Package ‘vcmeta’

August 21, 2021

Type Package

Title Varying Coefficient Meta-Analysis

Version 1.0.0

Description Implements functions for varying coefficient meta-analysis methods. These methods do not assume effect size homogeneity. Subgroup effect size comparisons, general linear effect size contrasts, and linear models of effect sizes based on varying coefficient methods can be used to describe effect size heterogeneity. Varying coefficient meta-analysis methods do not require the unrealistic assumptions of the traditional fixed-effect and random-effects meta-analysis methods.

For details, see:
Bonett (2008) <doi:10.1037/a0012868>,
Bonett (2009) <doi:10.1037/a0016619>,
Bonett (2010) <doi:10.1037/a0020142>,
Bonett & Price (2014) <doi:10.1111/bmsp.12024>,
Bonett (2020) <doi:10.1111/bmsp.12189>,

URL https://github.com/dgbonett/vcmeta

BugReports https://github.com/dgbonett/vcmeta/issues

License GPL-3

Encoding UTF-8

Imports stats, mathjaxr, Rdpack, ggplot2

RoxygenNote 7.1.1

RdMacros Rdpack, mathjaxr

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

NeedsCompilation no

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

Maintainer Douglas G. Bonett <dgbonett@ucsc.edu>
Repository: CRAN
Date/Publication: 2021-08-21 09:40:02 UTC

R topics documented:

- ci.fisher .................................................. 4
- cor.from.t .................................................. 4
- meta.ave.agree ............................................. 5
- meta.ave.cor ............................................... 6
- meta.ave.cronbach ......................................... 7
- meta.ave.gen ............................................... 8
- meta.ave.gen.cc ........................................... 10
- meta.ave.gen.rc ........................................... 11
- meta.ave.mean.ps ......................................... 12
- meta.ave.mean2 ............................................ 14
- meta.ave.meanratio.ps .................................... 15
- meta.ave.meanratio2 ...................................... 16
- meta.ave.odds ............................................. 18
- meta.ave.path ............................................. 19
- meta.ave.pbcor ........................................... 20
- meta.ave.plot ............................................. 22
- meta.ave.prop.ps ......................................... 23
- meta.ave.prop2 ............................................ 24
- meta.ave.propratio2 ...................................... 26
- meta.ave.semipart ................................-------- 27
- meta.ave.slope ............................................ 28
- meta.ave.spear ............................................ 29
- meta.ave.stdmean.ps ..................................... 30
- meta.ave.stdmean2 ........................................ 32
- meta.lc.agree ............................................. 33
- meta.lc.gen ............................................... 34
- meta.lc.mean.ps .......................................... 35
- meta.lc.mean1 ............................................. 36
- meta.lc.mean2 ............................................. 37
- meta.lc.meanratio.ps ..................................... 39
- meta.lc.meanratio2 ....................................... 40
- meta.lc.odds .............................................. 41
- meta.lc.prop.ps .......................................... 43
- meta.lc.prop1 .............................................. 44
- meta.lc.prop2 .............................................. 45
- meta.lc.propratio2 ....................................... 46
- meta.lc.stdmean.ps ...................................... 47
- meta.lc.stdmean2 ........................................ 49
- meta.lm.agree ............................................. 50
- meta.lm.cor ............................................... 51
- meta.lm.cor.gen .......................................... 52
- meta.lm.cronbach .......................................... 53
topics documented:

meta.lm.gen .................................................. 55
meta.lm.mean.ps ............................................ 56
meta.lm.mean1 ................................................ 57
meta.lm.mean2 ................................................ 58
meta.lm.meanratio.ps ...................................... 59
meta.lm.meanratio2 ......................................... 61
meta.lm.odds .................................................. 62
meta.lm.prop.ps .............................................. 64
meta.lm.prop1 ................................................ 65
meta.lm.prop2 ................................................ 66
meta.lm.propratio2 ......................................... 67
meta.lm.semiart .............................................. 69
meta.lm.spear ............................................... 70
meta.lm.stdmean.ps ...................................... 71
meta.lm.stdmean2 .......................................... 72
meta.sub.cor ............................................... 74
meta.sub.cronbach ......................................... 75
meta.sub.pbcor ............................................. 76
meta.sub.semiart .......................................... 78
meta.sub.spear ............................................. 79
replicate.cor ............................................... 81
replicate.gen ............................................... 82
replicate.mean.ps ........................................ 83
replicate.mean2 ........................................... 85
replicate.plot .............................................. 87
replicate.stdmean.ps ................................... 89
replicate.stdmean2 ........................................ 91
se.cor ......................................................... 93
se.mean.ps .................................................. 94
se.mean2 ..................................................... 95
se.meanratio.ps ........................................... 96
se.meanratio2 ............................................... 97
se.odds ......................................................... 98
se.pbcor ..................................................... 99
se.semiart .................................................. 100
se.slope ....................................................... 101
se.spear ..................................................... 102
se.stdmean.ps ............................................. 103
se.stdmean2 ............................................... 104

Index ......................................................... 106
ci.fisher  
*Fisher confidence interval for any type of correlation.*

**Description**

This function computes a confidence interval for any type correlation using an estimated correlation and its standard error. This function should be used with the meta.ave.gen function when the effect size is a correlation. Use the estimated average correlation and its standard error from meta.ave.gen (when the effect size is a correlation) in the ci.fisher function to obtain a more accurate confidence interval for the population average correlation.

**Usage**

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

**Arguments**

- `alpha`: alpha value for 1-alpha confidence
- `cor`: estimate of correlation
- `se`: standard error of estimated correlation

**Value**

A 2-element vector with lower and upper bounds of the confidence interval

**Examples**

```r
ci.fisher(0.05, 0.50, .10)
```

# Should return:
# [1] 0.2802723 0.6699402

---

cor.from.t  
*Compute Pearson correlation between paired measurements from t statistic*

**Description**

This function computes the Pearson correlation between paired measurements using a reported paired-samples t statistic and other sample information. This correlation estimate is needed in several functions that analyze mean differences and standardized mean differences in paired-samples studies.
### meta.ave.agree

**Usage**

```r
cor.from.t(m1, m2, sd1, sd2, t, n)
```

**Arguments**

- `m1`: sample mean for measurement 1
- `m2`: sample mean for measurement 2
- `sd1`: sample standard deviation for measurement 1
- `sd2`: sample standard deviation for measurement 2
- `t`: value for paired-samples t-test
- `n`: sample size

**Value**

Returns the sample Pearson correlation between the two paired measurements

**Examples**

```r
cor.from.t(9.4, 9.8, 1.26, 1.40, 2.27, 30)
```

# Should return:
# [1] 0.7415209

---

### meta.ave.agree

**Description**

Computes the estimate, standard error, and confidence interval for an G-index of agreement from two or more studies. This function assumes that two raters each provide a dichotomous rating to a sample of objects. The G-index of agreement is usually preferred to Cohen’s kappa.

**Usage**

```r
meta.ave.agree(alpha, f11, f12, f21, f22, bystudy = TRUE)
```

**Arguments**

- `alpha`: alpha level for 1-alpha confidence
- `f11`: vector of frequencies in cell 1,1
- `f12`: vector of frequencies in cell 1,2
- `f21`: vector of frequencies in cell 2,1
- `f22`: vector of frequencies in cell 2,2
- `bystudy`: logical to also return each study estimate (TRUE) or not
Value

Returns a matrix. The first row is the average estimate across all studies. If bystudy is true, there is 1 additional row for each study. The matrix has the following columns:

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

Examples

```r
f11 <- c(17, 28, 19)
f12 <- c(43, 56, 49)
f21 <- c(3, 5, 5)
f22 <- c(37, 54, 39)
meta.ave.agree(.05, f11, f12, f21, f22, bystudy = TRUE)
```

# Should return:

```r
# Estimate  SE   LL    UL
# Average  0.08657934 0.05312584 -0.01754538 0.1907041
# Study 1  0.07692308 0.09776752 -0.11469775 0.2685439
# Study 2  0.14285714 0.08163265 -0.01713992 0.3028542
# Study 3  0.03448276 0.09279245 -0.14738711 0.2163526
```

Description

Computes the estimate, standard error, and confidence interval for an average Pearson or partial correlation. The sample correlations must be all Pearson correlations or all partial correlations. Use the meta.ave.gen function to meta-analysis any combination of Pearson, partial, or Spearman correlations.

Usage

```
meta.ave.cor(alpha, n, cor, q, bystudy = TRUE)
```

Arguments

- `alpha`: alpha level for 1-alpha confidence
- `n`: vector of sample sizes
- `cor`: vector of sample correlations
- `q`: number of control variables
- `bystudy`: logical to also return each study estimate (TRUE) or not
meta.ave.cronbach

Value

Returns a matrix. The first row is the average estimate across all studies. If bystudy is true, there is 1 additional row for each study. The matrix has the following columns:

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

References


Examples

```r
n <- c(55, 190, 65, 35)
cor <- c(.40, .65, .60, .45)
meta.ave.cor(.05, n, cor, 0, bystudy = TRUE)
```

# Should return:

<table>
<thead>
<tr>
<th></th>
<th>Estimate</th>
<th>SE</th>
<th>LL</th>
<th>UL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average</td>
<td>0.525</td>
<td>0.05113361</td>
<td>0.4176678</td>
<td>0.6178816</td>
</tr>
<tr>
<td>Study 1</td>
<td>0.400</td>
<td>0.11430952</td>
<td>0.1506943</td>
<td>0.6014699</td>
</tr>
<tr>
<td>Study 2</td>
<td>0.650</td>
<td>0.04200694</td>
<td>0.5594086</td>
<td>0.7252465</td>
</tr>
<tr>
<td>Study 3</td>
<td>0.600</td>
<td>0.08000000</td>
<td>0.4171458</td>
<td>0.7361686</td>
</tr>
<tr>
<td>Study 4</td>
<td>0.450</td>
<td>0.13677012</td>
<td>0.1373507</td>
<td>0.6811071</td>
</tr>
</tbody>
</table>
```

meta.ave.cronbach

Confidence interval for an average Cronbach alpha reliability

Description

Computes the estimate, standard error, and confidence interval for an average Cronbach reliability coefficient from two or more studies.

Usage

```r
meta.ave.cronbach(alpha, n, rel, q, bystudy = TRUE)
```
Arguments

alpha alpha level for 1-alpha confidence
n vector of sample sizes
rel vector of sample reliabilities
q number of measurements (e.g., items) used to compute each reliability
bystudy logical to also return each study estimate (TRUE) or not

Value

Returns a matrix. The first row is the average estimate across all studies. If bystudy is true, there is 1 additional row for each study. The matrix has the following columns:

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

References


Examples

```r
n <- c(583, 470, 546, 680)
rel <- c(.91, .89, .90, .89)
meta.ave.cronbach(.05, n, rel, 10, bystudy = TRUE)
```

# Should return:

```
#   Estimate   SE      LL      UL
# Average 0.8975 0.003256081 0.8911102 0.9038592
# Study 1 0.9100 0.005566064 0.8985763 0.9204108
# Study 2 0.8900 0.007579900 0.8743616 0.9041013
# Study 3 0.9000 0.006391375 0.8868623 0.9119356
# Study 4 0.8900 0.006297549 0.8771189 0.9018203
```
**Usage**

`meta.ave.gen(alpha, est, se, bystudy = TRUE)`

**Arguments**

- `alpha` alpha level for 1-alpha confidence
- `est` vector of parameter estimates
- `se` vector of standard errors
- `bystudy` logical to also return each study estimate (TRUE) or not

**Value**

Returns a matrix. The first row is the average estimate across all studies. If `bystudy` is true, there is 1 additional row for each study. The matrix has the following columns:

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

**Examples**

```r
est <- c(.022, .751, .421, .287, .052, .146, .562, .904)
se <- c(.124, .464, .102, .592, .864, .241, .252, .318)
meta.ave.gen(.05, est, se, bystudy = TRUE)
```

# Should return:

<table>
<thead>
<tr>
<th></th>
<th>Estimate</th>
<th>SE</th>
<th>LL</th>
<th>UL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average</td>
<td>0.393125</td>
<td>0.1561622</td>
<td>0.08705266</td>
<td>0.6991973</td>
</tr>
<tr>
<td>Study 1</td>
<td>0.022000</td>
<td>0.1240000</td>
<td>-0.22103553</td>
<td>0.2650355</td>
</tr>
<tr>
<td>Study 2</td>
<td>0.751000</td>
<td>0.4640000</td>
<td>-0.15842329</td>
<td>1.6604233</td>
</tr>
<tr>
<td>Study 3</td>
<td>0.421000</td>
<td>0.1020000</td>
<td>0.22108367</td>
<td>0.6209163</td>
</tr>
<tr>
<td>Study 4</td>
<td>0.287000</td>
<td>0.5920000</td>
<td>-0.87329868</td>
<td>1.4472987</td>
</tr>
<tr>
<td>Study 5</td>
<td>0.052000</td>
<td>0.8640000</td>
<td>-1.64140888</td>
<td>1.7454089</td>
</tr>
<tr>
<td>Study 6</td>
<td>0.146000</td>
<td>0.2410000</td>
<td>-0.32635132</td>
<td>0.6183513</td>
</tr>
<tr>
<td>Study 7</td>
<td>0.562000</td>
<td>0.2520000</td>
<td>0.06889988</td>
<td>1.8559109</td>
</tr>
<tr>
<td>Study 8</td>
<td>0.904000</td>
<td>0.3180000</td>
<td>0.28073145</td>
<td>1.5272685</td>
</tr>
</tbody>
</table>
```
meta.ave.gen.cc

Confidence interval for an average effect size using a constant coefficient model

Description
Computes the estimate, standard error, and confidence interval for a weighted average effect size using the constant coefficient (fixed-effect) meta-analysis model. The weighted average estimate will be biased regardless of number of studies or sample size per study and the actual confidence interval coverage probability can be much smaller than the specified confidence level when the true effect sizes are not identical across studies.

