# Package ‘statpsych’

January 11, 2023

**Type** Package

**Title** Statistical Methods for Psychologists

**Version** 1.3.0

**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, mathjaxr, Rdpack, mnonr

**RoxygenNote** 7.2.1

**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** 2023-01-11 09:10:02 UTC

# R topics documented:

ci.2x2.mean.bs . . . 5
ci.2x2.mean.mixed . . . 6
ci.2x2.mean.ws . . . 7
ci.2x2.prop.bs . . . 9
ci.2x2.prop.mixed . . . 10
ci.agree . . . 11
ci.agree2 . . . 12
ci.biphi . . . 13
ci.cod1 . . . 14
ci.cod2 . . . 15
ci.condslope . . . 16
ci.condslope.log . . . 17
ci.cor . . . 18
ci.cor.dep . . . 19
ci.cor2 . . . 20
ci.cor2.gen . . . 21
ci.cramer . . . 22
ci.cronbach . . . 23
ci.etasqr . . . 24
ci.fisher . . . 25
ci.indirect . . . 25
ci.kappa . . . 26
ci.lc.gen.bs . . . 27
ci.lc.glm . . . 28
ci.lc.mean.bs . . . 30
ci.lc.median.bs . . . 31
ci.lc.prop.bs . . . 32
ci.lc.reg . . . 33
ci.lc.stdmean.bs . . . 34
ci.lc.stdmean.ws . . . 35
ci.mad1 . . . 36
ci.mann . . . 37
ci.mape . . . 38
ci.mean.ps . . . 39
ci.mean1 . . . 40
ci.mean2 . . . 41
ci.median.ps . . . 42
ci.median1 . . . 43
ci.median2 . . . 44
ci.oddsratio . . . 45
ci.pairs.prop.bs . . . 46
ci.pairs.prop1 . . . 47
ci.pbcor . . . 48
ci.phi . . . 49
ci.popsize . . . 50
ci.prop.ps . . . 51

ci.prop1 . . . . . 52
ci.prop2 . . . . . 53
ci.random.anova1 . . . . . 54
ci.ratio.mad.ps . . . . . 55
ci.ratio.mad2 . . . . . 56
ci.ratio.mean.ps . . . . . 57
ci.ratio.mean2 . . . . . 58
ci.ratio.median.ps . . . . . 59
ci.ratio.median2 . . . . . 60
ci.ratio.prop.ps . . . . . 61
ci.ratio.prop2 . . . . . 62
ci.reliability . . . . . 63
ci.rsqr . . . . . 64
ci.sign1 . . . . . 65
ci.slope.mean.bs . . . . . 66
ci.slope.prop.bs . . . . . 67
ci.spcor . . . . . 68
ci.spear . . . . . 69
ci.spear2 . . . . . 70
ci.stdmean.ps . . . . . 71
ci.stdmean.strat . . . . . 72
ci.stdmean1 . . . . . 73
ci.stdmean2 . . . . . 74
ci.tetra . . . . . 75
ci.theil . . . . . 76
ci.tukey . . . . . 77
ci.var.upper . . . . . 78
ci.yule . . . . . 79
etasqr.adj . . . . . 80
etasqr.gen.2way . . . . . 80
iqv . . . . . 81
pi.score.ps . . . . . 82
pi.score1 . . . . . 83
pi.score2 . . . . . 84
random.sample . . . . . 85
random.y . . . . . 86
random.yx . . . . . 86
randomize . . . . . 87
sim.ci.cor . . . . . 88
sim.ci.mean.ps . . . . . 89
sim.ci.mean1 . . . . . 90
sim.ci.mean2 . . . . . 91
sim.ci.median.ps . . . . . 92
sim.ci.median1 . . . . . 93
sim.ci.median2 . . . . . 94
sim.ci.spear . . . . . 95
sim.ci.stdmean.ps . . . . . 96
sim.ci.stdmean2 . . . . . 97

size.ci.agree . . . . . 98
size.ci.condmean . . . . . 99
size.ci.cor . . . . . 100
size.ci.cronbach . . . . . 101
size.ci.lc.ancova . . . . . 102
size.ci.lc.mean.bs . . . . . 103
size.ci.lc.mean.ws . . . . . 103
size.ci.lc.prop.bs . . . . . 104
size.ci.lc.stdmean.bs . . . . . 105
size.ci.lc.stdmean.ws . . . . . 106
size.ci.mean.ps . . . . . 107
size.ci.mean1 . . . . . 108
size.ci.mean2 . . . . . 108
size.ci.prop.ps . . . . . 109
size.ci.prop1 . . . . . 110
size.ci.prop2 . . . . . 111
size.ci.ratio.mean.ps . . . . . 111
size.ci.ratio.mean2 . . . . . 112
size.ci.ratio.prop.ps . . . . . 113
size.ci.ratio.prop2 . . . . . 114
size.ci.rsqr . . . . . 115
size.ci.second . . . . . 115
size.ci.slope . . . . . 116
size.ci.stdmean.ps . . . . . 117
size.ci.stdmean2 . . . . . 118
size.equiv.mean.ps . . . . . 119
size.equiv.mean2 . . . . . 120
size.equiv.prop.ps . . . . . 121
size.equiv.prop2 . . . . . 122
size.interval.cor . . . . . 123
size.supinf.mean.ps . . . . . 123
size.supinf.mean2 . . . . . 124
size.supinf.prop.ps . . . . . 125
size.supinf.prop2 . . . . . 126
size.test.cor . . . . . 127
size.test.cronbach . . . . . 128
size.test.lc.ancova . . . . . 129
size.test.lc.mean.bs . . . . . 130
size.test.lc.mean.ws . . . . . 130
size.test.lc.prop.bs . . . . . 131
size.test.mann . . . . . 132
size.test.mean.ps . . . . . 133
size.test.mean1 . . . . . 134
size.test.mean2 . . . . . 135
size.test.prop.ps . . . . . 135
size.test.prop1 . . . . . 136
size.test.prop2 . . . . . 137
size.test.sign.ps . . . . . 138

size.test.sign1 . . . . . 138
size.test.slope . . . . . 139
slope.contrast . . . . . 140
test.anova1.bs . . . . . 141
test.kurtosis . . . . . 142
test.mono.mean.bs . . . . . 143
test.mono.prop.bs . . . . . 144
test.prop.bs . . . . . 145
test.prop.ps . . . . . 146
test.prop1 . . . . . 147
test.prop2 . . . . . 148
test.skew . . . . . 149

