.822304 9.777696
```

---

`ci.mean2`*Confidence interval for a 2-group mean difference*

---

**Description**

Computes equal variance and unequal variance confidence intervals for a population 2-group mean difference using the estimated means, estimated standard deviations, and sample sizes. Also computes equal variance and unequal variance independent-samples t-tests. 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>	estimated mean for group 1
<code>m2</code>	estimated mean for group 2
<code>sd1</code>	estimated standard deviation for group 1
<code>sd2</code>	estimated 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 - two-sided 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	<i>Confidence interval for a median</i>
-----------	---

---

**Description**

Computes a distribution-free confidence interval for a population median.

ci.median1 is deprecated and will soon be removed from statspsych; please switch to ci.median

**Usage**

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

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

**Arguments**

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

**Value**

Returns a 1-row matrix. The columns are:

- Estimate - 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.median(.05, y)

# Should return:
# Estimate      SE LL UL
#      20 4.270922 10 30

```

---

ci.median.ps

*Confidence interval for a paired-samples median difference*


---

**Description**

Computes a distribution-free confidence interval for a difference of population medians in a paired-samples design. This function also computes the standard error of each median and the covariance between the two estimated 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 (paired with y1)

**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 the two estimated 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.

## 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
#      13      30          -17 3.362289 -23.58996 -10.41004
#      SE1      SE2      COV
# 3.085608 4.509735 9.276849
```

---

ci.median2

*Confidence interval for a 2-group median difference*

---

## Description

Computes a distribution-free confidence interval for a difference of population medians in a 2-group design.

## 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 for group 1
- Median2 - estimated median for group 2
- Median1-Median2 - estimated difference 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.

## 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
#   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 odds ratio
- 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.oddsratio(.05, 229, 28, 96, 24)

# Should return:
#   Estimate      SE      LL      UL
# [1,] 2.044451  0.6154578  1.133267  3.688254
```

---

ci.pairs.mult	<i>Confidence intervals for pairwise proportion differences of a multinomial variable</i>
---------------	---

---

## Description

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

## Usage

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

## Arguments

alpha	alpha level for 1-alpha confidence
f	vector of multinomial frequency counts

## Value

Returns a matrix with the number of rows equal to the number of pairwise comparisons. 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://doi.org/10.3102/1076998611411915).

**Examples**

```
f <- c(125, 82, 92)
ci.pairs.mult(.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.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 population proportions in a between-subjects design using 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 who have the attribute
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 - adjusted estimate of proportion difference
- SE - adjusted standard error
- z - z test statistic
- p - two-sided 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), 280-288. ISSN 00031305, doi:10.2307/2685779.

**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
```

---

ci.pbcor

*Confidence intervals for point-biserial correlations*


---

**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 (but not 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	estimated mean for group 1
m2	estimated mean for group 2
sd1	estimated standard deviation for group 1
sd2	estimated 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
- 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.

## Examples

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

# Should return:
#           Estimate      SE      LL      UL
# Weighted:  0.7065799 0.04890959 0.5885458 0.7854471
# Unweighted: 0.7020871 0.05018596 0.5808366 0.7828948
```

---

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

---

## Description

Computes a Fisher confidence interval for a population phi correlation. This function requires the frequency counts from a 2 x 2 contingency table for two dichotomous variables. This measure of association is usually most appropriate when 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 correlation
- 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
# 0.1229976 0.05477117 0.01462398 0.2285149
```

---

ci.poisson	<i>Confidence interval for a Poisson rate</i>
------------	---

---

**Description**

Computes a confidence interval for a population Poisson rate. This function requires the number of occurrences (f) of a specific event that were observed over a specific period of time (t).

**Usage**

```
ci.poisson(alpha, f, t)
```

**Arguments**

alpha	alpha value for 1-alpha confidence
f	number of event occurrences
t	time period

**Details**

The time period (t) does not need to be an integer and can be expressed in any unit of time such as seconds, hours, or months. The occurrences are assumed to be independent of one another and the unknown occurrence rate is assumed to be constant over time.

**Value**

Returns a 1-row matrix. The columns are:

- Estimate - estimated Poisson rate
- SE - recovered standard error
- LL - lower limit of the confidence interval
- UL - upper limit of the confidence interval

## References

Hahn GJ, Meeker WQ (1991). *Statistical Intervals: A Guide for Practitioners*. Wiley. ISBN 9780470316771, doi:10.1002/9780470316771, <http://dx.doi.org/10.1002/9780470316771>.

## Examples

```
ci.poisson(.05, 23, 5.25)

# Should return:
# Estimate      SE      LL      UL
# 4.380952 0.9684952 2.777148 6.57358
```

---

ci.popsiz	<i>Confidence interval for an unknown population size</i>
-----------	---

---

## 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. An approximate standard error is recovered from the confidence interval.

## 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
- SE - recovered standard error
- LL - lower limit of the confidence interval
- UL - upper limit of the confidence interval

**Examples**

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

# Should return:
# Estimate      SE  LL  UL
#    2908 49.49071 2818 3012
```

---

ci.prop

*Confidence intervals for a proportion*

---

**Description**

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

ci.prop1 is deprecated and will soon be removed from statspsych; please switch to ci.prop

**Usage**

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

```
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.

## Examples

```
ci.prop(.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
```

---

ci.prop.fpc	<i>Confidence interval for a proportion with a finite population correction</i>
-------------	---

---

## Description

Computes an adjusted Wald interval for a population proportion with a finite population correction (fpc). This confidence interval is useful when the sample size is not a small fraction of the population size.

## Usage

```
ci.prop.fpc(alpha, f, n, N)
```

## Arguments

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

## Value

Returns a 1-row matrix. The columns are:

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



## Examples

```
ci.prop.fpc(.05, 12, 100, 400)

# Should return:
#   Estimate      SE      LL      UL
# 0.1346154 0.0290208 0.07773565 0.1914951
```

---

ci.prop.inv

*Confidence interval for a proportion using inverse sampling*

---

## Description

Computes an exact confidence interval for a population proportion when inverse sampling has been used. An approximate standard error is recovered from the confidence interval. With inverse sampling, the number of participants who have the attribute (f) is predetermined and sampling continues until f attains its prespecified value. With inverse sampling, the sample size (n) will not be known in advance.

## Usage

```
ci.prop.inv(alpha, f, n)
```

## Arguments

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

## Value

Returns a 1-row matrix. The columns are:

- Estimate - estimate of proportion
- SE - recovered standard error
- LL - lower limit of confidence interval
- UL - upper limit of confidence interval

## References

Zou GY (2010). "Confidence interval estimation under inverse sampling." *Computational Statistics & Data Analysis*, **54**(1), 55–64. ISSN 0167-9473, doi:[10.1016/j.csda.2005.05.007](https://doi.org/10.1016/j.csda.2005.05.007).

**Examples**

```
ci.prop.inv(.05, 5, 67)

# Should return:
# Estimate      SE      LL      UL
# 0.07462687 0.03145284 0.02467471 0.1479676
```

---

ci.prop.ps

*Confidence interval for a paired-samples proportion difference*


---

**Description**

Computes an adjusted Wald confidence interval for a difference of population 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://doi.org/10.3102/1076998611411915).

**Examples**

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

# Should return:
# Estimate      SE      LL      UL
#      0.44 0.09448809 0.2548067 0.6251933
```

---

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

---

**Description**

Computes an adjusted Wald confidence interval for a population 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 for group 1
n2	sample size for 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), 280-288. ISSN 00031305, doi:[10.2307/2685779](https://doi.org/10.2307/2685779).

**Examples**

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

# Should return:
# Estimate      SE      LL      UL
# 0.09210526 0.04476077 0.004375769 0.1798348
```

---

ci.prop2.inv	<i>Confidence interval for a 2-group proportion difference using inverse sampling</i>
--------------	---

---

**Description**

Computes an approximate confidence interval for a population proportion difference when inverse sampling has been used. An approximate standard error is recovered from the confidence interval. With inverse sampling, the number of participants who have the attribute within group 1 (f1) and group 2 (f2) are predetermined, and sampling continues within each group until f1 and f2 attain their prespecified values. With inverse sampling, the sample sizes (n1 and n2) will not be known in advance.

**Usage**

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

**Arguments**

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

**Value**

Returns a 1-row matrix. The columns are:

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

**References**

Zou GY (2010). "Confidence interval estimation under inverse sampling." *Computational Statistics & Data Analysis*, **54**(1), 55–64. ISSN 0167-9473, doi:[10.1016/j.csda.2005.05.007](https://doi.org/10.1016/j.csda.2005.05.007).

**Examples**

```
ci.prop2.inv(.05, 10, 10, 48, 213)

# Should return:
# Estimate      SE      LL      UL
# 0.161385 0.05997618 0.05288277 0.2879851
```

---

ci.pv	<i>Confidence intervals for positive and negative predictive values with retrospective sampling</i>
-------	---

---

**Description**

Computes adjusted Wald confidence intervals for positive and negative predictive values (PPV and NPV) of a diagnostic test with retrospective sampling where the population prevalence rate is assumed to be known. With retrospective sampling, one random sample is obtained from a subpopulation that is known to have a "positive" outcome, a second random sample is obtained from a subpopulation that is known to have a "negative" outcome, and then the diagnostic test (scored "pass" or "fail") is given in each sample. PPV and NPV can be expressed as a function of proportion ratios and the known population prevalence rate (the population proportion who would "pass"). The confidence intervals for PPV and NPV are based on the Price-Bonett adjusted Wald confidence interval for a proportion ratio.

**Usage**

```
ci.pv(alpha, f1, f2, n1, n2, prev)
```

**Arguments**

alpha	alpha level for 1-alpha confidence
f1	number of participants with a positive outcome who pass the test
f2	number of participants with a negative outcome who fail the test
n1	sample size for the positive outcome group
n2	sample size for the negative outcome group
prev	known population proportion with a positive outcome

**Value**

Returns a 2-row matrix. The columns are:

- Estimate - adjusted estimate of the predictive value
- 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.

## Examples

```
ci.pv(.05, 89, 5, 100, 100, .16)

# Should return:
#      Estimate      LL      UL
# PPV: 0.7640449 0.5838940 0.8819671
# NPV: 0.9779978 0.9623406 0.9872318
```

---

ci.random.anova	<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.anova(alpha, m, sd, n)
```

## Arguments

alpha	1 - alpha confidence
m	vector of estimated group means
sd	vector of estimated group standard deviations
n	common sample size in each 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.anova(.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.cod2	<i>Confidence interval for a ratio of dispersion coefficients in a 2-group design</i>
---------------	---

---

### Description

Computes a confidence interval for a ratio of population dispersion coefficients (mean absolute deviation from median divided by median) in a 2-group design. Ratio-scale scores are assumed.

### Usage

```
ci.ratio.cod2(alpha, y1, y2)
```

### Arguments

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

### Value

Returns a 1-row matrix. The columns are:

- COD1 - estimated coefficient of dispersion in group 1
- COD2 - estimated coefficient of dispersion in group 2
- COD1/COD2 - estimated ratio of dispersion coefficients
- LL - lower limit of the confidence interval
- UL - upper limit of the confidence interval

**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.cod2(.05, y1, y2)

# Should return:
#   COD1      COD2 COD1/COD2      LL      UL
# 0.1333333 0.1232558 1.081761 0.494964 2.282254

```

---

ci.ratio.cv2	<i>Confidence interval for a ratio of coefficients of variation in a 2-group design</i>
--------------	---

---

**Description**

Computes a confidence interval for a ratio of population coefficients of variation (CV) in a 2-group design. This confidence interval uses the confidence interval for each CV and then uses the MOVER-DL method (see Newcombe, page 138) to obtain a confidence interval for CV1/CV2. The CV assumes ratio-scale scores.

**Usage**

```
ci.ratio.cv2(alpha, m1, m2, sd1, sd2, n1, n2)
```

**Arguments**

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

**Value**

Returns a 1-row matrix. The columns are:

- Estimate - estimated ratio of coefficients of variation
- LL - lower limit of the confidence interval
- UL - upper limit of the confidence interval



## References

Newcombe RG (2013). *Confidence Interval for Proportions and Related Measures of Effect Size*. CRC Press.

## Examples

```
ci.ratio.cv2(.05, 34.5, 26.1, 4.15, 2.26, 50, 50)

# Should return:
# Estimate      LL      UL
# 1.389188 1.041478 1.854101
```

---

ci.ratio.mad.ps	<i>Confidence interval for a paired-sample MAD ratio</i>
-----------------	--

---

## 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 (paired with y1)

## 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).

### 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
# 12.71429  7.5  1.695238 1.109176 2.590961
```

---

ci.ratio.mad2	<i>Confidence interval for a 2-group ratio of mean absolute deviations</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 for group 1
- MAD2 - estimated MAD for 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).

**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
# 5.111111 5.888889 0.8679245 0.4520879 1.666253

```

---

ci.ratio.mape2	<i>Confidence interval for a ratio of mean absolute prediction errors in a 2-group design</i>
----------------	---

---

**Description**

Computes a confidence interval for a ratio of population mean absolute prediction errors from a general linear model in two independent groups. The number of predictor variables can differ across groups and the two models can be non-nested. This function requires a vector of estimated residuals from each group. This function does not assume zero excess kurtosis but does assume symmetry in the population prediction errors for the two models.