Usage
meta.ave.gen.cc(alpha, est, se, bystudy = TRUE)

Arguments
alpha          alpha level for 1-alpha confidence
est            vector of parameter estimates
se             vector of standard errors
bystudy       logical to also return each study estimate (TRUE) or not

Value
Returns a matrix. The first row is the average estimate across all studies. If bystudy is true, there is 1 additional row for each study. The matrix has the following columns:

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

References
Examples

```r
est <- c(.022, .751, .421, .287, .052, .146, .562, .904)
se <- c(.124, .464, .102, .592, .864, .241, .252, .318)
meta.ave.gen.cc(.05, est, se, bystudy = TRUE)
```

# Should return:

<table>
<thead>
<tr>
<th>#</th>
<th>Estimate</th>
<th>SE</th>
<th>LL</th>
<th>UL</th>
</tr>
</thead>
<tbody>
<tr>
<td># Average</td>
<td>0.3127916</td>
<td>0.06854394</td>
<td>0.17844794</td>
<td>0.4471352</td>
</tr>
<tr>
<td># Study 1</td>
<td>0.0220000</td>
<td>0.12400000</td>
<td>-0.22103553</td>
<td>0.2650355</td>
</tr>
<tr>
<td># Study 2</td>
<td>0.7510000</td>
<td>0.46400000</td>
<td>-0.15842329</td>
<td>1.6604233</td>
</tr>
<tr>
<td># Study 3</td>
<td>0.4210000</td>
<td>0.10200000</td>
<td>0.22108367</td>
<td>0.6209163</td>
</tr>
<tr>
<td># Study 4</td>
<td>0.2870000</td>
<td>0.59200000</td>
<td>-0.87329868</td>
<td>1.4472987</td>
</tr>
<tr>
<td># Study 5</td>
<td>0.0520000</td>
<td>0.86400000</td>
<td>-1.64140888</td>
<td>1.7454089</td>
</tr>
<tr>
<td># Study 6</td>
<td>0.1460000</td>
<td>0.24100000</td>
<td>-0.32635132</td>
<td>0.6183513</td>
</tr>
<tr>
<td># Study 7</td>
<td>0.5620000</td>
<td>0.25200000</td>
<td>0.06808908</td>
<td>1.0559109</td>
</tr>
<tr>
<td># Study 8</td>
<td>0.9040000</td>
<td>0.31800000</td>
<td>0.28073145</td>
<td>1.5272685</td>
</tr>
</tbody>
</table>

Description

Computes the estimate, standard error, and confidence interval for a weighted average effect using the random coefficient (random-effects) meta-analysis model. An estimate of effect size heterogeneity (tau-squared) is also computed. The random coefficient model assumes that the studies in the meta-analysis are a random sample from some definable superpopulation of studies. This assumption is difficult to justify. The weighted average estimate will be biased regardless of number of studies or sample size per study and the actual confidence interval coverage probability can much smaller than the specified confidence level if the effect sizes are correlated with the weights. This method also assume that the true effects sizes in the superpopulation of studies have a normal distribution. A large number of studies, each with a large sample size, is required to assess the superpopulation normality assumption. The traditional confidence interval for the population tau-squared is hypersensitive to very minor and difficult to detect violations of the superpopulation normality assumption.

Usage

```r
meta.ave.gen.rc(alpha, est, se, bystudy = TRUE)
```

Arguments

- **alpha**: alpha level for 1-alpha confidence
- **est**: vector of parameter estimates
- **se**: vector of standard errors
- **bystudy**: logical to also return each study estimate (TRUE) or not
Value

Returns a matrix. The first row is the average estimate across all studies. If bystudy is true, there is 1 additional row for each study. The matrix has the following columns:

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

References


Examples

```r
est <- c(.022, .751, .421, .287, .052, .146, .562, .904)
se <- c(.124, .464, .102, .592, .864, .241, .252, .318)
meta.ave.gen.rc(.05, est, se, bystudy = TRUE)
# Should return:
# Estimate SE LL UL
# Tau-squared 0.03772628 0.0518109 0.0000000 0.1392738
# Average 0.35394806 0.1155239 0.12752528 0.5803708
# Study 1 0.02200000 0.1240000 -0.22103553 0.2650355
# Study 2 0.75100000 0.4640000 -0.15842329 1.6604233
# Study 3 0.42100000 0.1020000 0.22108367 0.6209163
# Study 4 0.28700000 0.5920000 -0.87329868 1.4472987
# Study 5 0.05200000 0.8640000 -1.64140888 1.7454089
# Study 6 0.14600000 0.2410000 -0.32635132 0.6183513
# Study 7 0.56200000 0.2520000 0.06808908 1.0559109
# Study 8 0.90400000 0.3180000 0.28073145 1.5272685
```

---

**meta.ave.mean.ps**

Confidence interval for an average mean difference from paired-samples studies

Description

Computes the estimate, standard error, and confidence interval for an average mean difference from two or more paired-samples studies. A Satterthwaite adjustment to the degrees of freedom is used to improve the accuracy of the confidence interval for the average effect size. Equal variances within or across studies is not assumed.
meta.ave.mean.ps

Usage

meta.ave.mean.ps(alpha, m1, m2, sd1, sd2, cor, n, bystudy = TRUE)

Arguments

alpha    alpha level for 1-alpha confidence
m1       vector of sample means for measurement 1
m2       vector of sample means for measurement 2
sd1      vector of sample SDs for measurement 1
sd2      vector of sample SDs for measurement 2
cor      vector of sample correlations for paired measurements
n        vector of sample sizes
bystudy  logical to also return each study estimate (TRUE) or not

Value

A matrix. First row is the overall average estimate. If bystudy is TRUE also returns 1 row per study. The matrix has the following columns:

- Estimate - the estimated effect size
- SE - standard error
- LL - lower limit of the confidence interval
- UL - upper limit of the confidence interval
- df - degrees of freedom

References


Examples

m1 <- c(53, 60, 53, 57)
m2 <- c(55, 62, 58, 61)
sd1 <- c(4.1, 4.2, 4.5, 4.0)
sd2 <- c(4.2, 4.7, 4.9, 4.8)
cor <- c(.7, .7, .8, .85)
n <- c(30, 50, 30, 70)
meta.ave.mean.ps(.05, m1, m2, sd1, sd2, cor, n, bystudy = TRUE)

# Should return:
# Estimate  SE     LL     UL      df
# Average  -3.25  0.2471557 -3.739691 -2.7603091 112.347
# Study 1   -2.00  0.5871400 -3.200836 -0.7991639  29.000
# Study 2   -2.00  0.4918335 -2.988335 -1.0116648  49.000
# Study 3   -5.00  0.5471136 -6.118973 -3.8810270  29.000
Description
Computes the estimate, standard error, and confidence interval for an average mean difference from two or more 2-group studies. A Satterthwaite adjustment to the degrees of freedom is used to improve the accuracy of the confidence intervals. Equal variances within or across studies is not assumed.

Usage
meta.ave.mean2(alpha, m1, m2, sd1, sd2, n1, n2, bystudy = TRUE)

Arguments
- alpha: alpha level for 1-alpha confidence
- m1: vector of sample means for group 1
- m2: vector of sample means for group 2
- sd1: vector of sample SDs for group 1
- sd2: vector of sample SDs for group 2
- n1: vector of group 1 sample size
- n2: vector of group 2 sample sizes
- bystudy: logical to also return each study estimate (TRUE) or not

Value
Returns a matrix. The first row is the average estimate across all studies. If bystudy is true, there is 1 additional row for each study. The matrix has the following columns:
- Estimate - the estimated effect size
- SE - standard error
- LL - lower limit of the confidence interval
- UL - upper limit of the confidence interval
- df - degrees of freedom

References
Examples

```r
m1 <- c(7.4, 6.9)
m2 <- c(6.3, 5.7)
sd1 <- c(1.72, 1.53)
sd2 <- c(2.35, 2.04)
n1 <- c(40, 60)
n2 <- c(40, 60)
meta.ave.mean2(.05, m1, m2, sd1, sd2, n1, n2, bystudy = TRUE)
```

# Should return:

<table>
<thead>
<tr>
<th></th>
<th>Estimate</th>
<th>SE</th>
<th>LL</th>
<th>UL</th>
<th>df</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average</td>
<td>1.15</td>
<td>0.2831</td>
<td>0.5904</td>
<td>1.71</td>
<td>139.41</td>
</tr>
<tr>
<td>Study 1</td>
<td>1.10</td>
<td>0.4605</td>
<td>0.1819</td>
<td>2.02</td>
<td>109.42</td>
</tr>
<tr>
<td>Study 2</td>
<td>1.20</td>
<td>0.3292</td>
<td>0.5476</td>
<td>1.85</td>
<td>109.42</td>
</tr>
</tbody>
</table>

---

**meta.ave.meanratio.ps**  
Confidence interval for an average mean ratio from paired-samples studies

Description

Computes the estimate, standard error, and confidence interval for an geometric average mean ratio from two or more paired-samples studies. A Satterthwaite adjustment to the degrees of freedom is used to improve the accuracy of the confidence interval for the average effect size. Equal variances within or across studies is not assumed.

Usage

```r
meta.ave.meanratio.ps(alpha, m1, m2, sd1, sd2, cor, n, bystudy = TRUE)
```

Arguments

- **alpha**: alpha level for 1-alpha confidence
- **m1**: vector of sample means for measurement 1
- **m2**: vector of sample means for measurement 2
- **sd1**: vector of sample SDs for measurement 1
- **sd2**: vector of sample SDs for measurement 2
- **cor**: vector of sample correlations for paired measurements
- **n**: vector of sample sizes
- **bystudy**: logical to also return each study estimate (TRUE) or not
Value

Returns a matrix. The first row is the average estimate across all studies. If bystudy is true, there is 1 additional row for each study. The matrix has the following columns:

- Estimate - the estimated effect size
- SE - standard error
- LL - lower limit of the confidence interval
- UL - upper limit of the confidence interval
- exp(Estimate) - the exponentiated estimate (the geometric mean)
- exp(LL) - lower limit of the exponentiated confidence interval
- exp(UL) - upper limit of the exponentiated confidence interval
- df - degrees of freedom

Examples

```r
m1 <- c(53, 60, 53, 57)
m2 <- c(55, 62, 58, 61)
sd1 <- c(4.1, 4.2, 4.5, 4.0)
sd2 <- c(4.2, 4.7, 4.9, 4.8)
cor <- c(.7, .7, .8, .85)
n <- c(30, 50, 30, 70)
meta.ave.meanratio.ps(.05, m1, m2, sd1, sd2, cor, n, bystudy = TRUE)
```