**Index** **150**

---

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

---

## Description

Computes confidence intervals and tests for the AB interaction effect, main effect of A, main efect 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 - 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` *Computes tests and confidence intervals of effects in a 2x2 mixed design for means*

---

## Description

Computes confidence intervals and tests for the AB interaction effect, main effect of A, main efect 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 in group 1 |
| `y12` | vector of scores at level 2 in group 1 |
| `y21` | vector of scores at level 1 in group 2 |
| `y22` | vector of scores at level 2 in group 2 |

### 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 - 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:       -3.8333333 0.9803627 -3.910117 8.346534 0.0041247610 -6.0778198 -1.588847
# A:         2.0833333 0.4901814  4.250128 8.346534 0.0025414549  0.9610901  3.205577
# B:         3.7500000 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 efect 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 - 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.prop.bs — *Computes tests and confidence intervals of effects in a 2x2 between-subjects design for proportions*

### Description

Computes adjusted Wald confidence intervals and tests for the AB interaction effect, main effect of A, main efect 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 f11, f12, f21, f22, and the input vector of sample sizes is n11, n12, n21, n22 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 with 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 of estimate
- z - z test statistic for test of null hypothesis
- p - p-value
- LL - lower limit of the adjusted Wald confidence interval
- UL - upper limit of the adjusted Wald confidence interval

### 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` *Computes tests and confidence intervals of effects in a 2x2 mixed factorial design for proportions*

---

## Description

Computes adjusted Wald confidence intervals and tests for the AB interaction effect, main effect of A, main efect 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 fij 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` | 2x2 contingency table for Factor A in group 1 |
| `group2` | 2x2 contingency table for Factor A 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 - p-value
- LL - lower limit of the adjusted Wald confidence interval
- UL - upper limit of the adjusted Wald confidence interval

### Examples

```
group1 = c(23, 42, 24, 11)
group2 = c(26, 27, 13, 34)
ci.2x2.prop.mixed (.05, group1, group2)

# Should return:
#             Estimate         SE         z           p           LL        UL
# AB:       0.03960396 0.09991818 0.3963639 0.691836584 -0.156232072 0.2354400
# A:        0.15841584 0.04995909 3.1709113 0.001519615  0.060497825 0.2563339
# B:        0.09803922 0.04926649 1.9899778 0.046593381  0.001478675 0.1945998
# A at b1: 0.17647059 0.07893437 2.2356621 0.025373912  0.021762060 0.3311791
# A at b2: 0.13725490 0.06206620 2.2114274 0.027006257  0.015607377 0.2589024
# B at a1: 0.11764706 0.06842118 1.7194539 0.085531754 -0.016455982 0.2517501
# B at a2: 0.07843137 0.06913363 1.1344894 0.256589309 -0.057068054 0.2139308
```

---

## ci.agree — *Confidence interval for a G-index of agreement*

---

### Description

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

### Usage

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

### Arguments

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

### Value

Returns a 1-row matrix. The columns are:

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

## Examples

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

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

---

ci.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. The point estimates are maximum likelihood estimates.

## 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 - estimate of G-index (single-group and difference)
- LL - lower limit of confidence interval
- UL - upper limit of confidence interval

### Examples

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

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

---

## ci.biphi *Confidence interval for a biserial-phi correlation*

---

### Description

Computes a confidence interval for a 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.

## Examples

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

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

---

ci.cod1 *Confidence interval for a single coefficient of dispersion*

---

## Description

Computes a confidence interval for a population coefficient of dispersion 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.

## Usage

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

## Arguments

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

## Value

Returns a 1-row matrix. The columns are:

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

## References

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

### Examples

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

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

---

## ci.cod2 — *Confidence interval for a ratio of dispersion coefficients in a 2-group design*

---

### Description

Computes a confidence interval for a ratio of population dispersion coefficients (MAD/median) in a 2-group design. Ratio-scale scores are assumed.

### Usage

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

### Arguments

| | |
|---|---|
| alpha | alpha level for 1-alpha confidence |
| y1 | vector of scores in gorup 1 |
| y2 | vector of scores in gorup 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.cod2(.05, y1, y2)

# Should return:
#           COD1      COD2 COD1/COD2       LL       UL
# [1,] 0.1333333 0.1232558  1.081761 0.494964 2.282254
```

---

`ci.condslope` *Confidence interval for conditional (simple) slopes in a linear model*

---

## Description

Computes confidence intervals and test statistics for population conditional slopes (simple slopes) in a general linear model that includes a predictor variable that is the product of a moderator variable and a predictor variable. Conditional slopes are computed at 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 condition slope
- t - t test statistic
- p - p-value
- LL - lower limit of the confidence interval
- UL - upper limit of the confidence interval

## Examples

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

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

---

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

---

## Description

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

## Usage

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

## Arguments

| | |
|---|---|
| `alpha` | alpha level for 1-alpha confidence |
| `b1` | 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 condition slope
- exp(Estimate) - estimated exponentiated condition slope
- z - z test statistic
- p - p-value
- LL - lower limit of the confidence interval
- UL - upper limit of the confidence interval

### Examples

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

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

---

## ci.cor — *Confidence interval for a Pearson or partial correlation*

---

### Description

Computes a Fisher confidence interval for a population Pearson correlation or partial correlation with s control variables. Set s = 0 for a Pearson correlation. A bias adjustmentment 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
# [1,]    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 adjustmentment is used to reduce the bias of each Fisher transformed correlation.

## 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
- 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.

### Examples

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

# Should return:
#      Estimate         LL      UL
# [1,]    0.217 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 adjustmentment 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 in group 1 |
| cor2 | estimated Pearson correlation 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 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.

## Examples

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

# Should return:
#      Estimate         LL        UL
# [1,]    0.084 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 function requires 100(1 - alpha)% confidence intervals for each correlation as input. This function also can be used to compute a confidence interval for the difference of two Cronbach reliability coefficients.

## Usage

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

## Arguments

| | |
|---|---|
| `cor1` | 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.

### Examples

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

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

---

## 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 x 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 for independence |
| `r` | number of rows in contingency table |
| `c` | number of columns in contengency table |
| `n` | sample size |

### Value

Returns a 1-row matrix. The columns are:

- Estimate - estimate of Cramer's V
- SE - approximate 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
# [1,]    0.3099 0.0674 0.1888 0.4529
```

---

`ci.cronbach` *Confidence interval for a Cronbach reliability*

---

## Description

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

## Usage

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

## Arguments

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

## Value

Returns a 1-row matrix. The columns are:

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

## References

Feldt LS (1965). "The approximate sampling distribution of Kuder-Richardson reliability coefficient twenty." *Psychometrika*, **30**(3), 357–370. ISSN 0033-3123, doi: 10.1007/BF02289499.