**Usage**

```
ci.ratio.mape2(alpha, res1, res2, s1, s2)
```

**Arguments**

alpha	alpha level for 1-alpha confidence
res1	vector of residuals from group 1
res2	vector of residuals from group 2
s1	number of predictor variables used in group 1
s2	number of predictor variables used in group 2

**Value**

Returns a 1-row matrix. The columns are:

- MAPE1 - bias adjusted mean absolute prediction error for group 1
- MAPE2 - bias adjusted mean absolute prediction error for group 2
- MAPE1/MAPE2 - ratio of bias adjusted mean absolute prediction errors
- LL - lower limit of the confidence interval
- UL - upper limit of the confidence interval

**Examples**

```

res1 <- 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)
res2 <- c(-0.71, -0.89, 0.72, -0.35, 0.33 -0.92, 2.37, 0.51, 0.68, -0.85,
         -0.15, 0.77, -1.52, 0.89, -0.29, -0.23, -0.94, 0.93, -0.31 -0.04)
ci.ratio.mape2(.05, res1, res2, 1, 1)

# Should return:
#   MAPE1      MAPE2 MAPE1/MAPE2      LL      UL
# 2.58087 0.8327273   3.099298 1.917003 5.010761

```

---

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 (paired with y1)

**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. 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
# 3.4875 3.075  1.134146 1.09417 1.175583

```

---

ci.ratio.mean2	<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**

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:

- Mean1 - estimated mean for group 1
- Mean2 - estimated mean for 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. 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
# 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 distribution-free confidence interval for a ratio of population medians in a paired-samples design. Ratio-scale measurements are assumed.

**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 (paired with y1)

**Value**

Returns a 1-row matrix. The columns are:

- Median1 - estimated median for measurement 1
- Median2 - estimated median for 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. 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
# 13 30 0.4333333 0.3094838 0.6067451

```

---

ci.ratio.median2      *Confidence interval for a 2-group median ratio*

---

**Description**

Computes a distribution-free 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**

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 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. 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
#      43      37      1.162162 0.927667 1.455933

```

---

ci.ratio.poisson2      *Confidence interval for a ratio of Poisson rates in a 2-group design*

---

**Description**

Computes a confidence interval for a ratio of population Poisson rates in a 2-group design. The confidence interval is based on the binomial method with an Agresti-Coull confidence interval. This function requires the number of occurrences of a specific event (f) that were observed over a specific period of time (t) within each group.

**Usage**

```
ci.ratio.poisson2(alpha, f1, f2, t1, t2)
```

**Arguments**

alpha	alpha value for 1-alpha confidence
f1	number of event occurrences for group 1
f2	number of event occurrences for group 2
t1	time period for group 1
t2	time period for group 2

**Details**

The time periods do not need to be integers and can be expressed in any unit of time such as seconds, hours, or months. The occurrences are assumed to be independent of one another and the unknown occurrence rate is assumed to be constant over time within each group condition.

**Value**

Returns a 1-row matrix. The columns are:

- Estimate - estimated ratio of Poisson rates
- LL - lower limit of the confidence interval
- UL - upper limit of the confidence interval



## References

Price RM, Bonett DG (2000). "Estimating the ratio of two Poisson rates." *Computational Statistics & Data Analysis*, **34**(3), 345–356. doi:10.1016/S01679473(99)001000.

## Examples

```
ci.ratio.poisson2(.05, 19, 5, 30, 40.5)

# Should return:
# Estimate      LL      UL
#    5.13 1.939576 13.71481
```

---

ci.ratio.prop.ps	<i>Confidence interval for a paired-samples proportion ratio</i>
------------------	--

---

## Description

Computes a confidence interval for a ratio of population 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, 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

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.

**Examples**

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

# Should return:
# Estimate      LL      UL
#      3.2 1.766544 5.796628
```

---

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

---

**Description**

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

**Usage**

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

**Arguments**

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

**Value**

Returns a 1-row matrix. The columns are:

- Estimate - adjusted estimate 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://doi.org/10.1002/sim.3376).

**Examples**

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

# Should return:
# Estimate      LL      UL
# 1.666667 1.017253 2.705025
```

---

ci.ratio.sd2

*Confidence interval for a 2-group ratio of standard deviations*


---

**Description**

Computes a robust confidence interval for a ratio of population standard deviations in a 2-group design. This function is a modification of the confidence interval proposed by Bonett (2006). The original Bonett method used a pooled kurtosis estimate in the standard error that assumed equal variances, which limited the confidence interval's use to tests of equal population variances and equivalence tests. This function uses a pooled kurtosis estimate that does not assume equal variances and provides a useful confidence interval for a ratio of standard deviations under general conditions. This function requires of minimum sample size of four per group but sample sizes of at least 10 per group are recommended.

**Usage**

```
ci.ratio.sd2(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:

- SD1 - estimated SD for group 1
- SD2 - estimated SD for group 2
- SD1/SD2 - estimate of SD ratio
- LL - lower limit of the confidence interval
- UL - upper limit of the confidence interval

**References**

Bonett DG (2006). "Robust confidence interval for a ratio of standard deviations." *Applied Psychological Measurement*, **30**(5), 432–439. ISSN 0146-6216, doi:10.1177/0146621605279551.

**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.sd2(.05, y1, y2)

# Should return:
#   SD1   SD2  SD1/SD2   LL   UL
# 5.711587 6.450667 0.8854257 0.486279 1.728396

```

---

ci.rel2

*Confidence interval for a 2-group reliability difference*


---

**Description**

Computes a 100(1 - alpha)% confidence interval for a difference in population reliabilities in a 2-group design. This function can be used with any type of reliability coefficient (e.g., Cronbach alpha, McDonald omega, intraclass reliability). The function requires a point estimate and a 100(1 - alpha)% confidence interval for each reliability as input.

**Usage**

```
ci.rel2(rel1, ll1, ul1, rel2, ll2, ul2)
```

**Arguments**

rel1	estimated reliability for group 1
ll1	lower limit for group 1 reliability
ul1	upper limit for group 1 reliability
rel2	estimated reliability for group 2
ll2	lower limit for group 2 reliability
ul2	upper limit for group 2 reliability

**Value**

Returns a 1-row matrix. The columns are:

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

**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).

**Examples**

```
ci.rel2(.4, .35, .47, .2, .1, .32)

# Should return:
# Estimate  LL      UL
#      0.2 0.07 0.3220656
```

---

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

---

**Description**

Computes a confidence interval for a population reliability coefficient such as Cronbach's alpha or McDonald's omega using an estimate of the reliability and its standard error. The standard error can be a robust standard error or bootstrap standard error obtained from an SEM program.

**Usage**

```
ci.reliability(alpha, rel, se, n)
```

**Arguments**

alpha	alpha level for 1-alpha confidence
rel	estimated reliability
se	standard error of reliability
n	sample size

**Value**

Returns a 1-row matrix. The columns are:

- Estimate - estimated reliability (from input)
- LL - lower limit of the confidence interval
- UL - upper limit of the confidence interval

**Examples**

```
ci.reliability(.05, .88, .0147, 100)

# Should return:
# Estimate      LL      UL
#      0.88 0.8489612 0.9065575
```

---

 ci.rsqr

*Confidence interval for squared multiple correlation*


---

### Description

Computes an approximate confidence interval for a population squared multiple correlation in a linear model with random predictor variables. This function uses the scaled central F approximation method. An approximate standard error is recovered from the confidence interval.

### Usage

```
ci.rsqr(alpha, r2, s, n)
```

### Arguments

alpha	alpha value for 1-alpha confidence
r2	estimated unadjusted squared multiple correlation
s	number of predictor variables
n	sample size

### Value

Returns a 1-row matrix. The columns are:

- R-squared - estimate of unadjusted R-squared (from input)
- adj R-squared - bias adjusted R-squared estimate
- SE - recovered standard error
- LL - lower limit of the confidence interval
- UL - upper limit of the confidence interval

### References

Helland IS (1987). "On the interpretation and use of R2 in regression analysis." *Biometrics*, **43**(1), 61–69. doi:[10.2307/2531949](https://doi.org/10.2307/2531949).

### Examples

```
ci.rsqr(.05, .241, 3, 116)

# Should return:
# R-squared adj R-squared      SE      LL      UL
#   0.241      0.2206696 0.06752263 0.09819599 0.3628798
```

---

`ci.sign`*Confidence interval for the parameter of the one-sample sign test*

---

**Description**

Computes adjusted Wald interval for the population proportion of quantitative scores that are greater than the null hypothesis value of the population median in a one-sample sign test.

**Usage**

```
ci.sign(alpha, y, h)
```

**Arguments**

alpha	alpha level for 1-alpha confidence
y	vector of y scores
h	null hypothesis value for population median

**Value**

Returns a 1-row matrix. The columns are:

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

**References**

Agresti, A, & Coull, BA (1998) Approximate is better than “exact” for interval estimation of binomial proportions. *American Statistician*, 52, 119–126. doi: 10.1080/00031305.1998.10480550

**Examples**

```
y <- c(30, 20, 15, 10, 10, 60, 20, 25, 20, 30, 10, 5, 50, 40, 20, 10,
      0, 20, 50)
ci.sign(.05, y, 9)

# Should return:
# Estimate      SE      LL      UL
# 0.826087 0.0790342 0.6711828 0.9809911
```

---

ci.slope.mean.bs	<i>Confidence interval for the slope of means in a one-factor experimental design with a quantitative between-subjects factor</i>
------------------	---

---

### Description

Computes a test statistic and confidence interval for the slope of means in a one-factor experimental design with a quantitative between-subjects factor. This function computes both the unequal variance 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.slope.mean.bs(alpha, m, sd, n, x)
```

### Arguments

alpha	alpha level for 1-alpha confidence
m	vector of sample means
sd	vector of sample standard deviations
n	vector of sample sizes
x	vector of numeric predictor variable values

### Value

Returns a 2-row matrix. The columns are:

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

### 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)
x <- c(5, 10, 20, 30)
ci.slope.mean.bs(.05, m, sd, n, x)
```

```
# Should return:
```

```
#           Estimate      SE      t      df
```



```

# Equal Variances Assumed:    0.3664407 0.06770529 5.412290 36.00000
# Equal Variances Not Assumed: 0.3664407 0.07336289 4.994905 18.65826
#                               p          LL          UL
# Equal Variances Assumed:    4.242080e-06 0.2291280 0.5037534
# Equal Variances Not Assumed: 8.468223e-05 0.2126998 0.5201815

```

---

ci.slope.prop.bs	<i>Confidence interval for a slope of a proportion in a single-factor experimental design with a quantitative between-subjects factor</i>
------------------	---

---

### Description

Computes a test statistic and an adjusted Wald confidence interval for the population slope of proportions in a one-factor experimental design with a quantitative between-subjects factor.

### Usage

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

### Arguments

alpha	alpha level for 1-alpha confidence
f	vector of frequency counts of participants who have the 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 - two-sided 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).

**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
# 0.007542293 0.002016793 3.739746 0.000184206 0.003589452 0.01149513
```

---

ci.spcor

---

*Confidence interval for a semipartial correlation*


---

**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	estimated semipartial correlation
r2	estimated squared multiple correlation in full model
n	sample size

**Value**

Returns a 1-row matrix. The columns are:

- Estimate - estimated semipartial correlation (from input)
- 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).

**Examples**

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

# Should return:
# Estimate      SE      LL      UL
# 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 (paired with y)

**Value**

Returns a 1-row matrix. The columns are:

- Estimate - estimated Spearman 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).

**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
# 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 of 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	estimated Spearman correlation for group 1
cor2	estimated Spearman correlation for group 2
n1	sample size for group 1
n2	sample size for group 2

**Value**

Returns a 1-row matrix. The columns are:

- Estimate - estimated correlation difference
- 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).

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).

**Examples**

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

# Should return:
# Estimate      SE      LL      UL
#      0.06 0.08124926 -0.1003977 0.2185085
```

---

ci.stdmean	<i>Confidence interval for a 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 (see ci.cv).

ci.stdmean1 is deprecated and will soon be removed from statspsych; please switch to ci.stdmean

**Usage**

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

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

**Arguments**

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

**Value**

Returns a 1-row matrix. The columns are:

- Estimate - estimated standardized mean difference
- adj Estimate - bias adjusted standardized mean difference estimate
- 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).

## Examples

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

# Should return:
# Estimate  adj Estimate      SE      LL      UL
# 1.232877    1.209015  0.2124335  0.8165146  1.649239
```

---

ci.stdmean.ps	<i>Confidence intervals 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	estimated mean for measurement 1
m2	estimated mean for measurement 2
sd1	estimated standard deviation for measurement 1
sd2	estimated standard deviation for measurement 2
cor	estimated correlation between measurements
n	sample size

## Value

Returns a 3-row matrix. The columns are:

- Estimate - estimated standardized mean difference
- adj Estimate - bias adjusted standardized mean difference estimate
- 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.