# Should return:
# | Estimate | SE     | LL     | UL     |
# |----------|--------|--------|--------|
# Average | -0.05695120 | 0.004350863 | -0.06558008 | -0.04832231 |
# Study 1  | -0.03704127 | 0.010871086 | -0.05927514 | -0.01480740 |
# Study 2  | -0.03278982 | 0.008021952 | -0.04891054 | -0.01666911 |
# Study 3  | -0.09015110 | 0.009779919 | -0.11015328 | -0.07014982 |
# Study 4  | -0.06782260 | 0.004970015 | -0.07773750 | -0.05790769 |
# | exp(Estimate) | exp(LL) | exp(UL) | df |
# | 0.9446402 | 0.9365240 | 0.9528266 | 103.0256 |
# Study 1  | 0.9636364 | 0.9424474 | 0.9853017 | 29.0000 |
# Study 2  | 0.9677419 | 0.9522663 | 0.9834691 | 49.0000 |
# Study 3  | 0.9137931 | 0.8956968 | 0.9322550 | 29.0000 |
# Study 4  | 0.9344262 | 0.9252073 | 0.9437371 | 69.0000 |
```

Description

Computes the estimate, standard error, and confidence interval for an geometric average mean ratio from two or more 2-group studies. A Satterthwaite adjustment to the degrees of freedom is used to improve the accuracy of the confidence intervals. Equal variances within or across studies is not assumed.
Usage

meta.ave.meanratio2(alpha, m1, m2, sd1, sd2, n1, n2, bystudy = TRUE)

Arguments

alpha       alpha level for 1-alpha confidence
m1          vector of sample means for group 1
m2          vector of sample means for group 2
sd1         vector of sample SDs for group 1
sd2         vector of sample SDs for group 2
n1          vector of group 1 sample size
n2          vector of group 2 sample sizes
bystudy     logical to also return each study estimate (TRUE) or not

Value

A matrix. First row is the overall average estimate. If bystudy is TRUE also returns 1 row per study.

The matrix has the following columns:

• Estimate - the estimated effect size
• SE - standard error
• LL - lower limit of the confidence interval
• UL - upper limit of the confidence interval
• exp(Estimate) - the exponentiated estimate (the geometric mean)
• exp(LL) - lower limit of the exponentiated confidence interval
• exp(UL) - upper limit of the exponentiated confidence interval
• df - degrees of freedom

References


Examples

m1 <- c(7.4, 6.9)
m2 <- c(6.3, 5.7)
sd1 <- c(1.7, 1.5)
sd2 <- c(2.3, 2.0)
n1 <- c(40, 20)
n2 <- c(40, 20)
meta.ave.meanratio2(.05, m1, m2, sd1, sd2, n1, n2, bystudy = TRUE)

# Should return:
#   Estimate    SE    LL    UL  exp(Estimate)
<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Average</td>
<td>0.1759928</td>
<td>0.05738065</td>
<td>0.061437186</td>
<td>0.2905484</td>
<td>1.192429</td>
</tr>
<tr>
<td>Study 1</td>
<td>0.1609304</td>
<td>0.06820167</td>
<td>0.024749712</td>
<td>0.2971110</td>
<td>1.174603</td>
</tr>
<tr>
<td>Study 2</td>
<td>0.1910552</td>
<td>0.09229675</td>
<td>0.002986265</td>
<td>0.3791242</td>
<td>1.210526</td>
</tr>
<tr>
<td></td>
<td>exp(LL)</td>
<td>exp(UL)</td>
<td>df</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Average</td>
<td>1.063364</td>
<td>1.337161</td>
<td>66.26499</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Study 1</td>
<td>1.025059</td>
<td>1.345965</td>
<td>65.69929</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Study 2</td>
<td>1.002991</td>
<td>1.461004</td>
<td>31.71341</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**meta.ave.odds**

*Confidence interval for average odds ratio from 2-group studies*

**Description**

Computes the estimate, standard error, and confidence interval for a geometric mean odds ratio from two or more studies.

**Usage**

`meta.ave.odds(alpha, f1, f2, n1, n2, bystudy = TRUE)`

**Arguments**

- **alpha**: alpha level for 1-alpha confidence
- **f1**: vector of group 1 event counts
- **f2**: vector of group 2 event counts
- **n1**: vector of group 1 sample sizes
- **n2**: vector of group 2 sample sizes
- **bystudy**: logical to also return each study estimate (TRUE) or not

**Value**

Returns a matrix. The first row is the average estimate across all studies. If bystudy is true, there is 1 additional row for each study. The matrix has the following columns:

- Estimate - the estimated effect size
- SE - standard error
- LL - lower limit of the confidence interval
- UL - upper limit of the confidence interval
- exp(Estimate) - the exponentiated estimate
- exp(LL) - lower limit of the exponentiated confidence interval
- exp(UL) - upper limit of the exponentiated confidence interval
References


Examples

n1 <- c(204, 201, 932, 130, 77)
n2 <- c(106, 103, 415, 132, 83)
f1 <- c(24, 40, 93, 14, 5)
f2 <- c(12, 9, 28, 3, 1)
meta.ave.odds(.05, f1, f2, n1, n2, bystudy = TRUE)

# Should return:
# Estimate SE LL UL
# Average 0.86211102 0.2512852 0.36960107 1.3546210
# Study 1 0.02581353 0.3700520 -0.69947512 0.7511022
# Study 2 0.91410487 0.3830515 0.16333766 1.6648721
# Study 3 0.41496672 0.2226089 -0.02133877 0.8512722
# Study 4 1.52717529 0.6090858 0.83338907 2.7209615
# Study 5 1.42849472 0.9350931 -0.40425414 3.2612436
# exp(Estimate) exp(LL) exp(UL)
# Average 2.368155 1.4471572 3.875292
# Study 1 1.026150 0.4968460 2.119335
# Study 2 2.494541 1.1774342 5.284997
# Study 3 1.514320 0.9788873 2.342625
# Study 4 4.605150 1.3956902 15.194925
# Study 5 4.172414 0.6674745 26.081952

---

**meta.ave.path**

Confidence interval for an average slope coefficient in a general linear model or a path model.

**Description**

Computes the estimate, standard error, and confidence interval for an average slope coefficient from a general linear model (ANOVA, ANCOVA, multiple regression) or a path model.

**Usage**

meta.ave.path(alpha, n, slope, se, s, bystudy = TRUE)

**Arguments**

- **alpha**
  - alpha level for 1-alpha confidence

- **n**
  - vector of sample sizes
### meta.ave.pbcorm

**slope**  
vector of slope estimates

**se**  
vector of slope standard errors

**s**  
number of predictors of the response variable

**bystudy**  
logical to also return each study estimate (TRUE) or not

### Value

Returns a matrix. The first row is the average estimate across all studies. If bystudy is true, there is 1 additional row for each study. The matrix has the following columns:

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

### Examples

```r
n <- c(75, 85, 250, 160)
slope <- c(1.57, 1.38, 1.08, 1.25)
se <- c(.658, .724, .307, .493)
meta.ave.path(.05, n, slope, se, 2, bystudy = TRUE)
```

# Should return:

<table>
<thead>
<tr>
<th></th>
<th>Estimate</th>
<th>SE</th>
<th>LL</th>
<th>UL</th>
<th>df</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average</td>
<td>1.32</td>
<td>0.2844</td>
<td>0.7599</td>
<td>1.88</td>
<td>263.18</td>
</tr>
<tr>
<td>Study 1</td>
<td>1.57</td>
<td>0.658</td>
<td>0.2583</td>
<td>2.88</td>
<td>72.00</td>
</tr>
<tr>
<td>Study 2</td>
<td>1.38</td>
<td>0.724</td>
<td>-0.06</td>
<td>2.82</td>
<td>82.00</td>
</tr>
<tr>
<td>Study 3</td>
<td>1.08</td>
<td>0.307</td>
<td>0.4753</td>
<td>1.68</td>
<td>247.00</td>
</tr>
<tr>
<td>Study 4</td>
<td>1.25</td>
<td>0.493</td>
<td>0.2762</td>
<td>2.22</td>
<td>157.00</td>
</tr>
</tbody>
</table>

---

### meta.ave.pbcorm

**Confidence interval for an average point-biserial correlation**

### Description

Computes the estimate, standard error, and confidence interval for an average point-biserial correlation from two or more studies. Two types of point-biserial correlations can be meta-analyzed. One type uses an unweighted variance and is appropriate in 2-group experimental designs. The other type uses a weighted variance and is appropriate in 2-group nonexperimental designs with simple random sampling within each group. This function requires all point-biserial correlations to be of the same type. Use the meta.ave.gen function to meta-analyze any combination of biserial correlation types.
Usage

meta.ave.pbcor(alpha, m1, m2, sd1, sd2, n1, n2, type, bystudy = TRUE)

Arguments

alpha  alpha level for 1-alpha confidence
m1     vector of sample means for group 1
m2     vector of sample means for group 2
sd1    vector of sample SDs for group 1
sd2    vector of sample SDs for group 2
n1     vector of group 1 sample sizes
n2     vector of group 2 sample sizes
type   • set to 1 for weighted variance
     • set to 2 for unweighted variance
bystudy logical to also return each study estimate (TRUE) or not

Value

Returns a matrix. The first row is the average estimate across all studies. If bystudy is true, there is
1 additional row for each study. The matrix has the following columns:

• Estimate - the estimated effect size
• 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),

Examples

m1 <- c(21.9, 23.1, 19.8)
m2 <- c(16.1, 17.4, 15.0)
sd1 <- c(3.82, 3.95, 3.67)
sd2 <- c(3.21, 3.30, 3.02)
n1 <- c(40, 30, 24)
n2 <- c(40, 28, 25)
meta.ave.pbcor(.05, m1, m2, sd1, sd2, n1, n2, 2, bystudy = TRUE)

# Should return:
# Average 0.6159094 0.04363432 0.5230976 0.6942842
# Study 1 0.6349786 0.06316796 0.4842020 0.7370220
meta.ave.plot

Forest plot for average effect sizes

Description

Generates a forest plot to visualize effect sizes estimates and overall averages from the meta.ave functions in vcmeta. If the column exp(Estimate) is present, this function plots the exponentiated effect size and CI found in columns exp(Estimate), exp(LL), and exp(UL). Otherwise, this function plots the effect size and CI found in the columns Estimate, LL, and UL.

Usage

meta.ave.plot(
  result,
  reference_line = NULL,
  diamond_height = 0.2,
  ggtheme = ggplot2::theme_classic()
)

Arguments

result • a result matrix from any of the replicate functions in vcmeta
reference_line Optional x-value for a reference line. Only applies if focus is 'Difference' or 'Both'. Defaults to NULL, in which case a reference line is not drawn.
diamond_height • Optional height of the diamond representing average effect size. Only applies if focus is 'Average' or 'Both'. Defaults to 0.2
ggtheme • optional ggplot2 theme object; defaults to theme_classic()

Value

Returns a ggplot object. If stored, can be further customized via the ggplot API

Examples

# Plot results from meta.ave.mean2
m1 <- c(7.4, 6.9)
m2 <- c(6.3, 5.7)
sd1 <- c(1.72, 1.53)
sd2 <- c(2.35, 2.04)
n1 <- c(40, 60)
n2 <- c(40, 60)
result <- meta.ave.mean2(.05, m1, m2, sd1, sd2, n1, n2, bystudy = TRUE)
meta.ave.plot(result, reference_line = 0)
# Plot results from meta.ave.meanratio2
# Note that this plots the exponentiated effect size and CI
m1 <- c(53, 60, 53, 57)
m2 <- c(55, 62, 58, 61)
sd1 <- c(4.1, 4.2, 4.5, 4.0)
sd2 <- c(4.2, 4.7, 4.9, 4.8)
cor <- c(.7, .7, .8, .85)
n <- c(30, 50, 30, 70)
result <- meta.ave.meanratio.ps(.05, m1, m2, sd1, sd2, cor, n, bystudy = TRUE)
myplot <- meta.ave.plot(result, reference_line = 1)
myplot

# Change x-scale to log2
library(ggplot2)
myplot <- myplot + scale_x_continuous(
  trans = 'log2',
  limits = c(0.75, 1.25),
  name = "Estimated Ratio of Means, Log2 Scale"
)
myplot

---

meta.ave.prop.ps  

Confidence interval for an average proportion difference in paired-samples studies

**Description**

Computes the estimate, standard error, and confidence interval for an average proportion difference from two or more studies.

**Usage**

meta.ave.prop.ps(alpha, f11, f12, f21, f22, bystudy = TRUE)

**Arguments**

- `alpha` alpha level for 1-alpha confidence
- `f11` vector of frequencies in cell 1,1
- `f12` vector of frequencies in cell 1,2
- `f21` vector of frequencies in cell 2,1
- `f22` vector of frequencies in cell 2,2
- `bystudy` logical to also return each study estimate (TRUE) or not
meta.ave.prop2

Value

Returns a matrix. The first row is the average estimate across all studies. If bystudy is true, there is 1 additional row for each study. The matrix has the following columns:

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

References


Examples