### Examples

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

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

---

## 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 also computed.

### 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 - estimate of eta-squared
- adj Eta-squared - bias adjusted eta-squared estimate
- 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        LL        UL
# [1,]       0.241       0.2213707 0.1040229 0.3493431
```

## ci.fisher *Fisher confidence interval*

### Description

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

### Usage

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

### Arguments

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

### Value

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

### Examples

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

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

## ci.indirect *Confidence interval for an indirect effect*

### Description

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

## Usage

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

## Arguments

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

## Value

Returns a 1-row matrix. The columns are:

- Estimate - estimated indirect effect
- SE - standard error of indirect effect
- 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
# [1,]   4.7616 1.625282 2.178812 7.972262
```

---

`ci.kappa` *Confidence interval for a kappa reliability*

---

## Description

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

## Usage

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

### Arguments

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

### Value

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

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

### References

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

### Examples

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

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

---

## ci.lc.gen.bs — *Confidence interval for a linear contrast of parameters in a between-subjects design*

---

### Description

Computes the estimate, standard error, and approximate confidence interval for a linear contrast of any type of parameter (e.g., quartile, ordinal regression slope, path coefficient, G-index) where each parameter value has been estimated from a different sample. The parameter vaues 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 simultaneous 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,]     1.63 0.2573806 1.125543 2.134457
```

---

## `ci.lc.glm` — *Confidence interval for a linear contrast of general linear model parameters*

---

### 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 coef(object) |
| `V` | covariance matrix of parameter estimates from vcov(object) |
| `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
# > ci.lc.glm(.05, n, b, V, q)
#      Estimate        SE       t df           p       LL       UL
# [1,] 2.931111 0.4462518 6.56829  7 0.000313428 1.875893 3.986329
```

`ci.lc.mean.bs` *Confidence interval for a linear contrast of means in a between-subjects design*

## 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 betwen-subjects contrast coefficients |

## Value

Returns a 2-row matrix. The columns are:

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

## References

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

### Examples

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

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

---

## ci.lc.median.bs — *Confidence interval for a linear contrast of medians in a between-subjects design*

---

### Description

Computes a 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.median1 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
# [1,]    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 proportions in a betwen-subjects design.

## Usage

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

## Arguments

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

## Value

Returns a 1-row matrix. The columns are:

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

## References

Price RM, Bonett DG (2004). "An improved confidence interval for a linear function of binomial proportions." *Computational Statistics & Data Analysis*, **45**(3), 449–456. ISSN 01679473, doi: 10.1016/S01679473(03)000070.

## Examples

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

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

---

`ci.lc.reg` *Confidence interval for a linear contrast of regression coefficients in multiple group regression model*

---

## Description

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

## Usage

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

## Arguments

| | |
|---|---|
| `alpha` | alpha level for 1-alpha confidence |
| `est` | vector of parameter estimates |
| `se` | vector of standard errors |
| `n` | vector of group sample sizes |
| `s` | number of predictor variables for 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,]    1.303 0.5085838 2.562016 78.8197 0.01231256 0.2906532 2.315347
```

---

## ci.lc.stdmean.bs — *Confidence interval for a standardized linear contrast of means in a between-subjects design*

---

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

### References

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

### Examples

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

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

---

## ci.lc.stdmean.ws — *Confidence interval for a standardized linear contrast of means in a within-subjects design*

---

### 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 within-subjects estimated means |
| sd | vector of within-subjects estimated standard deviations |
| 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 - bias adjusted standardized linear contrast
- SE - standard error
- LL - lower limit of the confidence interval
- UL - upper limit of the confidence interval

## References

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

## Examples

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

# Should return:
#                            Estimate        SE        LL         UL
# Unweighted standardizer: -1.266557 0.2096351 -1.712140 -0.8903860
# Level 1 standardizer:    -1.337500 0.2662156 -1.915002 -0.8714561
```

---

ci.mad1 *Confidence interval for a single MAD*

---

## 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.mad1(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
- 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.

### Examples

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

# Should return:
#       Estimate       LL       UL
# [1,]      12.5 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 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

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

### Arguments

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

### Value

Returns a 1-row matrix. The columns are:

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

### References

Sen PK (1967). "A note on asymptotically distribution-free confidence bounds for 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
# [1,]    0.795 0.1401834 0.5202456  1
```

---

`ci.mape` *Confidence interval for a mean absolute prediction error*

---

### Description

Computes a confidence interval for a population mean absolute prediction error (MAPE) in a general linear model. The MAPE is a 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
- 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       LL       UL
# [1,]    2.3744 1.751678 3.218499
```

---

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

---

### Description

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

### Usage

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

### Arguments

| | |
|---|---|
| alpha | alpha level for 1-alpha confidence |
| m1 | 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 - p-value
- LL - lower limit of the confidence interval
- UL - upper limit of the confidence interval

### Examples

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

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

---

## ci.mean1 *Confidence interval for a single mean*

---

### Description

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

### Usage

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

### Arguments

| | |
|---|---|
| alpha | alpha level for 1-alpha confidence |
| m | 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.mean1(.05, 24.5, 3.65, 40)

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

---

`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. Use the t.test function for raw data input.

## Usage

```
ci.mean2(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 mean difference
- SE - standard error
- t - t test statistic
- df - degrees of freedom
- p - p-value
- LL - lower limit of the confidence interval
- UL - upper limit of the confidence interval

### References

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

### Examples

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

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

---

## ci.median.ps — *Confidence interval for a paired-samples median difference*

---

### Description

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

### 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
# [1,]    13      30             -17 3.362289 -23.58996 -10.41004
#        SE1      SE2      COV
#   3.085608 4.509735 9.276849
```

---

ci.median1 *Confidence interval for a single median*

---

## Description

Computes a confidence interval for a single population median.

## Usage

```
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.median1(.05, y)

# Should return:
#      Estimate       SE LL UL
# [1,]       20 4.270922 10 30
```

---

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

---

## Description

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

## References

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

## Examples

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

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

---

`ci.oddsratio` *Confidence interval for an odds ratio*

---

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

### References

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

### Examples

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

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

---

`ci.pairs.prop.bs` *Bonferroni confidence intervals for all pairwise proportion differences in a between-subjects design*

---

### Description

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

### Usage

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

### Arguments

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

### Value

Returns a 1-row matrix. The columns are:

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

### References

Agresti A, Caffo B (2000). "Simple and effective confidence intervals for proportions and differences of proportions result from adding two successes and two failures." *The American Statistician*, **54**(4), 280. ISSN 00031305, doi: 10.2307/2685779.