## Examples

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

# Should return:
#
#           Estimate  adj Estimate      SE      LL      UL
# Unweighted standardizer:  0.5550319    0.5433457 0.1609934 0.2394905 0.8705732
# Measurement 1 standardizer: 0.5424837    0.5253526 0.1615500 0.2258515 0.8591158
# Measurement 2 standardizer: 0.5684932    0.5505407 0.1692955 0.2366800 0.9003063
```

---

ci.stdmean.strat	<i>Confidence intervals for a 2-group standardized mean difference with stratified sampling</i>
------------------	---

---

## Description

Computes confidence intervals for a population standardized mean difference in a 2-group non-experimental design with stratified random sampling (a random sample of a specified size from each subpopulation) 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	estimated mean for group 1
m2	estimated mean for group 2
sd1	estimated standard deviation for group 1
sd2	estimated standard deviation for group 2
n1	sample size for group 1
n2	sample size for group 2
p1	proportion of total population in subpopulation 1

**Value**

Returns a 3-row matrix. The columns are:

- Estimate - estimated standardized mean difference
- adj Estimate - bias adjusted standardized mean difference estimate
- 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.

**Examples**

```
ci.stdmean.strat(.05, 33.2, 30.8, 10.5, 11.2, 200, 200, .533)

# Should return:
#               Estimate  adj Estimate      SE      LL      UL
# Weighted standardizer: 0.2215549    0.2211371 0.10052057 0.02453817 0.4185716
# Group 1 standardizer: 0.2285714    0.2277089 0.10427785 0.02419059 0.4329523
# Group 2 standardizer: 0.2142857    0.2277089 0.09776049 0.02267868 0.4058927
```

---

ci.stdmean2

*Confidence intervals for a 2-group standardized mean difference*

---

**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	estimated mean for group 1
m2	estimated mean for group 2
sd1	estimated standard deviation for group 1
sd2	estimated standard deviation for group 2
n1	sample size for group 1
n2	sample size for group 2

**Value**

Returns a 4-row matrix. The columns are:

- Estimate - estimated standardized mean difference
- adj Estimate - bias adjusted standardized mean difference estimate
- 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).

**Examples**

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

```
# Should return:
#
# Unweighted standardizer: Estimate adj Estimate SE LL UL
# Weighted standardizer: 1.174493 1.159240 0.2844012 0.6170771 1.731909
# Group 1 standardizer: 1.174493 1.159240 0.2802826 0.6251494 1.723837
# Group 1 standardizer: 1.147541 1.117605 0.2975582 0.5643375 1.730744
# Group 2 standardizer: 1.203438 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. An approximate standard error is recovered from the confidence interval.

**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
- SE - recovered standard error
- 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://doi.org/10.3102/10769986030002213).

**Examples**

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

# Should return:
# Estimate      SE      LL      UL
# 0.5135167 0.09301703 0.3102345 0.6748546
```

---

ci.theil	<i>Theil-Sen estimate and confidence interval for slope</i>
----------	---

---

### Description

Computes a Theil-Sen estimate and distribution-free confidence interval for the population slope in a simple linear regression model. An approximate standard error is recovered from the confidence interval.

### Usage

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

### Arguments

alpha	alpha level for 1-alpha confidence
y	vector of response variable scores
x	vector of predictor variable scores (paired with y)

### Value

Returns a 1-row matrix. The columns are:

- Estimate - Theil-Sen estimate of population slope
- SE - recovered standard error
- LL - lower limit of confidence interval
- UL - upper limit of confidence interval

### References

Hollander M, Wolf DA (1999). *Nonparametric Statistical Methods*. Wiley.

### 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.theil(.05, y, x)

# Should return:
# Estimate      SE      LL      UL
#      0.5 0.1085927 0.3243243 0.75
```

---

`ci.tukey`*Tukey-Kramer confidence intervals for all pairwise mean differences in a between-subjects design*

---

**Description**

Computes heteroscedastic Tukey-Kramer (also known as Games-Howell) confidence intervals for all pairwise comparisons of population means using estimated means, estimated 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 estimated group means
sd	vector of estimated group 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 - two-sided 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).

**Examples**

```

m <- c(12.86, 17.57, 26.29, 30.21)
sd <- c(13.185, 12.995, 14.773, 15.145)
n <- c(20, 20, 20, 20)
ci.tukey(.05, m, sd, n)

# Should return:
#   Estimate      SE          t      df          p          LL          UL
# 1 2      -4.71 4.139530 -1.1378102 37.99200 0.668806358 -15.83085  6.4108517
# 1 3     -13.43 4.427673 -3.0331960 37.51894 0.021765570 -25.33172 -1.5282764
# 1 4     -17.35 4.490074 -3.8640790 37.29278 0.002333937 -29.42281 -5.2771918
# 2 3      -8.72 4.399497 -1.9820446 37.39179 0.212906199 -20.54783  3.1078269
# 2 4     -12.64 4.462292 -2.8326248 37.14275 0.035716267 -24.64034 -0.6396589
# 3 4      -3.92 4.730817 -0.8286096 37.97652 0.840551420 -16.62958  8.7895768

```

---

ci.var.upper	<i>Upper confidence limit of a variance</i>
--------------	---

---

**Description**

Computes an upper confidence limit for a population variance using an estimated 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 for desired power that require a planning value of the population variance.

**Usage**

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

**Arguments**

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

**Value**

Returns an upper limit (UL) variance planning value

**Examples**

```

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

# Should return:
#           UL
# 17.23264

```

---

`ci.yule`*Confidence intervals for generalized Yule coefficients*

---

### Description

Computes confidence intervals for four generalized Yule measures of association (Yule Q, Yule Y, Digby H, and Bonett-Price Y\*) 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. Digby H is sometimes used as a crude approximation to the tetrachoric correlation. Yule Y is equal to the phi coefficient only when all marginal frequencies are equal. Bonett-Price Y\* is a better approximation to the phi coefficient when the marginal frequencies are not equal.

### 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 generalized Yule coefficient
- SE - standard error
- LL - lower limit of the confidence interval
- UL - upper limit of the confidence interval

### References

Bonett DG, Price RM (2007). "Statistical inference for generalized Yule coefficients in 2x2 contingency tables." *Sociological Methods & Research*, **35**(3), 429–446. ISSN 0049-1241, doi:[10.1177/0049124106292358](https://doi.org/10.1177/0049124106292358).

**Examples**

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

# Should return:
#   Estimate      SE      LL      UL
# Q: 0.3430670 0.13280379 0.06247099 0.5734020
# Y: 0.1769015 0.07290438 0.03126603 0.3151817
# H: 0.2619244 0.10514465 0.04687994 0.4537659
# Y*: 0.1311480 0.05457236 0.02307188 0.2361941
```

---

etasqr.adj

*Bias adjustments for an eta-squared estimate*

---

**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:
# adj Eta-squared
#   0.282381
```

---

`etasqr.gen.2way`*Generalized eta-squared estimates in a two-factor design*

---

### Description

Computes generalized eta-square estimates in a two-factor design where one or both factors are classification factors. If both factors are treatment factors, then partial eta-square estimates are typically recommended. The eta-squared estimates from this function can be used in the [etasqr.adj](#) function to obtain bias adjusted estimates.

### Usage

```
etasqr.gen.2way(SSa, SSb, SSab, SSe)
```

### Arguments

SSa	sum of squares for factor A
SSb	sum of squares for factor B
SSab	sum of squares for A x B interaction
SSe	error (within) sum of squares

### Value

Returns a 3-row matrix. The columns are:

- A - estimate of eta-squared for factor A
- B - estimate of eta-squared for factor B
- AB - estimate of eta-squared for A x B interaction

### Examples

```
etasqr.gen.2way(12.3, 15.6, 5.2, 7.9)

# Should return:
#
#           A           B           AB
# A treatment, B classification:  0.300000  0.5435540  0.1811847
# A classification, B treatment:  0.484252  0.3804878  0.2047244
# A classification, B classification: 0.300000  0.3804878  0.1268293
```



---

fitindices	<i>SEM fit indices</i>
------------	------------------------

---

**Description**

Computes the normed fit index (NFI), adjusted normed fit index (adj NFI), comparative fit index (CFI), Tucker-Lewis fit index (TLI), and root mean square error of approximation index (RMSEA). Of the first four indices, the adj NFI index is recommended because it has smaller sampling variability than CFI and TLI and less negative bias than NFI.

**Usage**

```
fitindices(chi1, df1, chi2, df2, n)
```

**Arguments**

chi1	chi-square test statistic for full model
df1	degrees of freedom for full model
chi2	chi-square test statistic for reduced model
df2	degrees of freedom for reduced model
n	sample size

**Value**

Returns NFI, adj NFI, CFI, TLI, and RMSEA

**Examples**

```
fitindices(14.21, 10, 258.43, 20, 300)

# Should return:
#      NFI  adj NFI      CFI      TLI      RMSEA
# 0.9450141 0.9837093 0.9823428 0.9646857 0.03746109
```

---

iqv	<i>Indices of qualitative variation</i>
-----	---

---

**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 indices

**Examples**

```
f <- c(10, 46, 15, 3)
iqv(f)

# Should return:
# Simpson Berger Shannon
# 0.7367908 0.5045045 0.7
```

---

pi.cor

*Prediction limit for an estimated correlation*

---

**Description**

Computes approximate prediction interval for the estimated Pearson correlation in a future study with a planned sample size of n. The prediction interval uses a correlation estimate from a prior study that had a sample size of n0.

**Usage**

```
pi.cor(alpha, cor, n0, n)
```

**Arguments**

alpha                alpha value for 1-alpha confidence  
cor                   estimated Pearson correlation from prior study  
n0                    sample size used to estimate correlation in prior study  
n                     planned sample size of future study

**Value**

Returns a prediction interval of an estimated correlation in a future study

**Examples**

```
pi.cor(.1, .761, 50, 100)

# Should return:
#      LL      UL
# 0.6034092 0.8573224
```

---

<i>pi.prop</i>	<i>Prediction interval for an estimated proportion</i>
----------------	--

---

**Description**

Computes approximate prediction interval for the estimated proportion in a future study with a planned sample size of  $n$ . The prediction interval uses a proportion estimate from a prior study that had a sample size of  $n_0$ .

**Usage**

```
pi.prop(alpha, prop, n0, n)
```

**Arguments**

<code>alpha</code>	alpha value for 1-alpha confidence
<code>prop</code>	estimated proportion from prior study
<code>n0</code>	sample size used to estimate proportion in prior study
<code>n</code>	planned sample size of future study

**Value**

Returns a prediction interval for an estimated proportion in a future study

**Examples**

```
pi.prop(.1, .225, 80, 120)

# Should return:
#      LL      UL
# 0.1390955 0.337095
```

---

pi.score	<i>Prediction interval for one score</i>
----------	--

---

**Description**

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

**Usage**

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

**Arguments**

alpha	alpha level for 1-alpha confidence
m	estimated mean
sd	estimated 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.score(.05, 24.5, 3.65, 40)

# Should return:
# Predicted df      LL      UL
#      24.5  39 17.02546 31.97454
```

---

pi.score.ps	<i>Prediction interval for difference of scores in a 2-level within-subjects experiment</i>
-------------	---

---

### Description

For a 2-level within-subjects experiment, 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.

### Usage

```
pi.score.ps(alpha, m1, m2, sd1, sd2, cor, n)
```

### Arguments

alpha	alpha level for 1-alpha confidence
m1	estimated mean for measurement 1
m2	estimated mean for measurement 2
sd1	estimated standard deviation for measurement 1
sd2	estimated standard deviation for measurement 2
cor	estimated correlation of paired scores
n	sample size

### Value

Returns a 1-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

### Examples

```
pi.score.ps(.05, 265.1, 208.6, 23.51, 19.94, .814, 30)

# Should return:
# Predicted df      LL      UL
#      56.5 29 28.05936 84.94064
```

---

`pi.score2`*Prediction interval for a difference of scores in a 2-group experiment*

---

### 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

<code>alpha</code>	alpha level for 1-alpha confidence
<code>m1</code>	estimated mean for group 1
<code>m2</code>	estimated mean for group 2
<code>sd1</code>	estimated standard deviation for group 1
<code>sd2</code>	estimated 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:

- 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).

**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
```

---

pi.var.upper                      *Upper prediction limit for an estimated variance*

---

**Description**

Computes an upper prediction limit for the estimated variance in a future study for a planned sample size. The prediction limit uses a variance estimate from a prior study. Several confidence interval sample size functions in this package require a planning value of the estimated variance that is expected in the planned study. The upper variance prediction limit is useful as a variance planning value for the sample size required to obtain a confidence interval with desired width. This strategy for specifying a variance planning value is useful in applications where the population variance in the prior study is assumed to be very similar to the population variance in the planned study.

**Usage**

```
pi.var.upper(alpha, var, n0, n)
```

**Arguments**

alpha	alpha value for upper 1-alpha confidence
var	estimated variance from prior study
n0	sample size used to estimate variance
n	planned sample size of future study

**Value**

Returns an upper prediction estimate (UL) of an estimated variance in a future study

**References**

Hahn GJ (1972). "Simultaneous prediction intervals to contain the standard deviations or ranges of future samples from a normal population." *Journal of the American Statistical Association*, **67**(340), 938–942. doi:10.1080/01621459.1972.10481322.

**Examples**

```
pi.var.upper(.05, 15, 40, 100)

# Should return:
#      UL
# 23.9724
```

---

power.cor

*Approximates the power of a correlation test for a planned sample size*

---

**Description**

Computes the approximate power of a test for a population Pearson or partial correlation test for a planned sample size. Set  $s = 0$  for a Pearson correlation.

**Usage**

```
power.cor(alpha, n, cor, h, s)
```

**Arguments**

alpha	alpha level for hypothesis test
n	planned sample size
cor	planning value of correlation
h	null hypothesis value of correlation
s	number of control variables

**Value**

Returns the approximate power of the test

**Examples**

```
power.cor(.05, 80, .3, 0, 0)

# Should return:
#      Power
# 0.7751947
```



---

power.cor2	<i>Approximates the power of a test for equal correlations in a 2-group design for planned sample sizes</i>
------------	---

---

**Description**

Computes the approximate power of a test for equal population Pearson or partial correlations in a 2-group design for planned sample sizes. Set  $s = 0$  for a Pearson correlation.