```r
f11 <- c(17, 28, 19)
f12 <- c(43, 56, 49)
f21 <- c(3, 5, 5)
f22 <- c(37, 54, 39)
meta.ave.prop.ps(.05, f11, f12, f21, f22, bystudy = TRUE)
```

`meta.ave.prop2(alpha, f1, f2, n1, n2, bystudy = TRUE)`
Arguments

alpha alpha level for 1-alpha confidence
f1 vector of group 1 event counts
f2 vector of group 2 event counts
n1 vector of group 1 sample sizes
n2 vector of group 2 sample sizes
bystudy logical to also return each study estimate (TRUE) or not

Value

Returns a matrix. The first row is the average estimate across all studies. If bystudy is true, there is 1 additional row for each study. The matrix has the following columns:

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

References


Examples

n1 <- c(204, 201, 932, 130, 77)
n2 <- c(106, 103, 415, 132, 83)
f1 <- c(24, 40, 93, 14, 5)
f2 <- c(12, 9, 28, 3, 1)
meta.ave.prop2(.05, f1, f2, n1, n2, bystudy = TRUE)

# Should return:
# Estimate SE LL UL
# Average 0.0567907589 0.01441216 2.854345e-02 0.08503807
# Study 1 0.0009888529 0.03870413 -7.486985e-02 0.07684756
# Study 2 0.1067323481 0.04018243 2.797623e-02 0.18548847
# Study 3 0.0310980338 0.01587717 -2.064379e-05 0.06221671
# Study 4 0.0837856174 0.03129171 2.245499e-02 0.14511624
# Study 5 0.0524199553 0.03403926 -1.429577e-02 0.11913568
Description

Computes the estimate, standard error, and confidence interval for a geometric mean proportion ratio from two or more studies.

Usage

`meta.ave.propratio2(alpha, f1, f2, n1, n2, bystudy = TRUE)`

Arguments

- `alpha`: alpha level for 1-alpha confidence
- `f1`: vector of group 1 event counts
- `f2`: vector of group 2 event counts
- `n1`: vector of group 1 sample sizes
- `n2`: vector of group 2 sample sizes
- `bystudy`: logical to also return each study estimate (TRUE) or not

Value

Returns a matrix. The first row is the average estimate across all studies. If `bystudy` is true, there is 1 additional row for each study. The matrix has the following columns:

- **Estimate** - the estimated effect size
- **SE** - standard error
- **LL** - lower limit of the confidence interval
- **UL** - upper limit of the confidence interval
- **exp(Estimate)** - the exponentiated estimate
- **exp(LL)** - lower limit of the exponentiated confidence interval
- **exp(UL)** - upper limit of the exponentiated confidence interval

References

Examples

n1 <- c(204, 201, 932, 130, 77)
n2 <- c(106, 103, 415, 132, 83)
f1 <- c(24, 40, 93, 14, 5)
f2 <- c(12, 9, 28, 3, 1)
meta.ave.propratio2(.05, f1, f2, n1, n2, bystudy = TRUE)

# Should return:
# Estimate SE LL UL
# Average 0.84705608 0.2528742 0.35143178 1.3426804
# Study 1 0.03604257 0.3297404 -0.61023681 0.6823220
# Study 2 0.81008932 0.3442007 0.13546839 1.4847103
# Study 3 0.38746839 0.2065227 -0.01730864 0.7922454
# Study 4 1.49316811 0.6023296 0.31262374 2.6737125
# Study 5 1.50851199 0.9828420 -0.41782290 3.4348469

# exp(Estimate) exp(LL) exp(UL)
# Average 2.332769 1.4211008 3.829294
# Study 1 1.036700 0.5432222 1.978466
# Study 2 2.4211008 1.450730 4.413686
# Study 3 1.473246 0.9828403 2.208350
# Study 4 4.451175 1.3670071 14.493677
# Study 5 4.520000 0.6584788 31.026662

meta.ave.semipart

Confidence interval for an average semipartial correlation

Description

Computes the estimate, standard error, and confidence interval for an average semipartial correlation from two or more studies.

Usage

meta.ave.semipart(alpha, n, cor, r2, bystudy = TRUE)

Arguments

alpha alpha level for 1-alpha confidence
n vector of sample sizes
cor vector of sample semipartial correlations
r2 vector of squared multiple correlations for full model
bystudy logical to also return each study estimate (TRUE) or not
Value
Returns a matrix. The first row is the average estimate across all studies. If bystudy is true, there is 1 additional row for each study. The matrix has the following columns:

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

Examples

```r
n <- c(128, 97, 210, 217)
cor <- c(.35, .41, .44, .39)
r2 <- c(.29, .33, .36, .39)
meta.ave.semipart(.05, n, cor, r2, bystudy = TRUE)
```

# Should return:

<table>
<thead>
<tr>
<th>Estimate</th>
<th>SE</th>
<th>LL</th>
<th>UL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average</td>
<td>0.3975</td>
<td>0.33221240</td>
<td>0.4586965</td>
</tr>
<tr>
<td>Study 1</td>
<td>0.3500</td>
<td>0.2023485</td>
<td>0.4820930</td>
</tr>
<tr>
<td>Study 2</td>
<td>0.4100</td>
<td>0.2447442</td>
<td>0.5521076</td>
</tr>
<tr>
<td>Study 3</td>
<td>0.4400</td>
<td>0.3338366</td>
<td>0.5351410</td>
</tr>
<tr>
<td>Study 4</td>
<td>0.3900</td>
<td>0.2860431</td>
<td>0.4848830</td>
</tr>
</tbody>
</table>
```

Description
Computes the estimate, standard error, and confidence interval for an average slope coefficient in a simple linear regression model. A Satterthwaite adjustment to the degrees of freedom is used to improve the accuracy of the confidence interval.

Usage

```r
meta.ave.slope(alpha, n, cor, sdy, sdx, bystudy = TRUE)
```

Arguments

- **alpha**: alpha level for 1-alpha confidence
- **n**: vector of sample sizes
- **cor**: vector of sample correlations
- **sdy**: vector of SDs of y
- **sdx**: vector of SDs of x
- **bystudy**: logical to also return each study estimate (TRUE) or not
Value

Returns a matrix. The first row is the average estimate across all studies. If bystudy is true, there is 1 additional row for each study. The matrix has the following columns:

- Estimate - the estimated effect size
- SE - standard error
- LL - lower limit of the confidence interval
- UL - upper limit of the confidence interval
- df - degrees of freedom

Examples

```r
n <- c(45, 85, 50, 60)
cor <- c(.24, .35, .16, .20)
sdy <- c(12.2, 14.1, 11.7, 15.9)
sdx <- c(1.34, 1.87, 2.02, 2.37)
meta.ave.slope(.05, n, cor, sdy, sdx, bystudy = TRUE)
```

# Should return:
```
# Estimate      SE       LL        UL       df
# Average 1.7731542 0.4755417 0.8335021 2.712806 149.4777
# Study 1 2.1850746 1.3084468 -0.4536599 4.823809  43.0000
# Study 2 2.6390374 0.7262491 1.1945573 4.083518  83.0000
# Study 3 0.9267327 0.8146126 -0.7111558 2.564621  48.0000
# Study 4 1.3417722 0.8456799 -0.3510401 3.034584  58.0000
```

Confidence interval for an average Spearman correlation

Description

Computes the estimate, standard error, and confidence interval for an average Spearman correlation. The Spearman correlation is preferred to the Pearson correlation if the relation between the two quantitative variables is monotonic rather than linear or if the bivariate normality assumption is not plausible.

Usage

```r
meta.ave.spear(alpha, n, cor, bystudy = TRUE)
```

Arguments

- `alpha` - alpha level for 1-alpha confidence
- `n` - vector of sample sizes
- `cor` - vector of sample Spearman correlations
- `bystudy` - logical to also return each study estimate (TRUE) or not
Value

Returns a matrix. The first row is the average estimate across all studies. If bystudy is true, there is 1 additional row for each study. The matrix has the following columns:

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

References


Examples

```r
n <- c(150, 200, 300, 200, 350)
cor <- c(.14, .29, .16, .21, .23)
meta.ave.spear(.05, n, cor, bystudy = TRUE)
```

# Should return:

```
# Should return:
# Estimate SE LL UL
# Average 0.206 0.02944265 0.14763960 0.2629309
# Study 1 0.140 0.08031750 -0.02151639 0.2943944
# Study 2 0.290 0.06492643 0.15476515 0.4145671
# Study 3 0.160 0.05635101 0.04689807 0.2690514
# Study 4 0.210 0.06776195 0.07187439 0.3402225
# Study 5 0.230 0.05069710 0.12690280 0.3281809
```

**meta.ave.stdmean.ps**

*Confidence interval for an average standardized mean difference from paired-samples studies*

**Description**

Computes the estimate, standard error, and confidence interval for an average standardized mean difference from two or more paired-samples studies. Unweighted variances and single group variance are options for the standardizer. Equal variances within or across studies is not assumed.

**Usage**

`meta.ave.stdmean.ps(alpha, m1, m2, sd1, sd2, cor, n, stdzr, bystudy = TRUE)`
Arguments

- **alpha**: alpha level for 1-alpha confidence
- **m1**: vector of sample means for measurement 1
- **m2**: vector of sample means for measurement 2
- **sd1**: vector of sample SDs for measurement 1
- **sd2**: vector of sample SDs for measurement 2
- **cor**: vector of sample correlations for paired measurements
- **n**: vector of sample sizes
- **stdzr**: set to 0 for square root average variance standardizer set to 1 for measurement 1 SD standardizer set to 2 for measurement 2 SD standardizer
- **bystudy**: logical to also return each study estimate (TRUE) or not

Value

A matrix. First row is the overall average estimate. If bystudy is TRUE also returns 1 row per study. The matrix has the following columns:

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

References


Examples

```r
m1 <- c(23.9, 24.1)
m2 <- c(25.1, 26.9)
sd1 <- c(1.76, 1.58)
sd2 <- c(2.01, 1.76)
cor <- c(.78, .84)
n <- c(25, 30)
meta.ave.stdmean.ps(.05, m1, m2, sd1, sd2, cor, n, 1, bystudy = TRUE)
```

# Should return:

<table>
<thead>
<tr>
<th>Estimate</th>
<th>SE</th>
<th>LL</th>
<th>UL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average</td>
<td>-1.1931045</td>
<td>0.1568034</td>
<td>-1.500433</td>
</tr>
<tr>
<td>Study 1</td>
<td>-0.6818182</td>
<td>0.1773785</td>
<td>-1.029474</td>
</tr>
<tr>
<td>Study 2</td>
<td>-1.7721519</td>
<td>0.2586234</td>
<td>-2.279044</td>
</tr>
</tbody>
</table>
meta.ave.stdmean2

Confidence interval for an average standardized mean difference from 2-group studies

Description
Computes the estimate, standard error, and confidence interval for an average standardized mean difference from two or more 2-group studies. Unweighted variances, weighted variances, and single group variance are options for the standardizer. Equal variances within or across studies is not assumed.

Usage
meta.ave.stdmean2(alpha, m1, m2, sd1, sd2, n1, n2, stdzr, bystudy = TRUE)

Arguments
alpha alpha level for 1-alpha confidence
m1 vector of sample means for group 1
m2 vector of sample means for group 2
sd1 vector of sample SDs for group 1
sd2 vector of sample SDs for group 2
n1 vector of group 1 sample sizes
n2 vector of group 2 sample sizes
stdzr
• set to 0 for square root average variance standardizer
• set to 1 for group 1 SD standardizer
• set to 2 for group 2 SD standardizer
• set to 3 for square root weighted variance standardizer
bystudy logical to also return each study estimate (TRUE) or not

Value
A matrix. First row is the overall average estimate. If bystudy is TRUE also returns 1 row per study. The matrix has the following columns:
• Estimate - the estimated effect size
• SE - standard error
• LL - lower limit of the confidence interval
• UL - upper limit of the confidence interval

References
Examples

```r
m1 <- c(21.9, 23.1, 19.8)
m2 <- c(16.1, 17.4, 15.0)
sd1 <- c(3.82, 3.95, 3.67)
sd2 <- c(3.21, 3.30, 3.02)
n1 <- c(40, 30, 24)
n2 <- c(40, 28, 25)
meta.ave.stdmean2(.05, m1, m2, sd1, sd2, n1, n2, 0, bystudy = TRUE)
```

# Should return:
# Estimate  SE    LL  UL
# Average  1.526146 0.1734341 1.1862217 1.866071
# Study 1  1.643894 0.2629049 1.1286100 2.159178
# Study 2  1.566132 0.3056278 0.9671126 2.165152
# Study 3  1.428252 0.3289179 0.7835848 2.072919

---

**meta.lc.agree**

**Confidence interval for a linear contrast of G-index coefficients**

**Description**

Computes the estimate, standard error, and confidence interval for a linear contrast of G-index of agreement coefficients from two or more studies. This function assumes that two raters each provide a dichotomous rating for a sample of objects.