### 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  0.01175409
# 2 3   0.1435644 0.04358401  3.293968 9.878366e-04  0.03922511  0.24790360
```

---

## ci.pairs.prop1 — *Confidence intervals for pairwise proportion differences of a polychotomous variable*

---

### Description

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

### Usage

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

### Arguments

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

### Value

Returns a 1-row matrix. The columns are:

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

## References

Bonett DG, Price RM (2012). "Adjusted wald confidence interval for a difference of binomial proportions based on paired data." *Journal of Educational and Behavioral Statistics*, **37**(4), 479–488. ISSN 1076-9986, doi: 10.3102/1076998611411915.

## Examples

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

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

---

ci.pbcor *Confidence interval for a point-biserial correlation*

---

## Description

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

## Usage

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

## Arguments

| | |
|---|---|
| alpha | alpha level for 1-alpha confidence |
| m1 | 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
- 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        LL        UL
# Weighted:   0.7065799 0.5885458 0.7854471
# Unweighted: 0.7020871 0.5808366 0.7828948
```

---

`ci.phi` *Confidence interval for a phi correlation*

---

### Description

Computes a confidence interval for a 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
# [1,] 0.1229976 0.05746271 0.01037273 0.2356224
```

---

## ci.popsize *Confidence interval for an unknown population size*

---

### Description

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

### Usage

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

### Arguments

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

### Value

Returns a 1-row matrix. The columns are:

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

### Examples

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

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

---

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

---

### Description

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

### Examples

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

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

---

## ci.prop1 — *Confidence interval for a single proportion*

---

### Description

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

### Usage

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

### Arguments

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

### Value

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

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

The columns of row 2 are:

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

## References

Agresti A, Coull BA (1998). "Approximate is better than 'exact' for interval estimation of binomial proportions." *The American Statistician*, **52**(2), 119–126. ISSN 0003-1305, doi: 10.1080/00031305.1998.10480550.

## Examples

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

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

---

`ci.prop2` *Confidence interval for a 2-group proportion difference*

---

## Description

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

## Usage

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

## Arguments

| | |
|---|---|
| `alpha` | alpha level for 1-alpha confidence |
| `f1` | number of participants in group 1 who have the attribute |
| `f2` | number of participants in group 2 who have the attribute |
| `n1` | sample size 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. ISSN 00031305, doi: 10.2307/2685779.

### Examples

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

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

---

`ci.random.anova1` *Confidence intervals for parameters of one-way random effects ANOVA*

---

### Description

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

### Usage

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

### Arguments

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

### Value

Returns a 4-row matrix. The rows are:

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

The columns are:

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

## Examples

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

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

---

`ci.ratio.mad.ps` *Confidence interval for a paired-sample MAD ratio*

---

## Description

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

## Usage

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

## Arguments

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

## Value

Returns a 1-row matrix. The columns are:

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

### References

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

### Examples

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

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

---

`ci.ratio.mad2` *Confidence interval for a 2-group MAD ratio*

---

### Description

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

### Usage

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

### Arguments

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

### Value

Returns a 1-row matrix. The columns are:

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

## References

Bonett DG, Seier E (2003). "Confidence intervals for mean absolute deviations." *The American Statistician*, **57**(4), 233–236. ISSN 0003-1305, doi: 10.1198/0003130032323.

## Examples

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

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

---

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

---

## Description

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

## Usage

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

## Arguments

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

## Value

Returns a 1-row matrix. The columns are:

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

## References

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

## Examples

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

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

---

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

---

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

### References

Bonett DG, Price RM (2020). "Confidence intervals for ratios of means and medians." *Journal of Educational and Behavioral Statistics*, **45**(6), 750–770. ISSN 1076-9986, doi: 10.3102/1076998620934125.

### Examples

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

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

---

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

---

### Description

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

### Usage

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

### Arguments

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

### Value

Returns a 1-row matrix. The columns are:

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

## References

Bonett DG, Price RM (2020). "Confidence intervals for ratios of means and medians." *Journal of Educational and Behavioral Statistics*, **45**(6), 750–770. ISSN 1076-9986, doi: 10.3102/1076998620934125.

## Examples

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

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

---

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

---

## Description

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

## Usage

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

## Arguments

| | |
|---|---|
| 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. ISSN 1076-9986, doi: 10.3102/1076998620934125.

### Examples

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

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

---

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

---

### Description

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

### Usage

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

### Arguments

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

### Value

Returns a 1-row matrix. The columns are:

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

### References

Bonett DG, Price RM (2006). "Confidence intervals for a ratio of binomial proportions based on paired data." *Statistics in Medicine*, **25**(17), 3039–3047. ISSN 0277-6715, doi: 10.1002/sim.2440.

### Examples

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

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

---

## ci.ratio.prop2 — *Confidence interval for a 2-group proportion ratio*

---

### Description

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

### Usage

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

### Arguments

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

### Examples

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

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

---

## ci.reliability *Confidence interval for a reliability coefficient*

---

### Description

Computes a confidence interval for a population reliability coefficient such as Cronbach's alpha or McDonald's omega using an estimate of the reliablity 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:

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

### Examples

```
ci.reliability(.05, .88, .147, 100)

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

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.

## 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
- adj R-squared - bias adjusted R-squared estimate
- 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.

## Examples

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

# Should return:
#         R-squared    adj R-squared          LL        UL
# [1,]       0.241        0.2206696  0.09819599 0.3628798
```

## ci.sign1 — *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.sign1(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.sign1(.05, y, 9)

# Should return:
#      Estimate        SE        LL        UL
# [1,] 0.826087 0.0790342 0.6711828 0.9809911
```

ci.slope.mean.bs *Confidence interval for the slope of means in a single-factor design with a quantitative between-subjects factor*

## Description

Computes a test statistic and confidence interval for the slope of means in a single-factor 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 - 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` *Confidence interval for a slope of a proportion in a single-factor design with a quantitative between-subjects factor*

---

## Description

Computes a test statistic and an adjusted Wald confidence interval for the slope of proportions in a single-factor 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 - 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.

## Examples

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

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

---

ci.spcor *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
- 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.

### Examples

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

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

---

`ci.spear` *Confidence interval for a Spearman correlation*

---

### 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 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.

## Examples

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

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

---

`ci.spear2` *Confidence interval for a 2-group Spearman correlation difference*

---

## Description

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

## Usage

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

## Arguments

| | |
|---|---|
| `alpha` | alpha level for 1-alpha confidence |
| `cor1` | 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
- 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.