**Usage**

```
power.cor2(alpha, n1, n2, cor1, cor2, s)
```

**Arguments**

alpha	alpha level for hypothesis test
n1	planned sample size for group 1
n2	planned sample size for group 2
cor1	planning value of correlation for group 1
cor2	planning value of correlation for group 1
s	number of control variables

**Value**

Returns the approximate power of the test

**Examples**

```
power.cor2(.05, 200, 200, .4, .2, 0)

# Should return:
#   Power
# 0.5919682
```

---

power.lc.mean.bs	<i>Approximates the power of a test for a linear contrast of means for planned sample sizes in a between-subjects design</i>
------------------	--

---

### Description

Computes the approximate power of a test for a linear contrast of population means for planned sample sizes in a between-subject design. The groups can be the factor levels of a single factor design or the combinations of factors in a factorial design. For a conservatively low power approximation, set the variance planning values to the largest values within their plausible ranges, and set the effect size to a minimally interesting value. The within-group variances can be unequal across groups and a Satterthwaite degree of freedom adjustment is used to improve the accuracy of the power approximation.

### Usage

```
power.lc.mean.bs(alpha, n, var, es, v)
```

### Arguments

alpha	alpha level for hypothesis test
n	vector of planned sample sizes
var	vector of within-group variance planning values
es	planning value of linear contrast of means
v	vector of contrast coefficients

### Value

Returns the approximate power of the test

### Examples

```
n <- c(20, 20, 20, 20)
var <- c(70, 70, 80, 80)
v <- c(.5, .5, -.5, -.5)
power.lc.mean.bs(.05, n, var, 5, v)

# Should return:
#   Power
# 0.7221171
```

---

power.mean	<i>Approximates the power of a one-sample t-test for a planned sample size</i>
------------	--

---

### Description

Computes the approximate power of a one-sample t-test for a planned sample size. For a conservatively low power approximation, set the variance planning value to the largest value within its plausible range, and set the effect size to a minimally interesting value.

### Usage

```
power.mean(alpha, n, var, es)
```

### Arguments

alpha	alpha level for hypothesis test
n	planned sample size
var	planning value of response variable variance
es	planning value of mean minus null hypothesis value

### Value

Returns the approximate power of the test

### Examples

```
power.mean(.05, 15, 80.5, 7)

# Should return:
#   Power
# 0.8021669
```

---

power.mean.ps	<i>Approximates the power of a paired-samples t-test for a planned sample size</i>
---------------	--

---

### Description

Computes the approximate power of a paired-samples t-test for a planned sample size. For a conservatively low power approximation, set the variance planning values to the largest values within their plausible ranges, set the correlation planning value to the smallest value within its plausible range, and set the effect size to a minimally interesting value. The variances of the two measurements can be unequal.

**Usage**

```
power.mean.ps(alpha, n, var1, var2, es, cor)
```

**Arguments**

alpha	alpha level for hypothesis test
n	planned sample size
var1	planning value of measurement 1 variance
var2	planning value of measurement 2 variance
es	planning value of mean difference
cor	planning value of correlation between measurements

**Value**

Returns the approximate power of the test

**Examples**

```
power.mean.ps(.05, 20, 10.0, 12.0, 2, .7)

# Should return:
#   Power
# 0.9074354
```

---

power.mean2	<i>Approximates the power of a two-sample t-test for planned sample sizes</i>
-------------	---

---

**Description**

Computes the approximate power of a two-sample t-test for planned sample sizes. For a conservatively low power approximation, set the variance planning values to the largest values within their plausible ranges, and set the effect size to a minimally interesting value. The within-group variances can be unequal across groups and a Satterthwaite degree of freedom adjustment is used to improve the accuracy of the power approximation.

**Usage**

```
power.mean2(alpha, n1, n2, var1, var2, es)
```

**Arguments**

alpha	alpha level for hypothesis test
n1	planned sample size for group 1
n2	planned sample size for group 2
var1	planning value of within-group variance for group 1
var2	planning value of within-group variance for group 2
es	planning value of mean difference

**Value**

Returns the approximate power of the test

**Examples**

```
power.mean2(.05, 25, 25, 5.0, 6.0, 2)

# Should return:
#   Power
# 0.8398417
```

---

power.prop	<i>Approximates the power of a 1-group proportion test for a planned sample size</i>
------------	--

---

**Description**

Computes the approximate power of a one-sample proportion test for a planned sample size. Set the proportion planning value to .5 for a conservatively low power estimate. The value of the effect size need not be based on the proportion planning value.

**Usage**

```
power.prop(alpha, n, p, es)
```

**Arguments**

alpha	alpha level for hypothesis test
n	planned sample size
p	planning value of proportion
es	planning value of proportion minus null hypothesis value

**Value**

Returns the approximate power of the test

## Examples

```
power.prop(.05, 40, .5, .2)

# Should return:
#   Power
# 0.7156044
```

---

power.prop.ps	<i>Approximates the power of a paired-samples test of equal proportions for a planned sample size</i>
---------------	---

---

## Description

Computes the approximate power of a test for equal population proportions in a paired-samples design (the McNemar test). 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 low power estimate. The planning value for the proportion difference (effect size) could be set to the difference of the two proportion planning values or it could be set to a minimally interesting effect size. Set the phi correlation planning value to the smallest value within a plausible range for a conservatively low power estimate.

## Usage

```
power.prop.ps(alpha, n, p1, p2, phi, es)
```

## Arguments

alpha	alpha level for hypothesis test
n	planned sample size
p1	planning value of proportion for measurement 1
p2	planning value of proportion for measurement 2
phi	planning value of phi correlation
es	planning value of proportion difference

## Value

Returns the approximate power of the test

**Examples**

```
power.prop.ps(.05, 45, .5, .5, .4, .2)

# Should return:
#   Power
# 0.6877704
```

---

power.prop2	<i>Approximates the power of a 2-group proportion test for planned sample sizes</i>
-------------	---

---

**Description**

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

**Usage**

```
power.prop2(alpha, n1, n2, p1, p2, es)
```

**Arguments**

alpha	alpha level for hypothesis test
n1	planned sample size for group 1
n2	planned sample size for group 2
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 approximate power of the test

**Examples**

```
power.prop2(.05, 60, 40, .5, .5, .2)

# Should return:
#   Power
# 0.4998959
```

random.sample                    *Generate a random sample*

---

**Description**

Generates a random sample of participant IDs without replacement.

**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                    *Generate random sample of scores*

---

**Description**

Generates a random sample of scores from a normal distribution with a specified population mean and standard deviation. This function is useful for generating hypothetical data for classroom demonstrations.

**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

*Generates random bivariate scores*


---

**Description**

Generates a random sample of y scores and x scores from a bivariate normal distributions with specified population means, standard deviations, and correlation. This function is useful for generating hypothetical data for classroom demonstrations.

**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

*Randomize a sample into groups*

---

**Description**

Randomly assigns a sample of participants into k groups.

**Usage**

```
randomize(n)
```

**Arguments**

n                    k x 1 vector of sample sizes

**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
```

---

sim.ci.cor	<i>Simulates confidence interval coverage probability for a Pearson correlation</i>
------------	---

---

### Description

Performs a computer simulation of confidence interval performance for a Pearson correlation. A bias adjustment is used to reduce the bias of the Fisher transformed Pearson correlation. Sample data can be generated from bivariate population distributions with five different marginal distributions. All distributions are scaled to have standard deviations of 1.0. Bivariate random data with specified marginal skewness and kurtosis are generated using the `unonr` function in the `mnonr` package.

### Usage

```
sim.ci.cor(alpha, n, cor, dist1, dist2, rep)
```

### Arguments

alpha	alpha level for 1-alpha confidence
n	sample size
cor	population Pearson correlation
dist1	type of distribution for variable 1 (1, 2, 3, 4, or 5)
dist2	type of distribution for variable 2 (1, 2, 3, 4, or 5) <ul style="list-style-type: none"> <li>• 1 = Gaussian (skewness = 0 and excess kurtosis = 0)</li> <li>• 2 = platykurtic (skewness = 0 and excess kurtosis = -1.2)</li> <li>• 3 = leptokurtic (skewness = 0 and excess kurtosis = 6)</li> <li>• 4 = moderate skew (skewness = 1 and excess kurtosis = 1.5)</li> <li>• 5 = large skew (skewness = 2 and excess kurtosis = 6)</li> </ul>
rep	number of Monte Carlo samples

### Value

Returns a 1-row matrix. The columns are:

- Coverage - probability of confidence interval including population correlation
- Lower Error - probability of lower limit greater than population correlation
- Upper Error - probability of upper limit less than population correlation
- Ave CI Width - average confidence interval width

**Examples**

```
sim.ci.cor(.05, 30, .7, 4, 5, 1000)

# Should return (within sampling error):
# Coverage Lower Error Upper Error Ave CI Width
# [1,] 0.93815 0.05125 0.0106 0.7778518
```

---

sim.ci.mean	<i>Simulates confidence interval coverage probability for a mean</i>
-------------	--

---

**Description**

Performs a computer simulation of the confidence interval performance for a population mean. Sample data can be generated from five different population distributions. All distributions are scaled to have standard deviations of 1.0.

**Usage**

```
sim.ci.mean(alpha, n, dist, rep)
```

**Arguments**

alpha	alpha level for 1-alpha confidence
n	sample size
dist	type of distribution (1, 2, 3, 4, or 5) <ul style="list-style-type: none"> <li>• 1 = Gaussian (skewness = 0 and excess kurtosis = 0)</li> <li>• 2 = platykurtic (skewness = 0 and excess kurtosis = -1.2)</li> <li>• 3 = leptokurtic (skewness = 0 and excess kurtosis = 6)</li> <li>• 4 = moderate skew (skewness = 1 and excess kurtosis = 1.5)</li> <li>• 5 = large skew (skewness = 2 and excess kurtosis = 6)</li> </ul>
rep	number of Monte Carlo samples

**Value**

Returns a 1-row matrix. The columns are:

- Coverage - probability of confidence interval including population mean
- Lower Error - probability of lower limit greater than population mean
- Upper Error - probability of upper limit less than population mean
- Ave CI Width - average confidence interval width

**Examples**

```
sim.ci.mean(.05, 40, 4, 1000)

# Should return (within sampling error):
# Coverage Lower Error Upper Error Ave CI Width
# 0.94722    0.01738    0.0354    0.6333067
```

---

sim.ci.mean.ps	<i>Simulates confidence interval coverage probability for a paired-samples mean difference</i>
----------------	--

---

**Description**

Performs a computer simulation of confidence interval performance for a population mean difference in a paired-samples design. Sample data for the two levels of the within-subjects factor can be generated from bivariate population distributions with five different marginal distributions. All distributions are scaled to have standard deviations of 1.0 at level 1. Bivariate random data with specified marginal skewness and kurtosis are generated using the `unonr` function in the `mnonr` package.

**Usage**

```
sim.ci.mean.ps(alpha, n, sd.ratio, cor, dist1, dist2, rep)
```

**Arguments**

alpha	alpha level for 1-alpha confidence
n	sample size
sd.ratio	ratio of population standard deviations
cor	population correlation of paired observations
dist1	type of distribution at level 1 (1, 2, 3, 4, or 5)
dist2	type of distribution at level 2 (1, 2, 3, 4, or 5) <ul style="list-style-type: none"> <li>• 1 = Gaussian (skewness = 0 and excess kurtosis = 0)</li> <li>• 2 = platykurtic (skewness = 0 and excess kurtosis = -1.2)</li> <li>• 3 = leptokurtic (skewness = 0 and excess kurtosis = 6)</li> <li>• 4 = moderate skew (skewness = 1 and excess kurtosis = 1.5)</li> <li>• 5 = large skew (skewness = 2 and excess kurtosis = 6)</li> </ul>
rep	number of Monte Carlo samples

**Value**

Returns a 1-row matrix. The columns are:

- Coverage - probability of confidence interval including population mean difference
- Lower Error - probability of lower limit greater than population mean difference
- Upper Error - probability of upper limit less than population mean difference
- Ave CI Width - average confidence interval width

**Examples**

```
sim.ci.mean.ps(.05, 30, 1.5, .7, 4, 5, 1000)

# Should return (within sampling error):
# Coverage Lower Error Upper Error Ave CI Width
# 0.93815    0.05125    0.0106    0.7778518
```

---

sim.ci.mean2	<i>Simulates confidence interval coverage probability for a 2-group mean difference</i>
--------------	---

---

**Description**

Performs a computer simulation of separate variance and pooled variance confidence interval performance for a population mean difference in a 2-group design. Sample data within each group can be generated from five different population distributions. All distributions are scaled to have a standard deviation of 1.0 in group 1.

**Usage**

```
sim.ci.mean2(alpha, n1, n2, sd.ratio, dist1, dist2, rep)
```

**Arguments**

alpha	alpha level for 1-alpha confidence
n1	sample size in group 1
n2	sample size in group 2
sd.ratio	ratio of population standard deviations (sd2/sd1)
dist1	type of distribution for group 1 (1, 2, 3, 4, or 5)
dist2	type of distribution for group 2 (1, 2, 3, 4, or 5) <ul style="list-style-type: none"> <li>• 1 = Gaussian (skewness = 0 and excess kurtosis = 0)</li> <li>• 2 = platykurtic (skewness = 0 and excess kurtosis = -1.2)</li> <li>• 3 = leptokurtic (skewness = 0 and excess kurtosis = 6)</li> <li>• 4 = moderate skew (skewness = 1 and excess kurtosis = 1.5)</li> <li>• 5 = large skew (skewness = 2 and excess kurtosis = 6)</li> </ul>
rep	number of Monte Carlo samples

**Value**

Returns a 1-row matrix. The columns are:

- Coverage - probability of confidence interval including population mean difference
- Lower Error - probability of lower limit greater than population mean difference
- Upper Error - probability of upper limit less than population mean difference
- Ave CI Width - average confidence interval width

**Examples**