**Usage**

```r
meta.lc.agree(alpha, f11, f12, f21, f22, v)
```

**Arguments**

- **alpha**: alpha level for 1-alpha confidence
- **f11**: vector of frequencies in cell 1,1
- **f12**: vector of frequencies in cell 1,2
- **f21**: vector of frequencies in cell 2,1
- **f22**: vector of frequencies in cell 2,2
- **v**: vector of contrast coefficients

**Value**

Returns 1-row matrix with the estimate for the contrast. The matrix has the following columns:

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

```r
f11 <- c(17, 28, 19)
f12 <- c(43, 56, 49)
f21 <- c(3, 5, 5)
f22 <- c(37, 54, 39)
v <- c(.5, .5, -1)
meta.lc.agree(.05, f11, f12, f21, f22, v)
```

# Should return:
# Estimate SE LL UL
# Contrast 0.07692783 0.1138407 -0.1461958 0.3000515

---

**meta.lc.gen**  
*Confidence interval for a linear contrast of effect sizes*

**Description**

Computes the estimate, standard error, and confidence interval for a linear contrast of any type of effect size from two or more studies.

**Usage**

```r
meta.lc.gen(alpha, est, se, v)
```

**Arguments**

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

**Value**

Returns 1-row matrix with the estimate for the contrast. The matrix has the following columns:

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

```r
est <- c(.55, .59, .44, .48, .26, .19)
se <- c(.054, .098, .029, .084, .104, .065)
v <- c(.5, .5, -.25, -.25, -.25, -.25)
meta.lc.gen(.05, est, se, v)
```

# Should return:
# Contrast 0.2275 0.06755461 0.0950954 0.3599046

---

**Description**

Computes the estimate, standard error, and confidence interval for a linear contrast of paired-samples mean differences from two or more studies. A Satterthwaite adjustment to the degrees of freedom is used to improve the accuracy of the confidence interval. Equal variances within or across studies is not assumed.

**Usage**

```r
meta.lc.mean.ps(alpha, m1, m2, sd1, sd2, cor, n, v)
```

**Arguments**

- `alpha`: alpha level for 1-alpha confidence
- `m1`: vector of sample means for group 1
- `m2`: vector of sample means for group 2
- `sd1`: vector of sample SDs for group 1
- `sd2`: vector of sample SDs for group 2
- `cor`: vector of sample correlations for paired measurements
- `n`: vector of sample sizes
- `v`: vector of contrast coefficients

**Value**

Returns 1-row matrix with the estimate for the contrast. The matrix has the following columns:

- **Estimate** - estimate of linear contrast
- **SE** - standard error
- **LL** - lower limit of the confidence interval
- **UL** - upper limit of the confidence interval
- **df** - degrees of freedom
References


Examples

```r
m1 <- c(53, 60, 53, 57)
m2 <- c(55, 62, 58, 61)
sd1 <- c(4.1, 4.2, 4.5, 4.0)
sd2 <- c(4.2, 4.7, 4.9, 4.8)
cor <- c(.7, .7, .8, .85)
n <- c(30, 50, 30, 70)
v <- c(.5, .5, -.5, -.5)
meta.lc.mean.ps(.05, m1, m2, sd1, sd2, cor, n, v)
```

# Should return:

# Estimate SE LL UL df
# Contrast 2.5 0.4943114 1.520618 3.479382 112.347

---

**meta.lc.mean1**  
Confidence interval a for a linear contrast of means

**Description**

Computes the estimate, standard error, and confidence interval for a linear contrast of means from two or more studies. This function will use either an unequal variance (recommended) or an equal variance method. A Satterthwaite adjustment to the degrees of is used with the unequal variance method.

**Usage**

```r
meta.lc.mean1(alpha, m, sd, n, v, eqvar = FALSE)
```

**Arguments**

- `alpha` alpha level for 1-alpha confidence
- `m` vector of sample means
- `sd` vector of sample standard deviations
- `n` vector of sample sizes
- `v` vector of contrast coefficients
- `eqvar`  
  - FALSE for unequal variance method
  - TRUE for equal variance method
Value

Returns 1-row matrix with the estimate for the contrast. The matrix has the following columns:

• Estimate - estimate of linear contrast
• SE - standard error
• LL - lower limit of the confidence interval
• UL - upper limit of the confidence interval
• df - degrees of freedom

References


Examples

```r
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)
meta.lc.mean1(.05, m, sd, n, v, eqvar = FALSE)
```

# Should return:
# Estimate SE LL UL df
# Contrast -5.35 1.30136 -7.993583 -2.706417 33.52169

Description

Computes the estimate, standard error, and confidence interval for a linear contrast of 2-group mean differences from two or more studies. A Satterthwaite adjustment to the degrees of freedom is used to improve the accuracy of the confidence interval. Equal variances within or across studies is not assumed.

Usage

```r
meta.lc.mean2(alpha, m1, m2, sd1, sd2, n1, n2, v)
```
Arguments

alpha  alpha level for 1-alpha confidence
m1  vector of sample means for group 1
m2  vector of sample means for group 2
sd1  vector of sample SDs for group 1
sd2  vector of sample SDs for group 2
n1  vector of group 1 sample sizes
n2  vector of group 2 sample sizes
v  vector of contrast coefficients

Value

Returns 1-row matrix with the estimate for the contrast. The matrix has the following columns:

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

References


Examples

```r
m1 <- c(45.1, 39.2, 36.3, 34.5)
m2 <- c(30.0, 35.1, 35.3, 36.2)
sd1 <- c(10.7, 10.5, 9.4, 11.5)
sd2 <- c(12.3, 12.0, 10.4, 9.6)
n1 <- c(40, 20, 50, 25)
n2 <- c(40, 20, 48, 26)
v <- c(.5, .5, -.5, -.5)
meta.lc.mean2(.05, m1, m2, sd1, sd2, n1, n2, v)
```

# Should return:
# Contrast 9.95 2.837787 4.343938 15.55606 153.8362
Confidence interval for a log-linear contrast of mean ratios from paired-samples studies

Description

Computes the estimate, standard error, and confidence interval for a log-linear contrast of paired-sample mean ratios from two or more studies. A Satterthwaite adjustment to the degrees of freedom is used to improve the accuracy of the confidence interval. Equal variances within or across studies is not assumed.

Usage

```r
meta.lc.meanratio.ps(alpha, m1, m2, sd1, sd2, cor, n, v)
```

Arguments

- `alpha`: alpha level for 1-alpha confidence
- `m1`: vector of sample means for group 1
- `m2`: vector of sample means for group 2
- `sd1`: vector of sample SDs for group 1
- `sd2`: vector of sample SDs for group 2
- `cor`: vector of sample correlations for paired measurements
- `n`: vector of sample sizes
- `v`: vector of contrast coefficients

Value

Returns 1-row matrix with the estimate for the contrast. The matrix has the following columns:

- **Estimate**: estimate of log-linear contrast
- **SE**: standard error of log-linear contrast
- **LL**: lower limit of the confidence interval
- **UL**: upper limit of the confidence interval
- **exp(Estimate)**: exponentiated estimate of log-linear function
- **exp(LL)**: lower limit of the exponentiated confidence interval
- **exp(UL)**: upper limit of the exponentiated confidence interval
- **df**: degrees of freedom

References

Examples

m1 <- c(53, 60, 53, 57)
m2 <- c(55, 62, 58, 61)
sd1 <- c(4.1, 4.2, 4.5, 4.0)
sd2 <- c(4.2, 4.7, 4.9, 4.8)
cor <- c(.7, .7, .8, .85)
n <- c(30, 50, 30, 70)
v <- c(.5, .5, -.5, -.5)
meta.lc.meanratio.ps(.05, m1, m2, sd1, sd2, cor, n, v)

# Should return:
# Estimate SE LL UL exp(Estimate)
# Contrast 0.0440713 0.008701725 0.02681353 0.06132907 1.045057
# exp(LL) exp(UL) df
# Contrast 1.027176 1.063249 103.0256

---

### Description

Computes the estimate, standard error, and confidence interval for a log-linear contrast of 2-group mean ratio from two or more studies. A Satterthwaite adjustment to the degrees of freedom is used to improve the accuracy of the confidence interval. Equal variances within or across studies is not assumed.

### Usage

```r
meta.lc.meanratio2(alpha, m1, m2, sd1, sd2, n1, n2, v)
```

### Arguments

- **alpha**: alpha level for 1-alpha confidence
- **m1**: vector of sample means for group 1
- **m2**: vector of sample means for group 2
- **sd1**: vector of sample SDs for group 1
- **sd2**: vector of sample SDs for group 2
- **n1**: vector of group 1 sample sizes
- **n2**: vector of group 2 sample sizes
- **v**: vector of contrast coefficients
Value

Returns 1-row matrix with the estimate for the contrast. The matrix has the following columns:

- Estimate - estimate of log-linear contrast
- SE - standard error of log-linear contrast
- LL - lower limit of the confidence interval
- UL - upper limit of the confidence interval
- exp(Estimate) - the exponentiated estimate
- exp(LL) - lower limit of the exponentiated confidence interval
- exp(UL) - upper limit of the exponentiated confidence interval
- df - degrees of freedom

References


Examples

```r
m1 <- c(45.1, 39.2, 36.3, 34.5)
m2 <- c(30.0, 35.1, 35.3, 36.2)
sd1 <- c(10.7, 10.5, 9.4, 11.5)
sd2 <- c(12.3, 12.0, 10.4, 9.6)
n1 <- c(40, 20, 50, 25)
n2 <- c(40, 20, 48, 26)
v <- c(.5, .5, -.5, -.5)
meta.lc.meanratio2(.05, m1, m2, sd1, sd2, n1, n2, v)
```

# Should return:
# Estimate SE LL UL exp(Estimate)
# Contrast 0.2691627 0.07959269 0.1119191 0.4264064 1.308868
# exp(LL) exp(UL) df
# Contrast 1.118422 1.531743 152.8665
```

——

meta.lc.odds

Confidence interval for a log-linear contrast of odds ratios

Description

Computes the estimate, standard error, and confidence interval for an exponentiated log-linear contrast of odds ratios from two or more studies.
Usage

```r
meta.lc.odds(alpha, f1, f2, n1, n2, v)
```

Arguments

- `alpha`: alpha level for 1-alpha confidence
- `f1`: vector of group 1 event counts
- `f2`: vector of group 2 event counts
- `n1`: vector of group 1 sample sizes
- `n2`: vector of group 2 sample sizes
- `v`: vector of contrast coefficients

Value

Returns 1-row matrix with the estimate for the contrast. The matrix has the following columns:

- **Estimate**: estimate of log-linear contrast
- **SE**: standard error of log-linear contrast
- **exp(Estimate)**: the exponentiated contrast
- **exp(LL)**: lower limit of the exponentiated confidence interval
- **exp(UL)**: upper limit of the exponentiated confidence interval

References


Examples

```r
n1 <- c(50, 150, 150)
f1 <- c(16, 50, 25)
n2 <- c(50, 150, 150)
f2 <- c(7, 15, 20)
v <- c(1, -1, 0)
meta.lc.odds(.05, f1, f2, n1, n2, v)
```

# Should return:
# | Estimate | SE | exp(Estimate) | exp(LL) | exp(UL) |
# |----------|----|---------------|---------|---------|
# | Contrast | -0.4596883 | 0.5895438 | 0.6314805 | 0.1988563 | 2.005305 |
**Description**

Computes the estimate, standard error, and confidence interval for a linear contrast of paired-samples group proportion differences from two or more studies.

**Usage**

```r
meta.lc.prop.ps(alpha, f11, f12, f21, f22, v)
```

**Arguments**

- `alpha`: alpha level for 1-alpha confidence
- `f11`: vector of frequencies in cell 1,1
- `f12`: vector of frequencies in cell 1,2
- `f21`: vector of frequencies in cell 2,1
- `f22`: vector of frequencies in cell 2,2
- `v`: vector of contrast coefficients

**Value**

Returns 1-row matrix with the estimate for the contrast. The matrix has the following columns:

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

**References**


**Examples**

```r
f11 <- c(17, 28, 19)
f12 <- c(43, 56, 49)
f21 <- c(3, 5, 5)
f22 <- c(37, 54, 39)
v <- c(.5, .5, -1)
meta.lc.prop.ps(.05, f11, f12, f21, f22, v)
```
# Should return:
# Estimate    SE     LL     UL
# Contrast  -0.01436285 0.06511285 -0.1419817 0.113256

---

**meta.lc.prop1**  
*Confidence interval for a linear contrast of proportions.*

---

**Description**

Computes the estimate, standard error, and an adjusted Wald confidence interval for a linear contrast of proportions from two or more studies.

**Usage**