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.

## Examples

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

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

---

`ci.stdmean.ps` *Confidence interval for a paired-samples standardized mean difference*

---

## 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 of measurement 1 |
| `m2` | estimated mean of measurement 2 |
| `sd1` | estimated standard deviation of measurement 1 |
| `sd2` | estimated standard deviation of measurement 2 |
| `cor` | estimated correlation between measurements |
| `n` | sample size |

## Value

Returns a 3-row matrix. The columns are:

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

## References

Bonett DG (2008). “Confidence intervals for standardized linear contrasts of means.” *Psychological Methods*, **13**(2), 99–109. ISSN 1939-1463, doi: 10.1037/1082989X.13.2.99.

### Examples

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

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

---

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

---

### Description

Computes confidence intervals for a population standardized mean difference in a 2-group nonexperimental design with stratified random sampling (a random sample of a specificied 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 - bias adjusted standardized mean difference
- SE - standard error
- LL - lower limit of the confidence interval
- UL - upper limit of the confidence interval

## References

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

## Examples

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

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

---

ci.stdmean1 *Confidence interval for a single standardized mean*

---

## Description

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

## Usage

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

## Arguments

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

## Value

Returns a 1-row matrix. The columns are:

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

### References

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

### Examples

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

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

---

## ci.stdmean2 — *Confidence interval 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 - bias adjusted standardized mean difference
- SE - standard error
- LL - lower limit of the confidence interval
- UL - upper limit of the confidence interval

### References

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

### Examples

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

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

---

## ci.tetra *Confidence interval for a tetrachoric correlation*

---

### Description

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

### Usage

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

### Arguments

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

### Value

Returns a 1-row matrix. The columns are:

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

## References

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

## Examples

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

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

---

`ci.theil` *Theil-Sen estimate and confidence interval for slope*

---

## Description

Computes a Theil-Sen estimate and distribution-free confidence interval for the slope of 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 - approximate 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
# [1,]      0.5 0.1085927 0.3243243 0.75
```

---

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

### 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 *Upper confidence limit of variance*

---

### Description

Computes an upper 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 that require a variance planning value.

### 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
# [1,] 17.23264
```

---

`ci.yule` *Confidence interval for Yule's Q*

---

## Description

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

## Usage

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

## Arguments

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

## Value

Returns a 1-row matrix. The columns are:

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

## Examples

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

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

## etasqr.adj *Bias adjusts 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:
#      Adjusted eta-squared
# [1,]             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. Use the estimates from this function 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 classiciation: 0.300000 0.3804878 0.1268293
```

---

## iqv *Indices of qualitative variation*

---

### Description

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

### Usage

```
iqv(f)
```

### Arguments

| | |
|---|---|
| f | vector of multinomial frequency counts |

### Value

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

## Examples

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

# Should return:
#        Simpson    Berger   Shannon
# [1,] 0.7367908 0.5045045       0.7
```

---

pi.score.ps — *Prediction interval for difference of scores in a 2-level within-subjects experiment*

---

## 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 from group 1 |
| m2 | estimated mean from group 2 |
| sd1 | estimated standard deviation from group 1 |
| sd2 | estimated standard deviation from group 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
# [1,]      56.5 29 28.05936 84.94064
```

---

pi.score1 *Prediction interval for one score*

---

## Description

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

## Usage

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

## Arguments

| | |
|---|---|
| alpha | alpha level for 1-alpha confidence |
| m | 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.score1(.05, 24.5, 3.65, 40)

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

`pi.score2` *Prediction interval for a difference of scores 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

| | |
|---|---|
| `alpha` | alpha level for 1-alpha confidence |
| `m1` | estaimted mean for group 1 |
| `m2` | estimated mean for group 1 |
| `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:

- 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.

### Examples

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

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

---

`random.sample` *Generate a random sample*

---

### Description

Generates a random sample of participant IDs.

### Usage

```
random.sample(popsize, samsize)
```

### Arguments

| | |
|---|---|
| `popsize` | study population size |
| `samsize` | 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 functions 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 | vector of sample sizes per group |

### Value

Returns a vector of randomly generated group assignments.

### Examples

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

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

---

## sim.ci.cor — *Simulates confidence interval coverage probability for a Pearson correlation*

---

### 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)<br>• 1 = Gaussian (skewness = 0 and excess kurtosis = 0)<br>• 2 = platykurtic (skewness = 0 and excess kurtosis = -1.2)<br>• 3 = leptokurtic (skewness = 0 and excess kurtsois = 6)<br>• 4 = moderate skew (skewness = 1 and excess kurtosis = 1.5)<br>• 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 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.ps    *Simulates confidence interval coverage probability for a paired-samples mean difference*

---

### Description

Performs a computer simulation of confidence interval performance for a mean difference in a paired-samples design. Sample data within each level 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)<br>• 1 = Gaussian (skewness = 0 and excess kurtosis = 0)<br>• 2 = platykurtic (skewness = 0 and excess kurtosis = -1.2)<br>• 3 = leptokurtic (skewness = 0 and excess kurtsois = 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 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
# [1,]  0.93815     0.05125      0.0106    0.7778518
```

---

## sim.ci.mean1 — *Simulates confidence interval coverage probability for single mean*

---

### Description

Performs a computer simulation of the confidence interval performance for a single 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.mean1(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)

- 1 = Gaussian (skewness = 0 and excess kurtosis = 0)
- 2 = platykurtic (skewness = 0 and excess kurtosis = -1.2)
- 3 = leptokurtic (skewness = 0 and excess kurtsois = 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 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.mean1(.05, 40, 4, 1000)

# Should return (within sampling error):
#      Coverage Lower Error Upper Error Ave CI Width
# [1,]  0.94722     0.01738      0.0354    0.6333067
```

---

## sim.ci.mean2 — *Simulates confidence interval coverage probability for a 2-group mean difference*

---

### Description

Performs a computer simulation of separate variance and pooled variance confidence interval performance for a 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)<br>• 1 = Gaussian (skewness = 0 and excess kurtosis = 0)<br>• 2 = platykurtic (skewness = 0 and excess kurtosis = -1.2)<br>• 3 = leptokurtic (skewness = 0 and excess kurtsois = 6)<br>• 4 = moderate skew (skewness = 1 and excess kurtosis = 1.5)<br>• 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 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.ps *Simulates confidence interval coverage probability for a median difference in a paired-samples design*

---

### Description

Performs a computer simulation of confidence interval performance for a median difference in a paired-samples design. Sample data within each level 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) |

- 1 = Gaussian (skewness = 0 and excess kurtosis = 0)
- 2 = platykurtic (skewness = 0 and excess kurtosis = -1.2)

- 3 = leptokurtic (skewness = 0 and excess kurtsois = 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, 2000)

# Should return (within sampling error):
#      Coverage Lower Error Upper Error Ave CI Width
# [1,]    0.961       0.026       0.013    0.9435462
```