```
sim.ci.mean2(.05, 30, 25, 1.5, 4, 5, 1000)

# Should return (within sampling error):
#           Coverage Lower Error Upper Error Ave CI Width
# Equal Variances Assumed:    0.93986    0.04022    0.01992    1.344437
# Equal Variances Not Assumed: 0.94762    0.03862    0.01376    1.401305
```

---

sim.ci.median	<i>Simulates confidence interval coverage probability for a median</i>
---------------	--

---

**Description**

Performs a computer simulation of the confidence interval performance for a population median. Sample data can be generated from five different population distributions. All distributions are scaled to have standard deviations of 1.0.

**Usage**

```
sim.ci.median(alpha, n, dist, rep)
```

**Arguments**

alpha	alpha level for 1-alpha confidence
n	sample size
dist	type of distribution (1, 2, 3, 4, or 5) <ul style="list-style-type: none"> <li>• 1 = Gaussian (skewness = 0 and excess kurtosis = 0)</li> <li>• 2 = platykurtic (skewness = 0 and excess kurtosis = -1.2)</li> <li>• 3 = leptokurtic (skewness = 0 and excess kurtosis = 6)</li> <li>• 4 = moderate skew (skewness = 1 and excess kurtosis = 1.5)</li> <li>• 5 = large skew (skewness = 2 and excess kurtosis = 6)</li> </ul>
rep	number of Monte Carlo samples

**Value**

Returns a 1-row matrix. The columns are:

- Coverage - probability of confidence interval including population median
- Lower Error - probability of lower limit greater than population median
- Upper Error - probability of upper limit less than population median
- Ave CI Width - average confidence interval width

**Examples**

```
sim.ci.median(.05, 20, 5, 1000)

# Should return (within sampling error):
# Coverage Lower Error Upper Error Ave CI Width
# 0.9589      0.0216      0.0195      0.9735528
```

---

sim.ci.median.ps	<i>Simulates confidence interval coverage probability for a median difference in a paired-samples design</i>
------------------	--

---

**Description**

Performs a computer simulation of confidence interval performance for a population median difference in a paired-samples design. Sample data for the two levels of the within-subjects factor can be generated from bivariate population distributions with five different marginal distributions. All distributions are scaled to have standard deviations of 1.0 at level 1. Bivariate random data with specified marginal skewness and kurtosis are generated using the unonr function in the mnonr package.

**Usage**

```
sim.ci.median.ps(alpha, n, sd.ratio, cor, dist1, dist2, rep)
```

**Arguments**

alpha	alpha level for 1-alpha confidence
n	sample size
sd.ratio	ratio of population standard deviations
cor	population correlation of paired observations
dist1	type of distribution at level 1 (1, 2, 3, 4, or 5)
dist2	type of distribution at level 2 (1, 2, 3, 4, or 5) <ul style="list-style-type: none"> <li>• 1 = Gaussian (skewness = 0 and excess kurtosis = 0)</li> <li>• 2 = platykurtic (skewness = 0 and excess kurtosis = -1.2)</li> </ul>



- 3 = leptokurtic (skewness = 0 and excess kurtosis = 6)
- 4 = moderate skew (skewness = 1 and excess kurtosis = 1.5)
- 5 = large skew (skewness = 2 and excess kurtosis = 6)

rep                    number of Monte Carlo samples

### Value

Returns a 1-row matrix. The columns are:

- Coverage - probability of confidence interval including population median difference
- Lower Error - probability of lower limit greater than population median difference
- Upper Error - probability of upper limit less than population median difference
- Ave CI Width - average confidence interval width

### Examples

```
sim.ci.median.ps(.05, 30, 1.5, .7, 4, 3, 1000)

# Should return (within sampling error):
# Coverage Lower Error Upper Error Ave CI Width
# 0.961      0.026      0.013      0.9435462
```

---

sim.ci.median2	<i>Simulates confidence interval coverage probability for a median difference in a 2-group design</i>
----------------	---

---

### Description

Performs a computer simulation of the confidence interval performance for a difference of population medians in a 2-group design. Sample data for each group can be generated from five different population distributions. All distributions are scaled to have standard deviations of 1.0 in group 1.

### Usage

```
sim.ci.median2(alpha, n1, n2, sd.ratio, dist1, dist2, rep)
```

### Arguments

alpha	alpha level for 1-alpha confidence
n1	sample size for group 1
n2	sample size for group 2
sd.ratio	ratio of population standard deviations (sd2/sd1)
dist1	type of distribution for group 1 (1, 2, 3, 4, or 5)

dist2            type of distribution for group 2 (1, 2, 3, 4, or 5)

- 1 = Gaussian (skewness = 0 and excess kurtosis = 0)
- 2 = platykurtic (skewness = 0 and excess kurtosis = -1.2)
- 3 = leptokurtic (skewness = 0 and excess kurtosis = 6)
- 4 = moderate skew (skewness = 1 and excess kurtosis = 1.5)
- 5 = large skew (skewness = 2 and excess kurtosis = 6)

rep              number of Monte Carlo samples

### Value

Returns a 1-row matrix. The columns are:

- Coverage - Probability of confidence interval including population median difference
- Lower Error - Probability of lower limit greater than population median difference
- Upper Error - Probability of upper limit less than population median difference
- Ave CI Width - Average confidence interval width

### Examples

```
sim.ci.median2(.05, 20, 20, 2, 5, 4, 5000)

# Should return (within sampling error):
# Coverage Lower Error Upper Error Ave CI Width
# 0.952      0.027      0.021      2.368914
```

---

sim.ci.spear            *Simulates confidence interval coverage probability for a Spearman correlation*

---

### Description

Performs a computer simulation of confidence interval performance for a Spearman correlation. Sample data can be generated from bivariate population distributions with five different marginal distributions. All distributions are scaled to have standard deviations of 1.0. Bivariate random data with specified marginal skewness and kurtosis are generated using the unonr function in the mnonr package.

### Usage

```
sim.ci.spear(alpha, n, cor, dist1, dist2, rep)
```

**Arguments**

alpha	alpha level for 1-alpha confidence
n	sample size
cor	population Spearman correlation
dist1	type of distribution for variable 1 (1, 2, 3, 4, or 5)
dist2	type of distribution for variable 2 (1, 2, 3, 4, or 5) <ul style="list-style-type: none"> <li>• 1 = Gaussian (skewness = 0 and excess kurtosis = 0)</li> <li>• 2 = platykurtic (skewness = 0 and excess kurtosis = -1.2)</li> <li>• 3 = leptokurtic (skewness = 0 and excess kurtosis = 6)</li> <li>• 4 = moderate skew (skewness = 1 and excess kurtosis = 1.5)</li> <li>• 5 = large skew (skewness = 2 and excess kurtosis = 6)</li> </ul>
rep	number of Monte Carlo samples

**Value**

Returns a 1-row matrix. The columns are:

- Coverage - probability of confidence interval including population correlation
- Lower Error - probability of lower limit greater than population correlation
- Upper Error - probability of upper limit less than population correlation
- Ave CI Width - average confidence interval width

**Examples**

```
sim.ci.spear(.05, 30, .7, 4, 5, 1000)

# Should return (within sampling error):
#      Coverage Lower Error Upper Error Ave CI Width
# [1,] 0.96235    0.01255    0.0251    0.4257299
```

---

sim.ci.stdmean.ps	<i>Simulates confidence interval coverage probability for a standardized mean difference in a paired-samples design</i>
-------------------	---

---

**Description**

Performs a computer simulation of confidence interval performance for two types of standardized mean differences in a paired-samples design (see ci.stdmean.ps). Sample data for the two levels of the within-subjects factor can be generated from five different population distributions. All distributions are scaled to have standard deviations of 1.0 at level 1.

**Usage**

```
sim.ci.stdmean.ps(alpha, n, sd.ratio, cor, dist1, dist2, d, rep)
```

**Arguments**

alpha	alpha level for 1-alpha confidence
n	sample size
sd.ratio	ratio of population standard deviations (sd2/sd1)
cor	correlation between paired measurements
dist1	type of distribution at level 1 (1, 2, 3, 4, or 5)
dist2	type of distribution at level 2 (1, 2, 3, 4, or 5) <ul style="list-style-type: none"> <li>• 1 = Gaussian (skewness = 0 and excess kurtosis = 0)</li> <li>• 2 = platykurtic (skewness = 0 and excess kurtosis = -1.2)</li> <li>• 3 = leptokurtic (skewness = 0 and excess kurtosis = 6)</li> <li>• 4 = moderate skew (skewness = 1 and excess kurtosis = 1.5)</li> <li>• 5 = large skew (skewness = 2 and excess kurtosis = 6)</li> </ul>
d	population standardized mean difference
rep	number of Monte Carlo samples

**Value**

Returns a 1-row matrix. The columns are:

- Coverage - Probability of confidence interval including population std mean difference
- Lower Error - Probability of lower limit greater than population std mean difference
- Upper Error - Probability of upper limit less than population std mean difference
- Ave CI Width - Average confidence interval width

**Examples**

```
sim.ci.stdmean.ps(.05, 20, 1.5, .8, 4, 4, .5, 2000)

# Should return (within sampling error):
#           Coverage Lower Error Upper Error Ave CI Width  Ave Est
# Unweighted Standardizer  0.9095    0.0555    0.035    0.7354865 0.5186796
# Level 1 Standardizer    0.9525    0.0255    0.022    0.9330036 0.5058198
```

---

sim.ci.stdmean2	<i>Simulates confidence interval coverage probability for a standardized mean difference in a 2-group design</i>
-----------------	--

---

### Description

Performs a computer simulation of confidence interval performance for two types of standardized mean differences in a 2-group design (see ci.stdmean2). Sample data for each group can be generated from five different population distributions. All distributions are scaled to have standard deviations of 1.0 in group 1.

### Usage

```
sim.ci.stdmean2(alpha, n1, n2, sd.ratio, dist1, dist2, d, rep)
```

### Arguments

alpha	alpha level for 1-alpha confidence
n1	sample size for group 1
n2	sample size for group 2
sd.ratio	ratio of population standard deviations (sd2/sd1)
dist1	type of distribution for group 1 (1, 2, 3, 4, or 5)
dist2	type of distribution for group 2 (1, 2, 3, 4, or 5) <ul style="list-style-type: none"> <li>• 1 = Gaussian (skewness = 0 and excess kurtosis = 0)</li> <li>• 2 = platykurtic (skewness = 0 and excess kurtosis = -1.2)</li> <li>• 3 = leptokurtic (skewness = 0 and excess kurtosis = 6)</li> <li>• 4 = moderate skew (skewness = 1 and excess kurtosis = 1.5)</li> <li>• 5 = large skew (skewness = 2 and excess kurtosis = 6)</li> </ul>
d	population standardized mean difference
rep	number of Monte Carlo samples

### Value

Returns a 1-row matrix. The columns are:

- Coverage - Probability of confidence interval including population std mean difference
- Lower Error - Probability of lower limit greater than population std mean difference
- Upper Error - Probability of upper limit less than population std mean difference
- Ave CI Width - Average confidence interval width

**Examples**

```

sim.ci.stdmean2(.05, 20, 20, 1.5, 3, 4, .75, 5000)

# Should return (within sampling error):
#
#           Coverage Lower Error Upper Error Ave CI Width  Ave Est
# Unweighted Standardizer  0.9058    0.0610    0.0332    1.342560 0.7838679
# Group 1 Standardizer    0.9450    0.0322    0.0228    1.827583 0.7862640

```

---

size.ci.agree                      *Sample size for a G-index confidence interval*

---

**Description**

Computes the sample size required to estimate a population 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
#           139

```

---

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

---

### Description

Computes the total sample size required to estimate a population 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 variable
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
#           210
```

---

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

---

**Description**

Computes the sample size required to estimate a population Pearson or partial correlation with desired confidence interval precision. Set  $s = 0$  for a Pearson correlation. Set the correlation planning value to the smallest absolute 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).

**Examples**

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

# Should return:
# Sample size
#           188
```



---

size.ci.cor.prior	<i>Sample size for a Pearson correlation confidence interval using a planning value from a prior study</i>
-------------------	--

---

### Description

Computes the sample size required to estimate a Pearson correlation with desired confidence interval precision in applications where an estimated Pearson correlation from a prior study is available. The actual confidence interval width in the planned study will depend on the value of the estimated correlation in the planned study. An estimated correlation from a prior study is used to predict the value of the estimated correlation in the planned study, and the predicted correlation estimate is then used in the sample size computation.

This sample size approach assumes that the population Pearson correlation in the prior study is very similar to the population Pearson correlation in the planned study. In a typical sample size analysis, this type of information is not available, and the researcher must use expert opinion to guess the value of the Pearson correlation that will be observed in the planned study. The `size.ci.cor` function uses a correlation planning value that is based on expert opinion regarding the likely value of the correlation estimate that will be observed in the planned study.