```r
meta.lc.prop1(alpha, f, n, v)
```

**Arguments**

- `alpha`  
  alpha level for 1-alpha confidence
- `f`  
  vector of sample frequency counts
- `n`  
  vector of sample sizes
- `v`  
  vector of contrast coefficients

**Value**

Returns 1-row matrix with the estimate for the contrast. The matrix has the following columns:

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

**References**

Examples

```r
f <- c(26, 24, 38)
n <- c(60, 60, 60)
v <- c(-.5, -.5, 1)
meta.lc.prop1(.05, f, n, v)
```

# Should return:

<table>
<thead>
<tr>
<th>Estimate</th>
<th>SE</th>
<th>LL</th>
<th>UL</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.2119565</td>
<td>0.07602892</td>
<td>0.06294259</td>
<td>0.3609705</td>
</tr>
</tbody>
</table>

---

**meta.lc.prop2**  
*Confidence interval for a linear contrast of proportion differences in 2-group studies*

---

**Description**

Computes the estimate, standard error, and confidence interval for a linear contrast of 2-group proportion differences from two or more studies.

**Usage**

```r
meta.lc.prop2(alpha, f1, f2, n1, n2, v)
```

**Arguments**

- **alpha**: alpha level for 1-alpha confidence
- **f1**: vector of group 1 event counts
- **f2**: vector of group 2 event counts
- **n1**: vector of group 1 sample sizes
- **n2**: vector of group 2 sample sizes
- **v**: vector of contrast coefficients

**Value**

Returns 1-row matrix with the estimate for the contrast. The matrix has the following columns:

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


Examples

```r
n1 <- c(50, 150, 150)
n2 <- c(50, 150, 150)
f1 <- c(16, 50, 25)
f2 <- c(7, 15, 20)
v <- c(1, -1, 0)
meta.lc.propratio2(.05, f1, f2, n1, n2, v)
```

# Should return:
# Estimate  SE  LL  UL
# Contrast -0.05466931 0.09401019 -0.2389259 0.1295873

---

**meta.lc.propratio2**

*Confidence interval for a log-linear contrast of proportion ratios from 2-group studies*

**Description**

Computes the estimate, standard error, and confidence interval for an exponentiated log-linear contrast of 2-group proportion odds ratios from two or more studies.

**Usage**

```r
meta.lc.propratio2(alpha, f1, f2, n1, n2, v)
```

**Arguments**

- `alpha` alpha level for 1-alpha confidence
- `f1` vector of group 1 event counts
- `f2` vector of group 2 event counts
- `n1` vector of group 1 sample sizes
- `n2` vector of group 2 sample sizes
- `v` vector of contrast coefficients
Value

Returns 1-row matrix with the estimate for the contrast. The matrix has the following columns:

- **Estimate** - estimate of log-linear contrast
- **SE** - standard error of log-linear contrast
- **exp(Estimate)** - the exponentiated contrast
- **exp(LL)** - lower limit of the exponentiated confidence interval
- **exp(UL)** - upper limit of the exponentiated confidence interval

References


Examples

```r
n1 <- c(50, 150, 150)
f1 <- c(16, 50, 25)
n2 <- c(50, 150, 150)
f2 <- c(7, 15, 20)
v <- c(1, -1, 0)
meta.lc.propratio2(.05, f1, f2, n1, n2, v)
```

# Should return:
# Estimate   SE  exp(Estimate)  exp(LL)  exp(UL)
# Contrast  -0.3853396 0.4828218 0.6802196 0.2640405 1.752378

Description

Confidence interval for a linear contrast of standardized mean differences from paired-samples studies

Computes the estimate, standard error, and confidence interval for a linear contrast of paired-samples standardized mean differences from two or more studies. Equal variances within or across studies is not assumed. This function has an option to use a square root average variance standardizer or a single group standard deviation standardizer.

Usage

```r
meta.lc.stdmean.ps(alpha, m1, m2, sd1, sd2, cor, n, v, stdzr)
```
Arguments

alpha alpha level for 1-alpha confidence
m1 vector of sample means for group 1
m2 vector of sample means for group 2
sd1 vector of sample SDs for group 1
sd2 vector of sample SDs for group 2
cor vector of sample correlations for paired measurements
n vector of sample sizes
v vector of contrast coefficients
stdzr
  • set to 0 for square root average variance standardizer
  • set to 1 for group 1 SD standardizer
  • set to 2 for group 2 SD standardizer

Value

Returns 1-row matrix with the estimate for the contrast. The matrix has the following columns:

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

References


Examples

```r
m1 <- c(53, 60, 53, 57)
m2 <- c(55, 62, 58, 61)
sd1 <- c(4.1, 4.2, 4.5, 4.0)
sd2 <- c(4.2, 4.7, 4.9, 4.8)
cor <- c(.7, .7, .8, .85)
n <- c(30, 50, 30, 70)
v <- c(.5, .5, -.5, -.5)
meta.lc.stdmean.ps(.05, m1, m2, sd1, sd2, cor, n, v, 0)
```

# Should return:
```
# Estimate   SE    LL    UL
# Contrast  0.5127577 0.1392232 0.2398851 0.7856302
```
Confidence interval for a linear contrast of standardized mean differences from 2-group studies

Description
Computes the estimate, standard error, and confidence interval for a linear contrast of 2-group standardized mean differences from two or more studies. Equal variances within or across studies is not assumed. Use the square root average variance standardizer (stdzr = 0) for 2-group experimental designs. Use the square root weighted variance standardizer (stdzr = 3) for 2-group nonexperimental designs with simple random sampling. The stdzr = 1 and stdzr = 2 options can be used with either 2-group experimental or nonexperimental designs.

Usage
meta.lc.stdmean2(alpha, m1, m2, sd1, sd2, n1, n2, v, stdzr)

Arguments
- alpha: alpha level for 1-alpha confidence
- m1: vector of sample means for group 1
- m2: vector of sample means for group 2
- sd1: vector of sample SDs for group 1
- sd2: vector of sample SDs for group 2
- n1: vector of group 1 sample sizes
- n2: vector of group 2 sample sizes
- v: vector of contrast coefficients
- stdzr: set to 0 for square root average variance standardizer
  - set to 1 for group 1 SD standardizer
  - set to 2 for group 2 SD standardizer
  - set to 3 for square root weighted variance standardizer

Value
Returns 1-row matrix with the estimate for the contrast. The matrix has the following columns:
- Estimate - estimate of linear contrast
- SE - standard error
- LL - lower limit of the confidence interval
- UL - upper limit of the confidence interval
References


Examples

```r
m1 <- c(45.1, 39.2, 36.3, 34.5)
m2 <- c(30.0, 35.1, 35.3, 36.2)
sd1 <- c(10.7, 10.5, 9.4, 11.5)
sd2 <- c(12.3, 12.0, 10.4, 9.6)
n1 <- c(40, 20, 50, 25)
n2 <- c(40, 20, 48, 26)
v <- c(.5, .5, -.5, -.5)
meta.lc.stdmean2(.05, m1, m2, sd1, sd2, n1, n2, v, 0)
```

# Should return:

<table>
<thead>
<tr>
<th>Estimate</th>
<th>SE</th>
<th>LL</th>
<th>UL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contrast</td>
<td>0.8557914</td>
<td>0.2709192</td>
<td>0.3247995</td>
</tr>
</tbody>
</table>

---

**meta.lm.agree**  
*Meta-regression analysis for G agreement indices*

Description

This function estimates the intercept and slope coefficients in a meta-regression model where the dependent variable is a G-index of agreement. The estimates are OLS estimates with standard errors that accommodate residual heteroscedasticity.

Usage

```r
meta.lm.agree(alpha, f11, f12, f21, f22, X)
```

Arguments

- **alpha**: alpha level for 1-alpha confidence
- **f11**: vector of frequencies in cell 1,1
- **f12**: vector of frequencies in cell 1,2
- **f21**: vector of frequencies in cell 2,1
- **f22**: vector of frequencies in cell 2,2
- **X**: matrix of predictor values
Value

Returns a matrix. The first row is for the intercept with one additional row per predictor. The matrix has the following columns:

- Estimate - OLS estimate
- SE - standard error
- z - z-value
- p - p-value
- LL - lower limit of the confidence interval
- UL - upper limit of the confidence interval

Examples

```r
f11 <- c(40, 20, 25, 30)
f12 <- c(3, 2, 2, 1)
f21 <- c(7, 6, 8, 6)
f22 <- c(26, 25, 13, 25)
x1 <- c(1, 1, 4, 6)
x2 <- c(1, 1, 0, 0)
X <- matrix(cbind(x1, x2), 4, 2)
meta.lm.agree(.05, f11, f12, f21, f22, X)
```

# Should return:

<table>
<thead>
<tr>
<th>Estimate</th>
<th>SE</th>
<th>z</th>
<th>p</th>
<th>LL</th>
<th>UL</th>
</tr>
</thead>
<tbody>
<tr>
<td>b0</td>
<td>0.1904762</td>
<td>0.38772858</td>
<td>0.4912617</td>
<td>0.623</td>
<td>-0.56945786</td>
</tr>
<tr>
<td>b1</td>
<td>0.0952381</td>
<td>0.07141957</td>
<td>1.3335013</td>
<td>0.182</td>
<td>-0.04474169</td>
</tr>
<tr>
<td>b2</td>
<td>0.4205147</td>
<td>0.32383556</td>
<td>1.2985438</td>
<td>0.194</td>
<td>-0.21419136</td>
</tr>
</tbody>
</table>

Description

This function estimates the intercept and slope coefficients in a meta-regression model where the dependent variable is a Fisher- transformed Pearson or partial correlation. The estimates are OLS estimates with standard errors that accommodate residual heteroscedasticity. The correlations are Fisher-transformed and hence the parameter estimates do not have a simple interpretation. However, the hypothesis test results can be used to decide if a population slope is either positive or negative.

Usage

```r
meta.lm.cor(alpha, n, cor, q, X)
```
Arguments

- **alpha**: alpha level for 1-alpha confidence
- **n**: vector of sample sizes
- **cor**: vector of Pearson or partial correlations
- **q**: number of control variables
- **X**: matrix of predictor values

Value

Returns a matrix. The first row is for the intercept with one additional row per predictor. The matrix has the following columns:

- Estimate - OLS estimate
- SE - Standard error
- z - z-value
- p - p-value
- LL - lower limit of the confidence interval
- UL - upper limit of the confidence interval

Examples

```r
n <- c(55, 190, 65, 35)
cor <- c(.40, .65, .60, .45)
q <- 0
x1 <- c(18, 25, 23, 19)
X <- matrix(x1, 4, 1)
meta.lm.cor(.05, n, cor, q, X)
```

# Should return:

```
# Estimate  SE    z     p   LL      UL
# b0 -0.47832153 0.48631509 -0.983563 0.325 -1.431481595 0.47483852
# b1  0.05047154 0.02128496 2.371231 0.018 0.008753794 0.09218929
```

Description

This function estimates the intercept and slope coefficients in a meta-regression model where the dependent variable is a Fisher-transformed correlation. The correlations can be of different types (e.g., Pearson, partial, Spearman). The estimates are OLS estimates with standard errors that accommodate residual heteroscedasticity. This function uses sample correlations and their standard errors as input. The correlations are Fisher-transformed and hence the parameter estimates do not have a simple interpretation. However, the hypothesis test results can be used to decide if a population slope is either positive or negative.
Usage

meta.lm.cor.gen(alpha, cor, se, X)

Arguments

alpha alpha level for 1-alpha confidence
cor vector of sample correlations
se number of control variables
X matrix of predictor values

Value

Returns a matrix. The first row is for the intercept with one additional row per predictor. The matrix has the following columns:

- Estimate - OLS estimate
- SE - standard error
- z - z-value
- p - p-value
- LL - lower limit of the confidence interval
- UL - upper limit of the confidence interval

Examples

cor <- c(.40, .65, .60, .45)
se <- c(.182, .114, .098, .132)
x1 <- c(18, 25, 23, 19)
X <- matrix(x1, 4, 1)
meta.lm.cor.gen(.05, cor, se, X)
# Should return:
# Estimate SE z p
# b0 -0.4783215 0.6342793 -0.7541181 0.451
# b1  0.0504715 0.0287986  1.7525699 0.080

meta.lm.cronbach Meta-regression analysis for Cronbach reliabilities

Description

This function estimates the intercept and slope coefficients in a meta-regression model where the dependent variable is a log-complement Cronbach reliability. The estimates are OLS estimates with standard errors that accommodate residual heteroscedasticity. The exponentiated slope estimate for a predictor variable describes a multiplicative change in variable.
Usage

meta.lm.cronbach(alpha, n, rel, q, X)

Arguments

alpha alpha level for 1-alpha confidence
n vector of sample sizes
rel vector of sample alpha reliabilities
q number of measurements (e.g., items)
X matrix of predictor values

Value

Returns a matrix. The first row is for the intercept with one additional row per predictor. The matrix has the following columns:

- Estimate - exponentiated OLS estimate
- SE - standard error
- z - z-value
- p - p-value
- LL - lower limit of the exponentiated confidence interval
- UL - upper limit of the exponentiated confidence interval

References


Examples