---

## sim.ci.median1 — *Simulates confidence interval coverage probability for single median*

### Description

Performs a computer simulation of the confidence interval performance for a single 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.median1(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)

- 1 = Gaussian (skewness = 0 and excess kurtosis = 0)
- 2 = platykurtic (skewness = 0 and excess kurtosis = -1.2)
- 3 = leptokurtic (skewness = 0 and excess kurtsois = 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
- Lower Error - probability of lower limit greater than population mediann
- Upper Error - probability of upper limit less than population median
- Ave CI Width - average confidence interval width

### Examples

```
sim.ci.median1(.05, 20, 5, 1000)

# Should return (within sampling error):
#      Coverage Lower Error Upper Error Ave CI Width
# [1,]   0.9589      0.0216      0.0195    0.9735528
```

---

`sim.ci.median2` *Simulates confidence interval coverage probability for a median difference in a 2-group design*

---

### Description

Performs a computer simulation of the confidence interval performance for a difference of 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)<br>• 1 = Gaussian (skewness = 0 and excess kurtosis = 0)<br>• 2 = platykurtic (skewness = 0 and excess kurtosis = -1.2)<br>• 3 = leptokurtic (skewness = 0 and excess kurtsois = 6)<br>• 4 = moderate skew (skewness = 1 and excess kurtosis = 1.5)<br>• 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
# [1,]    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)<br>• 1 = Gaussian (skewness = 0 and excess kurtosis = 0)<br>• 2 = platykurtic (skewness = 0 and excess kurtosis = -1.2)<br>• 3 = leptokurtic (skewness = 0 and excess kurtsois = 6)<br>• 4 = moderate skew (skewness = 1 and excess kurtosis = 1.5)<br>• 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 correlation
- Lower Error - probability of lower limit greater than population correlation
- Upper Error - probability of upper limit less than population corrrelation
- 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` *Simulates confidence interval coverage probability for a standardized mean difference in a paired-samples design*

---

## Description

Performs a computer simulation of confidence interval performance for two types of standardized mean differences in a paired-samples design. Sample data for each level 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)<br>• 1 = Gaussian (skewness = 0 and excess kurtosis = 0)<br>• 2 = platykurtic (skewness = 0 and excess kurtosis = -1.2)<br>• 3 = leptokurtic (skewness = 0 and excess kurtsois = 6) |

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

| | |
|---|---|
| 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` *Simulates confidence interval coverage probability for a standardized mean difference in a 2-group design*

---

### Description

Performs a computer simulation of confidence interval performance for two types of standardized mean differences 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.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) |

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

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

### Usage

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

### Arguments

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

### Value

Returns the required sample size

### Examples

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

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

---

## size.ci.condmean *Sample size for a conditional mean confidence interval*

---

### Description

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

### Usage

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

### Arguments

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

### Value

Returns the required total sample size

## Examples

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

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

---

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

---

## Description

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

## Usage

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

## Arguments

| | |
|---|---|
| alpha | alpha level for 1-alpha confidence |
| cor | planning value of correlation |
| s | number of control variables |
| 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.cor(.05, .362, 0, .25)

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

size.ci.cronbach *Sample size for a Cronbach reliability confidence interval*

## Description

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

## 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.

## Examples

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

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

## 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 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 per group

### Examples

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

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

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

### Description

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

### Usage

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

### Arguments

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

### Value

Returns the required sample size for each group

### Examples

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

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

## size.ci.lc.mean.ws — *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
# [1,]          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 proportions with desired confidence interval precision in a between-subjects design. Set the proportion planning values to .5 for a conservatively large sample size.

### Usage

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

### Arguments

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

## Value

Returns the required sample size per group

## Examples

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

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

---

`size.ci.lc.stdmean.bs` *Sample size for a between-subjects standardized 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.

## 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.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
# [1,]          19
```

size.ci.mean1 *Sample size for a single mean confidence interval*

## Description

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

## Usage

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

## Arguments

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

## Value

Returns the required sample size

## Examples

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

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

size.ci.mean2 *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
# [1,] 47  47
```

---

`size.ci.prop.ps` *Sample size for a paired-sample proportion difference confidence interval*

---

### Description

Computes the sample size required to estimate a proportion difference with desired confidence interval precision in a paired-samples design. Set the proportion planning values to .5 for a conservatively large sample size. Set the phi 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 group 1 |
| p2 | planning value of proportion for group 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
# [1,]         118
```

---

## size.ci.prop1 — *Sample size for a single proportion confidence interval*

---

### Description

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

### Usage

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

### Arguments

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

### Value

Returns the required sample size

### Examples

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

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

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

### Description

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

### Usage

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

### Arguments

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

### Value

Returns the required sample size per group

### Examples

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

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

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

### Description

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

## Usage

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

## Arguments

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

## Value

Returns the required sample size

## Examples

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

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

---

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

---

## Description

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

## Usage

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

### Arguments

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

### Value

Returns the required sample size for each group

### Examples

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

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

---

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

---

### Description

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

### Usage

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

### Arguments

| | |
|---|---|
| alpha | alpha level for 1-alpha confidence |
| p1 | planning value of proportion for measurement 1 |
| p2 | planning value of proportion for measurement 2 |
| phi | planning value of phi 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
# [1,]          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 proportions with desired confidence interval precision in a 2-group design.

### Usage

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

### Arguments

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

### Value

Returns the required sample size per group

### Examples

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

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

size.ci.rsqr *Sample size for a squared multiple correlation confidence interval*

## Description

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

## Usage

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

## Arguments

| | |
|---|---|
| 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, .333, 2, .2)

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

size.ci.second *Sample size for a second-stage confidence interval*

## Description

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

### Usage

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

### Arguments

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

### Value

Returns the required sample size for the second-stage sample

### Examples

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

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

---

## size.ci.slope *Sample size for a slope confidence interval*

---

### Description

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

### Usage

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

### Arguments

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

### Value

Returns the required total sample size

### Examples

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

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

---

## size.ci.stdmean.ps — *Sample size for a paired-samples standardized mean difference confidence interval*

---

### 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. For a conservatively large sample size, 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.

### 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/a0014270.0

## 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 each 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.

### Examples

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

# Should return:
#                                n1  n2
# Unweighted standardizer:    132 132
# Single group standardizer:  141 141
```

---

## 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
# [1,]          68
```

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

## Description

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

## Usage

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

## Arguments

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

## Value

Returns the required sample size per group

## Examples

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

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

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

### Description

Computes the sample size 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 x alpha rather than alpha (e.g., use alpha = .10 rather 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 coefficient |
| `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
# [1,]         173
```

# size.equiv.prop2 *Sample size for a 2-group proportion equivalence test*

## Description

Computes the sample size in each group (assuming equal sample sizes) required to perform an equivalence test for the difference in population 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 x alpha rather than alpha (e.g., use alpha = .10 rather than alpha = .05).