### Usage

```
size.ci.cor.prior(alpha1, alpha2, cor0, n0, w)
```

### Arguments

alpha1	alpha level for 1-alpha1 confidence in the planned study
alpha2	alpha level for the 1-alpha2 prediction interval
cor0	estimated correlation in prior study
n0	sample size in prior study
w	desired confidence interval width

### Value

Returns the required sample size

### Examples

```
size.ci.cor.prior(.05, .10, .438, 100, .2)

# Should return:
# Sample size
#           331
```

---

size.ci.cor2	<i>Sample size for a 2-group Pearson correlation difference confidence interval</i>
--------------	---

---

### Description

Computes the sample size required to estimate a difference in population Pearson or partial correlations with desired confidence interval precision in a 2-group design. Set the correlation planning values to the smallest absolute values within their plausible ranges for a conservatively large sample size.

### Usage

```
size.ci.cor2(alpha, cor1, cor2, w)
```

### Arguments

alpha	alpha level for 1-alpha confidence
cor1	group 1 correlation planning value
cor2	group 2 correlation planning value
w	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).

### Examples

```
size.ci.cor2(.05, .8, .5, .2)

# Should return:
# Sample size per group
#                271
```

---

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. Set the reliability planning value to the smallest value within a plausible range for a conservatively large sample size.

### Usage

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

### Arguments

alpha	alpha value for 1-alpha confidence
rel	reliability planning value
r	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://doi.org/10.1002/job.1960).

### Examples

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

# Should return:
# Sample size
#           89
```

---

size.ci.cronbach2	<i>Sample size for a 2-group Cronbach reliability difference confidence interval</i>
-------------------	--

---

### Description

Computes the sample size required to estimate a difference in population Cronbach reliability coefficients with desired precision in a 2-group design.

### Usage

```
size.ci.cronbach2(alpha, rel1, rel2, r, w)
```

### Arguments

alpha	alpha level for hypothesis test
rel1	group 1 reliability planning value
rel2	group 2 reliability planning value
r	number of measurements (items, raters)
w	desired confidence interval width

### Value

Returns the required sample size for each group

### 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).

### Examples

```
size.ci.cronbach2(.05, .85, .70, 8, .15)

# Should return:
# Sample size per group
#           180
```

---

size.ci.etasqr      *Sample size for an eta-squared confidence interval*

---

**Description**

Computes the sample size required to estimate an eta-squared coefficient in a one-way ANOVA with desired confidence interval precision. Set the planning value of eta-squared to about 1/3 for a conservatively large sample size.

**Usage**

```
size.ci.etasqr(alpha, etasqr, groups, w)
```

**Arguments**

alpha	alpha level for 1-alpha confidence
etasqr	planning value of eta-squared
groups	number of groups
w	desired confidence interval width

**Value**

Returns the required sample size for each group

**Examples**

```
size.ci.etasqr(.05, .333, 3, .2)

# Should return:
# Sample size per group
#                   63
```

---

size.ci.indirect      *Sample size for an indirect effect confidence interval*

---

**Description**

Computes the approximate sample size required to estimate a population standardized indirect effect in a simple mediation model. The direct effect of the independent (exogenous) variable on the response variable, controlling for the mediator variable, is assumed to be negligible.

**Usage**

```
size.ci.indirect(alpha, cor1, cor2, w)
```

**Arguments**

alpha	alpha level for 1-alpha confidence
cor1	planning value of correlation between the independent and mediator variables
cor2	planning value of correlation between the mediator and response variables
w	desired confidence interval width

**Value**

Returns the required sample size

**Examples**

```
size.ci.indirect(.05, .4, .5, .2)

# Should return:
# Sample size
#           106
```

---

size.ci.lc.ancova      *Sample size for a linear contrast confidence interval in an ANCOVA*

---

**Description**

Computes the sample size for each group (assuming equal sample sizes) required to estimate a population linear contrast of means in an ANCOVA 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 for each group

**Examples**

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

# Should return:
# Sample size per group
#                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
#                34
```

---

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

---

### 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 Pearson 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
#           11
```



---

size.ci.lc.prop.bs      *Sample size for a between-subjects proportion linear contrast confidence interval*

---

### Description

Computes the sample size in each group (assuming equal sample sizes) required to estimate a linear contrast of population 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 for each 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
#                87
```

---

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

---

### Description

Computes the sample size per group (assuming equal sample sizes) required to estimate two types of standardized linear contrasts of population means (unweighted average standardizer and single group standardizer) with desired confidence interval precision in a between-subjects design. Set the standardized linear contrast of means 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 of means
w	desired confidence interval width
v	vector of between-subjects contrast coefficients

**Value**

Returns the required sample size per group for each standardizer

**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).

**Examples**

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

# Should return:
#                Sample size per group
# Unweighted standardizer:           49
# Single group standardizer:         65
```

---

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

---

**Description**

Computes the sample size required to estimate two types of standardized linear contrasts of population means (unweighted standardizer and single level standardizer) with desired confidence interval precision in a within-subjects design. For a conservatively large sample size, set the standardized linear contrast of means planning value to the largest value within a plausible range, and set the Pearson correlation planning value to the smallest value within a plausible range.

**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 for each standardizer

**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.

**Examples**

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

# Should return:
#                               Sample size
# Unweighted standardizer:         26
# Single level standardizer:       35
```

---

size.ci.mape	<i>Sample size for a mean absolute prediction error confidence interval</i>
--------------	---

---

**Description**

Computes the sample size required to estimate a population mean absolute prediction error for a general linear model with desired confidence interval precision. Setting  $s = 0$  gives the sample size requirement for a mean absolute deviation in a one-group design. This function assumes that the prediction errors have an approximate normal distribution.

**Usage**

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

**Arguments**

alpha	alpha value for 1-alpha confidence
mape	mean absolute prediction error planning value
s	number of predictor variables
w	desired confidence interval width

**Value**

Returns the required sample size

**Examples**

```
size.ci.mape(.05, 4.5, 5, 2)

# Should return:
# Sample size
#           57
```

---

size.ci.mean

*Sample size for a mean confidence interval*

---

**Description**

Computes the sample size required to estimate a 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.mean(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.mean(.05, 264.4, 10)

# Should return:
# Sample size
#           43
```

---

size.ci.mean.prior	<i>Sample size for a mean confidence interval using a planning value from a prior study</i>
--------------------	---

---

### Description

Computes the sample size required to estimate a population mean with desired confidence interval precision in applications where an estimated variance from a prior study is available. The actual confidence interval width in the planned study will depend on the value of the estimated variance in the planned study. An estimated variance from a prior study is used to predict the value of the estimated correlation in the planned study, and the predicted variance estimate is then used in the sample size computation.

This sample size approach assumes that the population variance in the prior study is very similar to the population variance in the planned study. In a typical sample size analysis, this type of information is not available, and the researcher must use expert opinion to guess the value of the variance that will be observed in the planned study. The `size.ci.mean` function uses a variance planning value that is based on expert opinion regarding the likely value of the variance estimate that will be observed in the planned study.

### Usage

```
size.ci.mean.prior(alpha1, alpha2, var0, n0, w)
```

### Arguments

alpha1	alpha level for 1-alpha1 confidence in the planned study
alpha2	alpha level for the 1-alpha2 prediction interval
var0	estimated variance in prior study
n0	sample size in prior study
w	desired confidence interval width

### Value

Returns the required sample size

### Examples

```
size.ci.mean.prior(.05, .10, 26.4, 25, 4)

# Should return:
# Sample size
#           44
```

---

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

---

### 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 Pearson 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
#           19
```

---

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

---

### 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
# 47 47
```

---

```
size.ci.pbcor
```

*Sample size for a point-biserial correlation confidence interval*

---

**Description**

Computes the sample size required to estimate a population point-biserial correlation with desired confidence interval precision in a two-group nonexperimental design with simple random sampling. A two-group nonexperimental design implies two subpopulations (e.g., all boys and all girls in a school district). This function requires a planning value for the proportion of population members who belong to one of the two subpopulations. Set the correlation planning value to the smallest absolute value within a plausible range for a conservatively large sample size.

**Usage**

```
size.ci.pbcor(alpha, cor, w, p)
```

**Arguments**

alpha	alpha level for 1-alpha confidence
cor	planning value of point-biserial correlation
w	desired confidence interval width
p	proportion of members in one of the two subpopulations

**Value**

Returns the required sample size

**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.

**Examples**

```
size.ci.pbcor(.05, .40, .25, .73)

# Should return:
# Sample size
#           168
```

---

size.ci.prop

*Sample size for a proportion confidence interval*

---

**Description**

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

**Usage**

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

# Should return:
# Sample size
#           93
```



---

size.ci.prop.prior	<i>Sample size for a proportion confidence interval using a planning value from a prior study</i>
--------------------	---

---

### Description

Computes the sample size required to estimate a population proportion with desired confidence interval precision in applications where an estimated proportion from a prior study is available. The actual confidence interval width in the planned study will depend on the value of the estimated proportion in the planned study. An estimated proportion from a prior study is used to predict the value of the estimated proportion in the planned study, and the predicted proportion estimate is then used in the sample size computation.

This sample size approach assumes that the population proportion in the prior study is very similar to the population proportion in the planned study. In a typical sample size analysis, this type of information is not available, and the researcher must use expert opinion to guess the value of the proportion that will be observed in the planned study. The `size.ci.prop` function uses a proportion planning value that is based on expert opinion regarding the likely value of the proportion estimate that will be observed in the planned study.

### Usage

```
size.ci.prop.prior(alpha1, alpha2, p0, n0, w)
```

### Arguments

alpha1	alpha level for 1-alpha1 confidence in the planned study
alpha2	alpha level for the 1-alpha2 prediction interval
p0	estimated proportion in prior study
n0	sample size in prior study
w	desired confidence interval width

### Value

Returns the required sample size

### Examples

```
size.ci.prop.prior(.05, .20, .1425, 200, .1)

# Should return:
# Sample size
#           318
```

---

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 population 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 correlation 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 measurement 1
p2	planning value of proportion for measurement 2
phi	planning value of phi correlation
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
#      118
```

---

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

---

### Description

Computes the sample size in each group (assuming equal sample sizes) required to estimate a difference of population 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 for each group

### Examples

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

# Should return:
# Sample size per group
#                      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
#           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
# 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 population proportions with desired confidence interval precision in a paired-samples design. Set the phi correlation 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 correlation
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
#           67
```

---

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

---

**Description**

Computes the sample size in each group (assuming equal sample sizes) required to estimate a ratio of population 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 for each group

**Examples**

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

# Should return:
# Sample size per group
#           416
```

---

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

---

**Description**

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

**Usage**

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

**Arguments**

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

**Value**

Returns the required sample size

**Examples**

```
size.ci.rsqr(.05, .25, 5, .2)

# Should return:
# Sample size
#           214
```

---

size.ci.second	<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
#                70
```

---

```
size.ci.slope
```

*Sample size for a slope confidence interval*

---

**Description**

Computes the total sample size required to estimate a population 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:
# Total sample size
#                83
```

---

size.ci.spear	<i>Sample size for a Spearman correlation confidence interval</i>
---------------	---

---

**Description**

Computes the sample size required to estimate a population Spearman correlation with desired confidence interval precision. Set the correlation planning value to the smallest absolute value within a plausible range for a conservatively large sample size.

**Usage**

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

**Arguments**

alpha	alpha level for 1-alpha confidence
cor	planning value of Spearman correlation
w	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).

**Examples**

```
size.ci.spear(.05, .362, .25)

# Should return:
# Sample size
#           200
```

---

size.ci.spear2	<i>Sample size for a 2-group Spearman correlation difference confidence interval</i>
----------------	--

---

**Description**

Computes the sample size required to estimate a difference in population Spearman correlations with desired confidence interval precision in a 2-group design. Set the correlation planning values to the smallest absolute values within their plausible ranges for a conservatively large sample size.

**Usage**

```
size.ci.spear2(alpha, cor1, cor2, w)
```

**Arguments**

alpha	alpha level for 1-alpha confidence
cor1	group 1 Spearman correlation planning value
cor2	group 2 Spearman correlation planning value
w	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.

**Examples**

```
size.ci.spear2(.05, .8, .5, .2)

# Should return:
# Sample size per group
#           314
```

---

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 two types of population standardized mean differences (unweighted standardizer and single group standardizer) 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, and set the Pearson 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 for each standardizer

## 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).

## Examples

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

# Should return:
#               Sample Size
# Unweighted standardizer: 46
# Single group standardizer: 52
```

---

size.ci.stdmean2      *Sample size for a 2-group standardized mean difference confidence interval*

---

### Description

Computes the sample size per group required to estimate two types of population standardized mean differences (unweighted standardizer and single group standardizer) 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	n2/n1 ratio

### Value

Returns the required sample size per group for each standardizer

### 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).

### Examples

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

# Should return:
#               n1  n2
# Unweighted standardizer: 132 132
# Single group standardizer: 141 141
```

---

`size.ci.tetra`*Sample size for a tetrachoric correlation confidence interval*

---

**Description**

Computes the sample size required to estimate a tetrachoric correlation with desired confidence interval precision. Set the tetrachoric planning value to the smallest value within a plausible range for a conservatively large sample size.

**Usage**

```
size.ci.tetra(alpha, p1, p2, cor, w)
```

**Arguments**

alpha	alpha level for 1 - alpha confidence
p1	planning value for row 1 marginal proportion
p2	planning value for column 1 marginal proportion
cor	tetrachoric planning value
w	desired confidence interval width

**Value**

Returns the required sample size

**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://doi.org/10.3102/10769986030002213).

**Examples**

```
size.ci.tetra(.05, .4, .3, .5, .3)