```R
n <- c(583, 470, 546, 680)
rel <- c(.91, .89, .90, .89)
x1 <- c(1, 0, 0, 0)
X <- matrix(x1, 4, 1)
meta.lm.cronbach(.05, n, rel, 10, X)
# Should return:
# Estimate   SE  z   p   LL  UL
# b0 -2.2408328 0.03675883 -60.960391 0.000 -2.3128788 -2.16878684
# b1 -0.1689006 0.07204625 -2.344336 0.019 -0.3101087 -0.02769259
```
Description

This function estimates the intercept and slope coefficients in a meta-regression model where the dependent variable is any type of effect size. The estimates are OLS estimates with standard errors that accommodate residual heteroscedasticity.

Usage

meta.lm.gen(alpha, est, se, X)

Arguments

alpha  alpha level for 1-alpha confidence
est    vector of parameter estimates
se     vector of standard errors
X      matrix of predictor values

Value

Returns a matrix. The first row is for the intercept with one additional row per predictor. The matrix has the following columns:

- Estimate - OLS estimate
- SE - standard error
- z - z-value
- p - p-value
- LL - lower limit of the confidence interval
- UL - upper limit of the confidence interval

Examples

est <- c(4.1, 4.7, 4.9, 5.7, 6.6, 7.3)
se <- c(1.2, 1.5, 1.3, 1.8, 2.0, 2.6)
x1 <- c(10, 20, 30, 40, 50, 60)
x2 <- c(1, 1, 1, 0, 0, 0)
X <- matrix(cbind(x1, x2), 6, 2)
meta.lm.gen(.05, est, se, X)

# Should return:
# Estimate SE    z    p    LL    UL
# b0  3.5333333 4.37468253 0.80767766 0.419 -5.0408869 12.1075535
# b1  0.0600000 0.09058835 0.66233679 0.508 -0.1175499 0.2375499
# b2 -0.1666667 2.81139793 -0.05928249 0.953 -5.6769054 5.3435720
Meta-regression analysis for paired-samples mean differences

Description

This function estimates the intercept and slope coefficients in a meta-regression model where the dependent variable is a paired-samples mean difference. The estimates are OLS estimates with standard errors that accommodate residual heteroscedasticity.

Usage

meta.lm.mean.ps(alpha, m1, m2, sd1, sd2, cor, n, X)

Arguments

- `alpha` alpha level for 1-alpha confidence
- `m1` vector of sample means for group 1
- `m2` vector of sample means for group 2
- `sd1` vector of sample SDs for group 1
- `sd2` vector of sample SDs for group 2
- `cor` vector of correlations
- `n` vector of sample sizes
- `X` matrix of predictor values

Value

Returns a matrix. The first row is for the intercept with one additional row per predictor. The matrix has the following columns:

- Estimate - OLS estimate
- SE - standard error
- t - t-value
- p - p-value
- LL - lower limit of the confidence interval
- UL - upper limit of the confidence interval
- df - degrees of freedom

References

Examples

\[ n \leftarrow c(65, 30, 29, 45, 50) \]
\[ cor \leftarrow c(.87, .92, .85, .90, .88) \]
\[ m1 \leftarrow c(20.1, 20.5, 19.3, 21.5, 19.4) \]
\[ m2 \leftarrow c(10.4, 10.2, 8.5, 10.3, 7.8) \]
\[ sd1 \leftarrow c(9.3, 9.9, 10.1, 10.5, 9.8) \]
\[ sd2 \leftarrow c(7.8, 8.0, 8.4, 8.1, 8.7) \]
\[ x1 \leftarrow c(2, 3, 3, 4, 4) \]
\[ X \leftarrow \text{matrix}(x1, 5, 1) \]
\[ \text{meta.lm.mean.ps}(.05, m1, m2, sd1, sd2, cor, n, X) \]

# Should return:
# Estimate SE t p LL UL df
# b0 8.00 1.2491990 6.404104 0.000 5.5378833 10.462117 217
# b1 0.85 0.3796019 2.239188 0.026 0.1018213 1.598179 217

---

**meta.lm.mean1 Meta-regression analysis for single means**

**Description**

This function estimates the intercept and slope coefficients in a meta-regression model where the dependent variable is a mean from a single group. The estimates are OLS estimates with standard errors that accommodate residual heteroscedasticity.

**Usage**

\[ \text{meta.lm.mean1}(\alpha, m, sd, n, X) \]

**Arguments**

- \( \alpha \) alpha level for \( 1-\alpha \) confidence
- \( m \) vector of sample means
- \( sd \) vector of sample standard deviations
- \( n \) vector of sample sizes
- \( X \) matrix of predictor values

**Value**

Returns a matrix. The first row is for the intercept with one additional row per predictor. The matrix has the following columns:

- Estimate - OLS estimate
- SE - standard error
- t - t-value
• p - p-value
• LL - lower limit of the confidence interval
• UL - upper limit of the confidence interval
• df - degrees of freedom

Examples

```r
n <- c(25, 15, 30, 25, 40)
m <- c(20.1, 20.5, 19.3, 21.5, 19.4)
sd <- c(10.4, 10.2, 8.5, 10.3, 7.8)
x1 <- c(1, 1, 0, 0, 0)
x2 <- c(12, 13, 11, 13, 15)
X <- matrix(cbind(x1, x2), 5, 2)
meta.lm.mean1(.05, m, sd, n, X)
```

# Should return:

```
  Estimate   SE    t   p     LL    UL  df
# b0   19.45490196 6.7873381 2.86635227 0.005 6.0288763 32.880928 132
# b1    0.25686275 1.9834765 0.12950128 0.897 -3.6666499 4.180375 132
# b2    0.04705882 0.5064693 0.09291544 0.926 -0.9547876 1.048905 132
```

### meta.lm.mean2

*Meta-regression analysis for 2-group mean differences*

**Description**

This function estimates the intercept and slope coefficients in a meta-regression model where the dependent variable is a 2-group mean difference. The estimates are OLS estimates with standard errors that accommodate residual heteroscedasticity.

**Usage**

```r
meta.lm.mean2(alpha, m1, m2, sd1, sd2, n1, n2, X)
```

**Arguments**

- `alpha` : alpha level for 1-alpha confidence
- `m1` : vector of sample means for group 1
- `m2` : vector of sample means for group 2
- `sd1` : vector of sample SDs for group 1
- `sd2` : vector of sample SDs for group 2
- `n1` : vector of group 1 sample sizes
- `n2` : vector of group 2 sample sizes
- `X` : matrix of predictor values
Value

Returns a matrix. The first row is for the intercept with one additional row per predictor. The matrix has the following columns:

- **Estimate** - OLS estimate
- **SE** - standard error
- **t** - t-value
- **p** - p-value
- **LL** - lower limit of the confidence interval
- **UL** - upper limit of the confidence interval
- **df** - degrees of freedom

References


Examples

```r
n1 <- c(65, 30, 29, 45, 50)
n2 <- c(67, 32, 31, 20, 52)
m1 <- c(31.1, 32.3, 31.9, 29.7, 33.0)
m2 <- c(34.1, 33.2, 30.6, 28.7, 26.5)
sd1 <- c(7.1, 8.1, 7.8, 6.8, 7.6)
sd2 <- c(7.8, 7.3, 7.5, 7.2, 6.8)
x1 <- c(4, 6, 7, 7, 8)
x2 <- c(1, 0, 0, 0, 1)
X <- matrix(cbind(x1, x2), 5, 2)
meta.lm.mean2(.05, m1, m2, sd1, sd2, n1, n2, X)
# Should return:
# Estimate SE t p LL UL df
# b0 -15.20 3.4097610 -4.457791 0.000 -21.902415 -8.497585 418
# b1 2.35 0.4821523 4.873979 0.000 1.402255 3.297745 418
# b2 2.85 1.5358109 1.855697 0.064 -0.168875 5.868875 418
```

---

**meta.lm.meanratio.ps**  
*Meta-regression analysis for paired-samples log mean ratios*

Description

This function estimates the intercept and slope coefficients in a meta-regression model where the dependent variable is a paired-samples log mean ratio. The estimates are OLS estimates with standard errors that accommodate residual heteroscedasticity.
Usage

meta.lm.meanratio.ps(alpha, m1, m2, sd1, sd2, cor, n, X)

Arguments

alpha   alpha level for 1-alpha confidence
m1      vector of sample means for group 1
m2      vector of sample means for group 2
sd1     vector of sample SDs for group 1
sd2     vector of sample SDs for group 2
cor     vector of correlations
n       vector of sample sizes
X       matrix of predictor values

Value

Returns a matrix. The first row is for the intercept with one additional row per predictor. The matrix has the following columns:

- Estimate - OLS estimate
- SE - standard error
- z - z-value
- p - p-value
- LL - lower limit of the confidence interval
- UL - upper limit of the confidence interval
- exp(Estimate) - the exponentiated estimate
- exp(LL) - lower limit of the exponentiated confidence interval
- exp(UL) - upper limit of the exponentiated confidence interval

Examples

n <- c(65, 30, 29, 45, 50)
cor <- c(.87, .92, .85, .90, .88)
m1 <- c(20.1, 20.5, 19.3, 21.5, 19.4)
m2 <- c(10.4, 10.2, 8.5, 10.3, 7.8)
sd1 <- c(9.3, 9.9, 10.1, 10.5, 9.8)
sd2 <- c(7.8, 8.0, 8.4, 8.1, 8.7)
x1 <- c(2, 3, 3, 4, 4)
X <- matrix(x1, 5, 1)
meta.lm.meanratio.ps(.05, m1, m2, sd1, sd2, cor, n, X)

# Should return:
# Estimate SE LL UL z p
# b0 0.50957008 0.13000068 0.254773424 0.7643667 3.919749 0.000
# b1 0.07976238 0.04133414 -0.001251047 0.1607758 1.929697 0.054
Description

This function estimates the intercept and slope coefficients in a meta-regression model where the dependent variable is a 2-group log mean ratio. The estimates are OLS estimates with standard errors that accommodate residual heteroscedasticity.

Usage

`meta.lm.meanratio2(alpha, m1, m2, sd1, sd2, n1, n2, X)`

Arguments

- `alpha`: alpha level for 1-alpha confidence
- `m1`: vector of sample means for group 1
- `m2`: vector of sample means for group 2
- `sd1`: vector of sample SDs for group 1
- `sd2`: vector of sample SDs for group 2
- `n1`: vector of group 1 sample sizes
- `n2`: vector of group 2 sample sizes
- `X`: matrix of predictor values

Value

Returns a matrix. The first row is for the intercept with one additional row per predictor. The matrix has the following columns:

- Estimate - OLS estimate
- SE - standard error
- z - z-value
- p - p-value
- LL - lower limit of the confidence interval
- UL - upper limit of the confidence interval
- exp(Estimate) - the exponentiated estimate
- exp(LL) - lower limit of the exponentiated confidence interval
- exp(UL) - upper limit of the exponentiated confidence interval
Examples

n1 <- c(65, 30, 29, 45, 50)
n2 <- c(67, 32, 31, 20, 52)
m1 <- c(31.1, 32.3, 31.9, 29.7, 33.0)
m2 <- c(34.1, 33.2, 30.6, 28.7, 26.5)
sd1 <- c(7.1, 8.1, 7.8, 6.8, 7.6)
sd2 <- c(7.8, 7.3, 7.5, 7.2, 6.8)
x1 <- c(4, 6, 7, 7, 8)
X <- matrix(x1, 5, 1)
meta.lm.meanratio2(.05, m1, m2, sd1, sd2, n1, n2, X)

# Should return:
# Estimate SE LL UL z p
# b0 -0.40208954 0.09321976 -0.58479692 -0.21938216 -4.313351 0
# b1 0.06831545 0.01484125 0.03922712 0.09740377 4.603078 0
# exp(Estimate) exp(LL) exp(UL)
# b0 0.6689208 0.557219 0.8030148
# b1 1.0707030 1.040007 1.1023054

meta.lm.odds

Meta-regression analysis for odds ratios

Description

This function estimates the intercept and slope coefficients in a meta-regression model where the dependent variable is a log odds ratio. The estimates are OLS estimates with standard errors that accommodate residual heteroscedasticity. The exponentiated slope estimate for a predictor variable describes a multiplicative change in the odds ratio associated with a 1-unit increase in that predictor variable.

Usage

meta.lm.odds(alpha, f1, f2, n1, n2, X)

Arguments

alpha alpha level for 1-alpha confidence
f1 vector of group 1 event counts
f2 vector of group 2 event counts
n1 vector of group 1 sample sizes
n2 vector of group 2 sample sizes
X matrix of predictor values
meta.lm.odds

Value

Returns a matrix. The first row is for the intercept with one additional row per predictor. The matrix has the following columns:

- Estimate - OLS estimate
- SE - standard error
- z - z-value
- p - p-value
- LL - lower limit of the confidence interval
- UL - upper limit of the confidence interval
- exp(Estimate) - the exponentiated estimate
- exp(LL) - lower limit of the exponentiated confidence interval
- exp(UL) - upper limit of the exponentiated confidence interval

References


Examples