## Usage

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

## Arguments

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

## Value

Returns the required sample size per group

## Examples

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

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

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

### Description

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

### Usage

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

### Arguments

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

### Value

Returns the required sample size

### Examples

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

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

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

### Description

Computes the sample size required to perform a superiority or noninferiority test for the difference in population means with desired power in a paired-samples design. For a superiority test, specify the upper limit (h) for the range of practical equivalence and specify an effect size (es) such that es > h. For a noninferiority test, specify the lower limit (-h) for the range of practical equivalence and specify an effect size such that es > -h. Set the 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
# [1,]          38
```

---

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

---

### Description

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

### Usage

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

### Arguments

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

### Value

Returns the required sample size per group

### Examples

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

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

---

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

---

### Description

Computes the sample size required to perform a superiority or inferiority test for the difference in population proportions with desired power in a paired-samples design. For a superiority test, specify the upper limit (h) for the range of practical equivalence and specify values of p1 and p2 such that p1 - p2 > h. For an inferiority test, specify the lower limit (-h) for the range of practical equivalence and specify values of p1 and p2 such that p1 - p2 > -h. This function sets the effect size equal to the difference in proportion planning values. Set the phi 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
# [1,]         227
```

---

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

---

## Description

Computes the sample size in each group (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 per group

## Examples

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

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

---

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

---

## Description

Computes the sample size required to test a Pearson or a partial correlation with desired power. Set s = 0 for a Pearson correlation.

## Usage

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

## Arguments

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

## Value

Returns the required sample size

## Examples

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

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

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

## Description

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

## Usage

```
size.test.cronbach(alpha, pow, rel, 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
# [1,]         139
```

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

### Description

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

### Usage

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

### Arguments

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

### Value

Returns the required sample size per group

### Examples

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

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

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

### Description

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

### Usage

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

### Arguments

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

### Value

Returns the required sample size per group

### Examples

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

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

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

### Description

Computes the sample size required to test a linear contrast of population means with desired power in a within-subjects design. Set the 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
# [1,]          29
```

---

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

---

### Description

Computes the sample size in each group (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 per group

### Examples

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

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

---

## `size.test.mann` *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

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

### Value

Returns the required sample size per group

### References

Noether GE (1987). "Sample size determination for some common nonparametric tests." *Journal of the American Statistical Association*, **82**(398), 645–647. ISSN 0162-1459, doi: 10.1080/01621459.1987.10478478.

### Examples

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

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

---

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

---

### Description

Computes the sample size required to test a difference in population means with desired power in a paired-samples design. Set the 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
# [1,]          22
```

---

size.test.mean1 *Sample size for a test of a single mean*

---

## Description

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

## Usage

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

## Arguments

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

## Value

Returns the required sample size

## Examples

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

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

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

### Description

Computes the sample size in each group 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 per group

### Examples

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

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

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

### Description

Computes the sample size required to test a difference in population proportions with desired power in a paired-samples design. This function requires planning values for both proportions and a phi coefficient that describes the correlation between the two dichotomous measurements. The proportion planning values can set to .5 for a conservatively large sample size. The planning value for the 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
# [1,]         177
```

---

## size.test.prop1 *Sample size for a test of a single proportion*

---

### Description

Computes the sample size required to test a single population proportion with desired power in a 1-group design. Set the proportion planning value to .5 for a conservatively large sample size. The value of the effect size (expected population proportion minus hypothesized value) need not be based on the proportion planning value.

### Usage

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

### Arguments

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

## Value

Returns the required sample size

## Examples

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

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

---

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

---

## Description

Computes the sample size in each group required to test a difference in population proportions with desired power in a 2-group design. This function requires planning values for both proportions. Set the proportion planning values to .5 for a conservatively large sample size. The planning value for the 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.

## Usage

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

## Arguments

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

## Value

Returns the required sample size per group

## Examples

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

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

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

### Description

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

### Usage

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

### Arguments

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

### Value

Returns the required sample size

### Examples

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

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

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

### Description

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

### Usage

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

### Arguments

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

### Value

Returns the required sample size

### Examples

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

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

---

## size.test.slope    *Sample size for a test of a slope*

---

### Description

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

### Usage

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

### Arguments

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

### Value

Returns the required total sample size

### Examples

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

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

---

## slope.contrast *Contrast coefficients for the slope of a quantitative factor*

---

### Description

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

### Usage

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

### Arguments

| | |
|---|---|
| x | vector of numeric factor levels |

### Value

Returns the vector of contrast coefficients

### Examples

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

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

`test.anova1.bs` *Between-subjects F statistic and eta-squared from summary information*

## Description

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

## Usage

```
test.anova1.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
- dfA - degrees of freedom for between-subjects factor
- p - p-value for F-test
- eta-squared - estimate of eta-squared
- adj eta-squared - a bias adjusted estimate of eta-squared

## Examples

```
m <- c(12.4, 8.6, 10.5)
sd <- c(3.84, 3.12, 3.48)
n <- c(20, 20, 20)
test.anova1.bs(m, sd, n)

#  Should return:
#             F dfA  dfE           p eta-squared adj eta-squared
# [1,] 5.919585   2   57 0.004614428   0.1719831       0.1429298
```

test.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 for test of zero excess kurtosis

## 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
# [1,]   4.8149  1.8149 0.0385
```

`test.mono.mean.bs` *Test of a monotonic trend in means for an ordered between-subjects factor*

### 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 monotoic 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 monotoic 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` *Test of monotonic trend in proportions for an ordered between-subjects factor*

---

## 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 monotoic 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 monotoic 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.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
# [1,]   17.41071  2 0.0001656958
```

`test.prop.ps` *Hypothesis test for a paired-samples proportion difference*

## Description

Computes a continuity-corrected McNemar test for equality of proportions in a paired-samples design. This function requires the frequency counts from a 2 x 2 contingency table for two paired dichotomous measurements.