# Should return:
# Sample size
#           296
```

---

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 x alpha rather than alpha (e.g., use alpha = .10 rather alpha = .05). Set the Pearson correlation value to the smallest value within a plausible range, and 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 variance of the two measurements
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
#           68
```

---

size.equiv.mean2	<i>Sample size for a 2-group mean equivalence test</i>
------------------	--

---

### 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 for each group

### Examples

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

# Should return:
# Sample size per group
#                50
```

---

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

---

### Description

Computes the sample size 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 correlation 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

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 correlation
h	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
#           173
```



---

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

---

### Description

Computes the sample size in each group (assuming equal sample sizes) 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 for each group

### Examples

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

# Should return:
# Sample size per group
#           288
```

---

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

---

## Description

Computes the sample size required to perform an interval test for a population Pearson or a partial correlation with desired power where the interval midpoint is equal to zero. This function can be used to plan a study where the goal is to show that the population correlation is small. Set  $s = 0$  for a Pearson correlation. The correlation planning value must be a value within the hypothesized 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
#           360
```

---

size.supinf.mean.ps	<i>Sample size for a paired-samples mean superiority or noninferiority test</i>
---------------------	---

---

### 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 Pearson correlation planning value to the smallest value within a plausible range, and 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 variance of the two measurements
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
#           38
```

---

size.supinf.mean2	<i>Sample size for a 2-group mean superiority or noninferiority test</i>
-------------------	--

---

### 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 for each group

### Examples

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

# Should return:
# Sample size per group
#                143
```

---

size.supinf.prop.ps	<i>Sample size for a paired-samples superiority or inferiority test of proportions</i>
---------------------	--

---

### 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 correlation 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 correlation
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
#      227
```

---

size.supinf.prop2	<i>Sample size for a 2-group superiority or inferiority test of proportions</i>
-------------------	---

---

### Description

Computes the sample size in each group (assuming equal sample sizes) 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 for each group

### Examples

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

# Should return:
# Sample size per group
#                408
```

---

size.test.cor	<i>Sample size for a test of a Pearson or partial correlation</i>
---------------	---

---

**Description**

Computes the sample size required to test a population Pearson or a partial correlation with desired power. Set  $s = 0$  for a Pearson correlation.

**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	null hypothesis value of correlation

**Value**

Returns the required sample size

**Examples**

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

# Should return:
# Sample size
#           48
```

---

size.test.cor2	<i>Sample size for a test of equal Pearson or partial correlation in a 2-group design</i>
----------------	---

---

**Description**

Computes the sample size required to test the equality of two population Pearson or partial correlations with desired power in a 2-group design. Set  $s = 0$  for a Pearson correlation.

**Usage**

```
size.test.cor2(alpha, pow, cor1, cor2, s)
```

**Arguments**

alpha	alpha level for hypothesis test
pow	desired power
cor1	planning value of correlation for group 1
cor2	planning value of correlation for group 2
s	number of control variables

**Value**

Returns the required sample size for each group

**Examples**

```
size.test.cor2(.05, .8, .4, .2, 0)

# Should return:
# Sample size per group
#           325
```

---

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, r, h)
```

**Arguments**

alpha	alpha level for hypothesis test
pow	desired power
rel	reliability planning value
r	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.

## Examples

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

# Should return:
# Sample size
#           139
```

---

size.test.cronbach2	<i>Sample size to test equality of Cronbach reliability coefficients in a 2-group design</i>
---------------------	--

---

## Description

Computes the sample size required to test a difference in population Cronbach reliability coefficients with desired power in a 2-group design.

## Usage

```
size.test.cronbach2(alpha, pow, rel1, rel2, r)
```

## Arguments

alpha	alpha level for hypothesis test
pow	desired power
rel1	reliability planning value for group 1
rel2	reliability planning value for group 2
r	number of measurements (items, raters)

## Value

Returns the required sample size for each group

## 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.

## Examples

```
size.test.cronbach2(.05, .80, .85, .70, 8)

# Should return:
# Sample size per group
#                77
```

---

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

---

## Description

Computes the sample size for each group (assuming equal sample sizes) required to test a linear contrast of population 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 for each 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
#                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 for each 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
#                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 average variance planning value to the largest value within a plausible range for a conservatively large sample size. Set the average 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
#           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 (assuming equal sample sizes) 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 effect size (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 for each 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
#           105
```

---

`size.test.mann`*Sample size for a Mann-Whitney test*

---

### 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

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

### Value

Returns the required sample size for each 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).

### Examples

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

# Should return:
# Sample size per group
#                44
```

---

size.test.mean	<i>Sample size for a test of a 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.mean(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.mean(.05, .9, 80.5, 7)

# Should return:
# Sample size
#           20
```

---

size.test.mean.ps	<i>Sample size for a test of a paired-samples mean difference</i>
-------------------	---

---

**Description**

Computes the sample size required to test a difference in population means with desired power in a paired-samples design. Set the Pearson correlation planning value to the smallest value within a plausible range, and 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
#           22
```

---

size.test.mean2	<i>Sample size for a test of a 2-group mean difference</i>
-----------------	--

---

**Description**

Computes the sample size in each group required 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 for each group

**Examples**

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

# Should return:
# n1 n2
# 27 27
```

---

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

---

**Description**

Computes the sample size required to test a population proportion with desired power using a correction for continuity in a 1-group design.

**Usage**

```
size.test.prop(alpha, pow, p, h)
```

**Arguments**

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

**Value**

Returns the required sample size

**References**

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

**Examples**

```
size.test.prop(.05, .9, .5, .3)

# Should return:
# Sample size
#           65
```

---

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 be set to .5 for a conservatively large sample size. The planning value for the effect size (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 correlation 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 correlation
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
#           177
```

---

`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 and a continuity correction in a 2-group design. This function requires planning values for both proportions. Set each proportion planning value to .5, or a value closest to .5 within a plausible range, for a conservatively large sample size requirement. This function does not use the typical sample size approach where the effect size is assumed to equal the difference in proportion planning values. This function does not require the planning value for the proportion difference (effect size) to equal the difference of the two proportion planning values; for example, the planning value of the proportion difference could be set equal to a minimally interesting effect size.

**Usage**

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

**Arguments**

<code>alpha</code>	alpha level for hypothesis test
<code>pow</code>	desired power
<code>p1</code>	planning value of proportion for group 1
<code>p2</code>	planning value of proportion for group 2
<code>es</code>	planning value of proportion difference (effect size)

**Value**

Returns the required sample size for each group

**Examples**

```
size.test.prop2(.05, .8, .5, .5, .2)

# Should return:
# Sample size per group
#           109

size.test.prop2(.05, .8, .3, .1, .2)

# Should return:
# Sample size per group
#           71
```

size.test.sign      *Sample size for a 1-group sign test*

---

### Description

Computes the sample size required for a 1-group sign test with desired power (see size.test.sign.ps for a paired-samples sign test). The Sign test is a test of the null hypothesis that the population median is equal to some specified value. This null hypothesis can also be expressed in terms of the proportion of scores in the population that are greater than the hypothesized population median value. Under the null hypothesis, this proportion is equal to .5. This function requires a planning value of this population proportion.

### Usage

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

# Should return:
# Sample size
#           67
```

---

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

---

### Description

Computes sample size required for a paired-samples sign test with desired power. The null hypothesis can be expressed in terms of a population proportion. In a paired-samples experiment, the proportion is defined as the proportion of members in the population with scores that would be larger under treatment 1 than treatment 2. In a paired-samples nonexperiment, the proportion is the proportion of members in the population with measurement 1 scores that are larger than their measurement 2 scores. Under the null hypothesis, the proportion is equal to .5. This function requires a planning value of this population proportion.

**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
#           42
```

---

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	null hypothesis 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:
# Total sample size
#           100
```

---

slope.contrast

*Contrast coefficients for the slope of a quantitative factor*

---

**Description**

Computes the contrast coefficients that are needed to estimate the slope of a line in a one-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
```

---

spearmanbrown	<i>Computes the reliability of a scale with r2 measurements given the reliability of a scale with r1 measurements</i>
---------------	---

---

**Description**

Computes the reliability of a scale that is the sum or average of r2 parallel measurements given the reliability of a scale that is the sum or average of r1 parallel measurements. The "measurements" can be items, trials, raters, or occasions.

**Usage**

```
spearmanbrown(re1, r1, r2)
```

**Arguments**

re1	reliability of the sum or average of r1 measurements
r1	number of measurements in the original scale
r2	number of measurements in the new scale

**Value**

Returns the reliability of the sum or average of r2 measurements

**Examples**

```
spearmanbrown(.6, 10, 20)

# Should return:
# Reliability of r2 measurements
#                .75
```

---

test.anova.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 in a one-way between-subjects ANOVA using the estimated group means, estimated group standard deviations, and group sample sizes.

**Usage**

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

**Arguments**

m	vector of estimated group means
sd	vector of estimated group standard deviations
n	vector of group 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
- 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.anova.bs(m, sd, n)

# Should return:
#      F dfA dfE      p Eta-squared  adj Eta-squared
# 5.919585  2  57 0.004614428  0.1719831  0.1429298
```

---

test.cor

*Hypothesis test for a Pearson or partial correlation*


---

**Description**

Computes a t test for a test of the null hypothesis that a population Pearson or partial correlations is equal to 0, or a z test using a Fisher transformation for a test of the null hypothesis that a Pearson or partial correlation is equal to some specified nonzero value. Set  $s = 0$  for a Pearson correlation. The hypothesis testing results should be accompanied with a confidence interval for the population Pearson or partial correlation value.

**Usage**

```
test.cor(cor, n, s, h)
```



**Arguments**

cor	estimated correlation
n	sample size
s	number of control variables
h	null hypothesis value of correlation

**Value**

Returns a 1-row matrix. The columns are:

- Estimate - estimate of correlation
- t or z - t test statistic (for  $h = 0$ ) or z test statistic
- p - two-sided p-value

**See Also**

[ci.cor](#)

**Examples**

```
test.cor(.484, 100, 0, .2)

# Should return:
# Estimate      z          p
# 0.484 3.205432 0.001348601

test.cor(.372, 100, 0, 0)

# Should return:
# Estimate      t df          p
# 0.372 3.967337 98 0.000138436
```

---

test.cor2

*Hypothesis test for a 2-group Pearson or partial correlation difference*


---

**Description**

Computes a z test for a difference of population Pearson or partial correlations in a 2-group design. Set  $s = 0$  for a Pearson correlation. The hypothesis testing results should be accompanied with a confidence interval for the difference in population correlation values.

**Usage**

```
test.cor2(cor1, cor2, n1, n2, s)
```

**Arguments**

cor1	estimated correlation for group 1
cor2	estimated correlation for group 2
n1	sample size for group 1
n2	sample size for group 2
s	number of control variables

**Value**

Returns a 1-row matrix. The columns are:

- Estimate - estimate of correlation difference
- z - z test statistic
- p - two-sided p-value

**See Also**

[ci.cor2](#)

**Examples**

```
test.cor2(.684, .437, 100, 125, 0)

# Should return:
# Estimate      z      p
# 0.247 2.705709 0.006815877
```

---

test.kurtosis

*Computes p-value for test of excess kurtosis*

---

**Description**

Computes a Monte Carlo p-value (250,000 replications) for the null hypothesis that the sample data come from a normal distribution. If the p-value is small (e.g., less than .05) and excess kurtosis is positive, then the normality assumption can be rejected due to leptokurtosis. If the p-value is small (e.g., less than .05) and excess kurtosis is negative, then the normality assumption can be rejected due to platykurtosis.

**Usage**

```
test.kurtosis(y)
```

**Arguments**

y                    vector of quantitative scores

**Value**

Returns a 1-row matrix. The columns are:

- Kurtosis - estimate of kurtosis coefficient
- Excess - estimate of excess kurtosis (kurtosis - 3)
- p - Monte Carlo two-sided p-value

**Examples**

```
y <- c(30, 20, 15, 10, 10, 60, 20, 25, 20, 30, 10, 5, 50, 40, 95)
test.kurtosis(y)

# Should return:
# Kurtosis Excess      p
# 4.8149 1.8149 0.0385
```

---

test.mean	<i>Hypothesis test for a mean</i>
-----------	-----------------------------------

---

**Description**

Computes a one-sample t-test for a population mean using the estimated mean, estimated standard deviation, sample size, and null hypothesis value. Use the `t.test` function for raw data input. A confidence interval for a population mean is a recommended supplement to the t-test (see [ci.mean](#)).

**Usage**

```
test.mean(m, sd, n, h)
```

**Arguments**

m	estimated mean
sd	estimated standard deviation
n	sample size
h	null hypothesis value of mean

**Value**

Returns a 1-row matrix. The columns are:

- t - t test statistic
- df - degrees of freedom
- p - two-sided p-value

**References**

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

**Examples**

```
test.mean(24.5, 3.65, 40, 23)

# Should return:
#      t df      p
# 2.599132 39 0.01312665
```

---

test.mono.mean.bs	<i>Test of a monotonic trend in means for an ordered between-subjects factor</i>
-------------------	--

---

**Description**

Computes simultaneous confidence intervals for all adjacent pairwise comparisons of population means using estimated group means, estimated group standard deviations, and samples sizes as input. Equal variances are not assumed. A Satterthwaite adjustment to the degrees of freedom is used to improve the accuracy of the confidence intervals. If one or more lower limits are greater than 0 and no upper limit is less than 0, then conclude that the population means are monotonic decreasing. If one or more upper limits are less than 0 and no lower limits are greater than 0, then conclude that the population means are monotonic increasing. Reject the hypothesis of a monotonic trend if any lower limit is greater than 0 and any upper limit is less than 0.

**Usage**

```
test.mono.mean.bs(alpha, m, sd, n)
```

**Arguments**

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

**Value**

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

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

**Examples**

```

m <- c(12.86, 24.57, 36.29, 53.21)
sd <- c(13.185, 12.995, 14.773, 15.145)
n <- c(20, 20, 20, 20)
test.mono.mean.bs(.05, m, sd, n)

# Should return:
#   Estimate      SE      LL      UL
# 1 2   -11.71  4.139530 -22.07803 -1.3419744
# 2 3   -11.72  4.399497 -22.74731 -0.6926939
# 3 4   -16.92  4.730817 -28.76921 -5.0707936

```

---

test.mono.prop.bs	<i>Test of monotonic trend in proportions for an ordered between-subjects factor</i>
-------------------	--

---

**Description**

Computes simultaneous confidence intervals for all adjacent pairwise comparisons of population proportions using group frequency counts and samples sizes as input. If one or more lower limits are greater than 0 and no upper limit is less than 0, then conclude that the population proportions are monotonic decreasing. If one or more upper limits are less than 0 and no lower limits are greater than 0, then conclude that the population proportions are monotonic increasing. Reject the hypothesis of a monotonic trend if any lower limit is greater than 0 and any upper limit is less than 0.

**Usage**

```
test.mono.prop.bs(alpha, f, n)
```

**Arguments**

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

**Value**

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

- Estimate - estimated proportion difference
- SE - standard error
- LL - one-sided lower limit of the confidence interval
- UL - one-sided upper limit of the confidence interval

**Examples**

```
f <- c(67, 49, 30, 10)
n <- c(100, 100, 100, 100)
test.mono.prop.bs(.05, f, n)

# Should return:
#   Estimate      SE      LL      UL
# 1 2 0.1764706 0.06803446 0.01359747 0.3393437
# 2 3 0.1862745 0.06726135 0.02525219 0.3472968
# 3 4 0.1960784 0.05493010 0.06457688 0.3275800
```

---

test.prop

*Hypothesis test for a proportion*


---

**Description**

Computes a continuity-corrected z-test for a population proportion in a 1-group design. A confidence interval for a population proportion is a recommended supplement to the z-test (see [ci.prop](#)).

**Usage**

```
test.prop(f, n, h)
```

**Arguments**

f	number of participants who have the attribute
n	sample size
h	null hypothesis value of proportion

**Value**

Returns a 1-row matrix. The columns are:

- Estimate - ML estimate of proportion
- z - z test statistic
- p - two-sided p-value

**References**

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

**Examples**

```
test.prop(9, 20, .2)

# Should return:
# Estimate      z      p
#    0.45 2.515576 0.01188379
```

---

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 who have 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
#    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 population proportions in a paired-samples design. This function requires the frequency counts from a 2 x 2 contingency table for two paired dichotomous measurements. A confidence interval for a difference in population proportions (see [ci.prop.ps](#)) is a recommended supplement to the McNemar test.

**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 - two-sided 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
#    0.07 2.108346 0.03500109
```



---

`test.prop2`*Hypothesis test for a 2-group proportion difference*

---

**Description**

Computes a continuity-corrected z-test for a difference of population proportions in a 2-group design. A confidence interval for a difference in population proportions is a recommended supplement to the z-test (see [ci.prop2](#)).

**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 for group 1
n2	sample size for group 2

**Value**

Returns a 1-row matrix. The columns are:

- Estimate - ML estimate of proportion difference
- z - z test statistic
- p - two-sided 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
```

---

`test.skew`*Computes p-value for test of skewness*

---

**Description**

Computes a Monte Carlo p-value (250,000 replications) for the null hypothesis that the sample data come from a normal distribution. If the p-value is small (e.g., less than .05) and the skewness estimate is positive, then the normality assumption can be rejected due to positive skewness. If the p-value is small (e.g., less than .05) and the skewness estimate is negative, then the normality assumption can be rejected due to negative skewness.

**Usage**

```
test.skew(y)
```

**Arguments**

`y` vector of quantitative scores

**Value**

Returns a 1-row matrix. The columns are:

- Skewness - estimate of skewness coefficient
- p - Monte Carlo two-sided p-value

**Examples**

```
y <- c(30, 20, 15, 10, 10, 60, 20, 25, 20, 30, 10, 5, 50, 40, 95)
test.skew(y)

# Should return:
# Skewness      p
# 1.5201 0.0067
```

---

`test.spear`*Hypothesis test for a Spearman correlation*

---

**Description**

Computes a t test for a test of the null hypothesis that a population Spearman correlation is equal to 0, or a z test using a Fisher transformation for a test of the null hypothesis that a Spearman correlation is equal to some specified nonzero value. The hypothesis testing results should be accompanied with a confidence interval for the population Spearman correlation value.

**Usage**

```
test.spear(cor, h, n)
```

**Arguments**

cor	estimated correlation
h	null hypothesis value of correlation
n	sample size

**Value**

Returns a 1-row matrix. The columns are:

- Estimate - estimate of correlation
- t or z - t test statistic (for  $h = 0$ ) or z test statistic
- p - two-sided p-value

**See Also**

[ci.spear](#)

**Examples**

```
test.spear(.471, .2, 100)

# Should return:
# Estimate      z          p
#    0.471 3.009628 0.00261568

test.spear(.342, 0, 100)

# Should return:
# Estimate      t df          p
#    0.342 3.602881 98 0.0004965008
```

---

test.spear2

*Hypothesis test for a 2-group Spearman correlation difference*


---

**Description**

Computes a z test for a difference of population Spearman correlations in a 2-group design. The test statistic uses a Bonett-Wright standard error for each Spearman correlation. The hypothesis testing results should be accompanied with a confidence interval for a difference in population Spearman correlation values.

**Usage**

```
test.spear2(cor1, cor2, n1, n2)
```

**Arguments**

cor1	estimated Spearman correlation for group 1
cor2	estimated Spearman correlation for group 2
n1	sample size for group 1
n2	sample size for group 2

**Value**

Returns a 1-row matrix. The columns are:

- Estimate - estimate of correlation difference
- z - z test statistic
- p - two-sided p-value

**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).

**See Also**

[ci.spear2](#)

**Examples**

```
test.spear2(.684, .437, 100, 125)

# Should return:
# Estimate      z      p
# 0.247 2.498645 0.01246691
```

# Index

adj.se, 6

ci.2x2.mean.bs, 7  
ci.2x2.mean.mixed, 8  
ci.2x2.mean.ws, 10  
ci.2x2.median.bs, 11  
ci.2x2.median.mixed, 12  
ci.2x2.median.ws, 13  
ci.2x2.prop.bs, 15  
ci.2x2.prop.mixed, 16  
ci.2x2.stdmean.bs, 17  
ci.2x2.stdmean.mixed, 18  
ci.2x2.stdmean.ws, 20  
ci.agree, 21  
ci.agree.3rater, 22  
ci.agree2, 23  
ci.bayes.normal, 24  
ci.bayes.prop, 25  
ci.biphi, 26  
ci.bscor, 27  
ci.cod, 29  
ci.condslope, 30  
ci.condslope.log, 31  
ci.cor, 32, 201  
ci.cor.dep, 33  
ci.cor2, 34, 202  
ci.cor2.gen, 35  
ci.cqv, 36  
ci.cramer, 37  
ci.cronbach, 38  
ci.cronbach2, 39  
ci.cv, 40  
ci.etasqr, 41  
ci.fisher, 42  
ci.indirect, 43  
ci.kappa, 44  
ci.lc.gen.bs, 45  
ci.lc.glm, 46  
ci.lc.mean.bs, 47  
ci.lc.median.bs, 48  
ci.lc.prop.bs, 49  
ci.lc.reg, 50  
ci.lc.stdmean.bs, 51  
ci.lc.stdmean.ws, 53  
ci.mad, 54  
ci.mann, 55  
ci.mape, 56  
ci.mean, 57, 203  
ci.mean.fpc, 58  
ci.mean.ps, 59  
ci.mean1 (ci.mean), 57  
ci.mean2, 60  
ci.median, 48, 61  
ci.median.ps, 62  
ci.median1 (ci.median), 61  
ci.median2, 63  
ci.oddsratio, 64  
ci.pairs.mult, 65  
ci.pairs.prop.bs, 66  
ci.pbcor, 67  
ci.phi, 68  
ci.poisson, 69  
ci.popsiz, 70  
ci.prop, 71, 206  
ci.prop.fpc, 72  
ci.prop.inv, 73  
ci.prop.ps, 74, 208  
ci.prop1 (ci.prop), 71  
ci.prop2, 75, 209  
ci.prop2.inv, 76  
ci.pv, 77  
ci.random.anova, 78  
ci.ratio.cod2, 79  
ci.ratio.cv2, 80  
ci.ratio.mad.ps, 81  
ci.ratio.mad2, 82  
ci.ratio.mape2, 83  
ci.ratio.mean.ps, 84  
ci.ratio.mean2, 85

- ci.ratio.median.ps, 86
- ci.ratio.median2, 87
- ci.ratio.poisson2, 88
- ci.ratio.prop.ps, 89
- ci.ratio.prop2, 90
- ci.ratio.sd2, 91
- ci.rel2, 92
- ci.reliability, 93
- ci.rsqr, 94
- ci.sign, 95
- ci.slope.mean.bs, 96
- ci.slope.prop.bs, 97
- ci.spcor, 98
- ci.spear, 99, 211
- ci.spear2, 100, 212
- ci.stdmean, 40, 101
- ci.stdmean.ps, 102
- ci.stdmean.strat, 103
- ci.stdmean1 (ci.stdmean), 101
- ci.stdmean2, 104
- ci.tetra, 106
- ci.theil, 107
- ci.tukey, 108
- ci.var.upper, 109
- ci.yule, 110
  
- etasqr.adj, 111, 112
- etasqr.gen.2way, 112
  
- fitindices, 113
  
- iqv, 113
  
- pi.cor, 114
- pi.prop, 115
- pi.score, 116
- pi.score.ps, 117
- pi.score2, 118
- pi.var.upper, 119
- power.cor, 120
- power.cor2, 121
- power.lc.mean.bs, 122
- power.mean, 123
- power.mean.ps, 123
- power.mean2, 124
- power.prop, 125
- power.prop.ps, 126
- power.prop2, 127
  
- random.sample, 128
- random.y, 128
- random.yx, 129
- randomize, 130
  
- sim.ci.cor, 131
- sim.ci.mean, 132
- sim.ci.mean.ps, 133
- sim.ci.mean2, 134
- sim.ci.median, 135
- sim.ci.median.ps, 136
- sim.ci.median2, 137
- sim.ci.spear, 138
- sim.ci.stdmean.ps, 139
- sim.ci.stdmean2, 141
- size.ci.agree, 142
- size.ci.condmean, 143
- size.ci.cor, 144, 145
- size.ci.cor.prior, 145
- size.ci.cor2, 146
- size.ci.cronbach, 147
- size.ci.cronbach2, 148
- size.ci.etasqr, 149
- size.ci.indirect, 149
- size.ci.lc.ancova, 150
- size.ci.lc.mean.bs, 151
- size.ci.lc.mean.ws, 152
- size.ci.lc.prop.bs, 153
- size.ci.lc.stdmean.bs, 153
- size.ci.lc.stdmean.ws, 154
- size.ci.mape, 155
- size.ci.mean, 156, 157
- size.ci.mean.prior, 157
- size.ci.mean.ps, 158
- size.ci.mean2, 158
- size.ci.pbcor, 159
- size.ci.prop, 160, 161
- size.ci.prop.prior, 161
- size.ci.prop.ps, 162
- size.ci.prop2, 163
- size.ci.ratio.mean.ps, 163
- size.ci.ratio.mean2, 164
- size.ci.ratio.prop.ps, 165
- size.ci.ratio.prop2, 166
- size.ci.rsqr, 167
- size.ci.second, 167
- size.ci.slope, 168
- size.ci.spear, 169
- size.ci.spear2, 170
- size.ci.stdmean.ps, 171

size.ci.stdmean2, 172  
size.ci.tetra, 173  
size.equiv.mean.ps, 174  
size.equiv.mean2, 175  
size.equiv.prop.ps, 176  
size.equiv.prop2, 177  
size.interval.cor, 178  
size.supinf.mean.ps, 179  
size.supinf.mean2, 180  
size.supinf.prop.ps, 181  
size.supinf.prop2, 182  
size.test.cor, 183  
size.test.cor2, 183  
size.test.cronbach, 184  
size.test.cronbach2, 185  
size.test.lc.ancova, 186  
size.test.lc.mean.bs, 187  
size.test.lc.mean.ws, 188  
size.test.lc.prop.bs, 189  
size.test.mann, 190  
size.test.mean, 191  
size.test.mean.ps, 191  
size.test.mean2, 192  
size.test.prop, 193  
size.test.prop.ps, 194  
size.test.prop2, 195  
size.test.sign, 196  
size.test.sign.ps, 196  
size.test.slope, 197  
slope.contrast, 198  
spearmanbrown, 199

test.anova.bs, 199  
test.cor, 200  
test.cor2, 201  
test.kurtosis, 202  
test.mean, 203  
test.mono.mean.bs, 204  
test.mono.prop.bs, 205  
test.prop, 206  
test.prop.bs, 207  
test.prop.ps, 208  
test.prop2, 209  
test.skew, 210  
test.spear, 210  
test.spear2, 211