## Usage

```
test.prop.ps(f00, f01, f10, f11)
```

## Arguments

| | |
|---|---|
| `f00` | number participants with y = 0 and x = 0 |
| `f01` | number participants with y = 0 and x = 1 |
| `f10` | number participants with y = 1 and x = 0 |
| `f11` | number participants with y = 1 and x = 1 |

## Value

Returns a 1-row matrix. The columns are:

- Estimate - ML estimate of proportion difference
- z - z test statistic
- p - p-value

## References

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

## Examples

```
test.prop.ps(156, 96, 68, 80)

# Should return:
#      Estimate        z          p
# [1,]     0.07 2.108346 0.03500109
```

`test.prop1` *Hypothesis test for a single proportion*

### Description

Computes a continuity-corrected z test for a single proportion in a 1-group design.

### Usage

```
test.prop1(f, n, h)
```

### Arguments

| | |
|---|---|
| `f` | number of participants who have the attribute |
| `n` | sample size |
| `h` | population proportion under null hypothesis |

### Value

Returns a 1-row matrix. The columns are:

- Estimate - ML estimate of proportion
- z - z test statistic
- p - p-value

### References

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

### Examples

```
test.prop1(9, 20, .2)

# Should return:
#      Estimate        z          p
# [1,]     0.45 2.515576 0.01188379
```

`test.prop2` *Hypothesis test for a 2-group proportion difference*

## Description

Computes a continuity-corrected z test for a difference of proportions in a 2-group design.

## Usage

```
test.prop2(f1, f2, n1, n2)
```

## Arguments

| | |
|---|---|
| `f1` | number of group 1 participants who have the attribute |
| `f2` | number of group 2 participants who have the attribute |
| `n1` | sample size 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 - 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 for test of zero skewness

### 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,]   1.5201 0.0067
```

# Index

ci.2x2.mean.bs, 5
ci.2x2.mean.mixed, 6
ci.2x2.mean.ws, 7
ci.2x2.prop.bs, 9
ci.2x2.prop.mixed, 10
ci.agree, 11
ci.agree2, 12
ci.biphi, 13
ci.cod1, 14
ci.cod2, 15
ci.condslope, 16
ci.condslope.log, 17
ci.cor, 18
ci.cor.dep, 19
ci.cor2, 20
ci.cor2.gen, 21
ci.cramer, 22
ci.cronbach, 23
ci.etasqr, 24
ci.fisher, 25
ci.indirect, 25
ci.kappa, 26
ci.lc.gen.bs, 27
ci.lc.glm, 28
ci.lc.mean.bs, 30
ci.lc.median.bs, 31
ci.lc.prop.bs, 32
ci.lc.reg, 33
ci.lc.stdmean.bs, 34
ci.lc.stdmean.ws, 35
ci.mad1, 36
ci.mann, 37
ci.mape, 38
ci.mean.ps, 39
ci.mean1, 40
ci.mean2, 41
ci.median.ps, 42
ci.median1, 43
ci.median2, 44
ci.oddsratio, 45
ci.pairs.prop.bs, 46
ci.pairs.prop1, 47
ci.pbcor, 48
ci.phi, 49
ci.popsize, 50
ci.prop.ps, 51
ci.prop1, 52
ci.prop2, 53
ci.random.anova1, 54
ci.ratio.mad.ps, 55
ci.ratio.mad2, 56
ci.ratio.mean.ps, 57
ci.ratio.mean2, 58
ci.ratio.median.ps, 59
ci.ratio.median2, 60
ci.ratio.prop.ps, 61
ci.ratio.prop2, 62
ci.reliability, 63
ci.rsqr, 64
ci.sign1, 65
ci.slope.mean.bs, 66
ci.slope.prop.bs, 67
ci.spcor, 68
ci.spear, 69
ci.spear2, 70
ci.stdmean.ps, 71
ci.stdmean.strat, 72
ci.stdmean1, 73
ci.stdmean2, 74
ci.tetra, 75
ci.theil, 76
ci.tukey, 77
ci.var.upper, 78
ci.yule, 79

etasqr.adj, 80
etasqr.gen.2way, 80

iqv, 81

pi.score.ps, 82
pi.score1, 83
pi.score2, 84

random.sample, 85
random.y, 86
random.yx, 86
randomize, 87

sim.ci.cor, 88
sim.ci.mean.ps, 89
sim.ci.mean1, 90
sim.ci.mean2, 91
sim.ci.median.ps, 92
sim.ci.median1, 93
sim.ci.median2, 94
sim.ci.spear, 95
sim.ci.stdmean.ps, 96
sim.ci.stdmean2, 97
size.ci.agree, 98
size.ci.condmean, 99
size.ci.cor, 100
size.ci.cronbach, 101
size.ci.lc.ancova, 102
size.ci.lc.mean.bs, 103
size.ci.lc.mean.ws, 103
size.ci.lc.prop.bs, 104
size.ci.lc.stdmean.bs, 105
size.ci.lc.stdmean.ws, 106
size.ci.mean.ps, 107
size.ci.mean1, 108
size.ci.mean2, 108
size.ci.prop.ps, 109
size.ci.prop1, 110
size.ci.prop2, 111
size.ci.ratio.mean.ps, 111
size.ci.ratio.mean2, 112
size.ci.ratio.prop.ps, 113
size.ci.ratio.prop2, 114
size.ci.rsqr, 115
size.ci.second, 115
size.ci.slope, 116
size.ci.stdmean.ps, 117
size.ci.stdmean2, 118
size.equiv.mean.ps, 119
size.equiv.mean2, 120
size.equiv.prop.ps, 121
size.equiv.prop2, 122
size.interval.cor, 123
size.supinf.mean.ps, 123
size.supinf.mean2, 124
size.supinf.prop.ps, 125
size.supinf.prop2, 126
size.test.cor, 127
size.test.cronbach, 128
size.test.lc.ancova, 129
size.test.lc.mean.bs, 130
size.test.lc.mean.ws, 130
size.test.lc.prop.bs, 131
size.test.mann, 132
size.test.mean.ps, 133
size.test.mean1, 134
size.test.mean2, 135
size.test.prop.ps, 135
size.test.prop1, 136
size.test.prop2, 137
size.test.sign.ps, 138
size.test.sign1, 138
size.test.slope, 139
slope.contrast, 140

test.anova1.bs, 141
test.kurtosis, 142
test.mono.mean.bs, 143
test.mono.prop.bs, 144
test.prop.bs, 145
test.prop.ps, 146
test.prop1, 147
test.prop2, 148
test.skew, 149