```r
n1 <- c(204, 201, 932, 130, 77)
n2 <- c(106, 103, 415, 132, 83)
f1 <- c(24, 40, 93, 14, 5)
f2 <- c(12, 9, 28, 3, 1)
x1 <- c(4, 4, 5, 3, 2)
x2 <- c(1, 1, 1, 0, 0)
X <- matrix(cbind(x1, x2), 5, 2)
meta.lm.odds(.05, f1, f2, n1, n2, X)
```

# Should return:
# Estimate  SE   z   p   LL   UL
# b0 1.541895013 0.69815801 2.20851868 0.027 0.1735305 2.91025958
# b1 -0.004417932 0.04840623 -0.09126784 0.927 -0.0992924 0.09045653
# b2 -1.07112269 0.60582695 -1.76803337 0.077 -2.2585213 0.11627674
# exp(Estimate)  exp(LL)  exp(UL)
# b0 4.6734381 1.1894969 18.361564
# b1 0.9955918 0.9054779 1.094674
# b2 0.3426238 0.1045049 1.123307
Description

This function estimates the intercept and slope coefficients in a meta-regression model where the dependent variable is a paired-samples proportion difference. The estimates are OLS estimates with standard errors that accommodate residual heteroscedasticity.

Usage

```r
meta.lm.prop.ps(alpha, f11, f12, f21, f22, X)
```

Arguments

- `alpha`  
  alpha level for 1-alpha confidence
- `f11`  
  vector of frequencies in cell 1,1
- `f12`  
  vector of frequencies in cell 1,2
- `f21`  
  vector of frequencies in cell 2,1
- `f22`  
  vector of frequencies in cell 2,2
- `X`  
  matrix of predictor values

Value

Returns a matrix. The first row is for the intercept with one additional row per predictor. The matrix has the following columns:

- Estimate - OLS estimate
- SE - standard error
- z - z-value
- p - p-value
- LL - lower limit of the confidence interval
- UL - upper limit of the confidence interval

References

Examples

```r
f11 <- c(40, 20, 25, 30)
f12 <- c(3, 2, 2, 1)
f21 <- c(7, 6, 8, 6)
f22 <- c(26, 25, 13, 25)
x1 <- c(1, 1, 4, 6)
x2 <- c(1, 1, 0, 0)
X <- matrix(cbind(x1, x2), 4, 2)
meta.lm.prop.ps(.05, f11, f12, f21, f22, X)
```

# Should return:
# | Estimate | SE  | z     | p     | LL    | UL    |
# |----------|-----|-------|-------|-------|-------|
# | b0       | -0.21113402 | 0.21119823 | -0.9996960 | 0.317 | -0.62507494 | 0.20280690 |
# | b1       | 0.02185567  | 0.03861947  | 0.5659236 | 0.571 | -0.05383711 | 0.09754845 |
# | b2       | 0.12575138  | 0.17655623  | 0.7122455 | 0.476 | -0.22029248 | 0.47179524 |

**meta.lm.prop1**

*Meta-regression analysis for single proportions*

**Description**

This function estimates the intercept and slope coefficients in a meta-regression model where the dependent variable is a proportion from a single group. The estimates are OLS estimates with standard errors that accommodate residual heteroscedasticity.

**Usage**

```r
meta.lm.prop1(alpha, f, n, X)
```

**Arguments**

- `alpha` alpha level for 1-alpha confidence
- `f` vector of sample frequency counts
- `n` vector of sample sizes
- `X` matrix of predictor values

**Value**

Returns a matrix. The first row is for the intercept with one additional row per predictor. The matrix has the following columns:

- Estimate - OLS estimate
- SE - standard error
- z - z-value
- p - p-value
- LL - lower limit of the confidence interval
- UL - upper limit of the confidence interval
Examples

```r
f <- c(38, 26, 24, 15, 45, 38)
n <- c(80, 60, 70, 50, 180, 200)
x1 <- c(10, 15, 18, 22, 24, 30)
X <- matrix(x1, 6, 1)
meta.lm.prop2(.05, f, n, X)
```

# Should return:

<table>
<thead>
<tr>
<th></th>
<th>Estimate</th>
<th>SE</th>
<th>z</th>
<th>p</th>
<th>LL</th>
<th>UL</th>
</tr>
</thead>
<tbody>
<tr>
<td>b0</td>
<td>0.63262816</td>
<td>0.06845707</td>
<td>9.241239</td>
<td>0</td>
<td>0.49845477</td>
<td>0.766801546</td>
</tr>
<tr>
<td>b1</td>
<td>-0.01510565</td>
<td>0.00290210</td>
<td>-5.205076</td>
<td>0</td>
<td>-0.02079367</td>
<td>-0.009417641</td>
</tr>
</tbody>
</table>

---

**meta.lm.prop2**

Meta-regression analysis for 2-group proportion differences

**Description**

This function estimates the intercept and slope coefficients in a meta-regression model where the dependent variable is a 2-group proportion difference. The estimates are OLS estimates with standard errors that accommodate residual heteroscedasticity.

**Usage**

```r
meta.lm.prop2(alpha, f1, f2, n1, n2, X)
```

**Arguments**

- `alpha`: alpha level for 1-alpha confidence
- `f1`: vector of group 1 event counts
- `f2`: vector of group 2 event counts
- `n1`: vector of group 1 sample sizes
- `n2`: vector of group 2 sample sizes
- `X`: matrix of predictor values

**Value**

Returns a matrix. The first row is for the intercept with one additional row per predictor. The matrix has the following columns:

- Estimate - OLS estimate
- SE - standard error
- z - z-value
- p - p-value
- LL - lower limit of the confidence interval
- UL - upper limit of the confidence interval
References


Examples

```r
f1 <- c(24, 40, 93, 14, 5)
f2 <- c(12, 9, 28, 3, 1)
n1 <- c(204, 201, 932, 130, 77)
n2 <- c(106, 103, 415, 132, 83)
x1 <- c(4, 4, 5, 3, 26)
x2 <- c(1, 1, 1, 0, 0)
X <- matrix(cbind(x1, x2), 5, 2)
meta.lm.prop2(.05, f1, f2, n1, n2, X)
```

# Should return:

<table>
<thead>
<tr>
<th>Estimate</th>
<th>SE</th>
<th>z</th>
<th>p</th>
<th>LL</th>
<th>UL</th>
</tr>
</thead>
<tbody>
<tr>
<td>b0</td>
<td>0.089756283</td>
<td>0.034538077</td>
<td>2.5987632</td>
<td>0.009</td>
<td>0.02206290</td>
</tr>
<tr>
<td>b1</td>
<td>-0.001447968</td>
<td>0.001893097</td>
<td>-0.7648672</td>
<td>0.444</td>
<td>-0.00515837</td>
</tr>
<tr>
<td>b2</td>
<td>-0.034670988</td>
<td>0.034125708</td>
<td>-1.0159786</td>
<td>0.310</td>
<td>-0.10156515</td>
</tr>
</tbody>
</table>

---

**meta.lm.propratio2**  
*Meta-regression analysis for proportion ratios*

Description

This function estimates the intercept and slope coefficients in a meta-regression model where the dependent variable is a log proportion ratio. The estimates are OLS estimates with standard errors that accommodate residual heteroscedasticity. The exponentiated slope estimate for a predictor variable describes a multiplicative change in the proportion ratio associated with a 1-unit increase in that predictor variable.

Usage

```r
meta.lm.propratio2(alpha, f1, f2, n1, n2, X)
```

Arguments

- **alpha**: alpha level for 1-alpha confidence
- **f1**: vector of group 1 event counts
- **f2**: vector of group 2 event counts
- **n1**: vector of group 1 sample sizes
- **n2**: vector of group 2 sample sizes
- **X**: matrix of predictor values
Value

Returns a matrix. The first row is for the intercept with one additional row per predictor. The matrix has the following columns:

- **Estimate** - OLS estimate
- **SE** - standard error
- **z** - z-value
- **p** - p-value
- **LL** - lower limit of the confidence interval
- **UL** - upper limit of the confidence interval
- **exp(Estimate)** - the exponentiated estimate
- **exp(LL)** - lower limit of the exponentiated confidence interval
- **exp(UL)** - upper limit of the exponentiated confidence interval

References


Examples

```r
n1 <- c(204, 201, 932, 130, 77)
n2 <- c(106, 103, 415, 132, 83)
f1 <- c(24, 40, 93, 14, 5)
f2 <- c(12, 9, 28, 3, 1)
x1 <- c(4, 4, 5, 3, 26)
x2 <- c(1, 1, 1, 0, 0)
X <- matrix(cbind(x1, x2), 5, 2)
meta.lm.propratio2(.05, f1, f2, n1, n2, X)
```

# Should return:

```
Estimate    SE   z     p   LL    UL
 b0  1.4924887636  0.69172794  2.15762393  0.031  0.13672691  2.84825062
 b1  0.0005759509  0.04999884  0.01151928  0.991 -0.09741998  0.09857188
 b2 -1.0837844594  0.59448206 -1.82307345  0.068 -2.24894789  0.08137897
```

```r
# exp(Estimate)  exp(LL)  exp(UL)
 b0  4.4481522  1.1465150  17.257565
 b1  1.0005761  0.9071749  1.103594
 b2  0.3383128  0.1055102  1.084782
```
**meta.lm.semipart**  
*Meta-regression analysis for semipartial correlations*

**Description**

This function estimates the intercept and slope coefficients in a meta-regression model where the dependent variable is a Fisher-transformed semipartial correlation. The estimates are OLS estimates with standard errors that accommodate residual heteroscedasticity. The correlations are Fisher-transformed and hence the parameter estimates do not have a simple interpretation. However, the hypothesis test results can be used to decide if a population slope is either positive or negative.

**Usage**

```r
meta.lm.semipart(alpha, n, cor, r2, X)
```

**Arguments**

- `alpha`: alpha level for 1-alpha confidence
- `n`: vector of sample sizes
- `cor`: vector of sample semipartial correlations
- `r2`: vector of squared multiple correlations for full model
- `X`: matrix of predictor values

**Value**

Returns a matrix. The first row is for the intercept with one additional row per predictor. The matrix has the following columns:

- Estimate - OLS estimate
- SE - standard error
- z - z-value
- p - p-value
- LL - lower limit of the confidence interval
- UL - upper limit of the confidence interval

**Examples**

```r
n <- c(128, 97, 210, 217)
cor <- c(.35, .41, .44, .39)
r2 <- c(.29, .33, .36, .39)
x1 <- c(18, 25, 23, 19)
X <- matrix(x1, 4, 1)
meta.lm.semipart(.05, n, cor, r2, X)
```

# Should return:
Description

This function estimates the intercept and slope coefficients in a meta-regression model where the dependent variable is a Fisher-transformed Spearman correlation. The estimates are OLS estimates with standard errors that accommodate residual heteroscedasticity. The correlations are Fisher-transformed and hence the parameter estimates do not have a simple interpretation. However, the hypothesis test results can be used to decide if a population slope is either positive or negative.

Usage

```
meta.lm.spear(alpha, n, cor, X)
```

Arguments

- `alpha` alpha level for 1-alpha confidence
- `n` vector of sample sizes
- `cor` vector of Spearman correlations
- `X` matrix of predictor values

Value

Returns a matrix. The first row is for the intercept with one additional row per predictor. The matrix has the following columns:

- Estimate - OLS estimate
- SE - standard error
- z - z-value
- p - p-value
- LL - lower limit of the confidence interval
- UL - upper limit of the confidence interval
Examples

```
n <- c(150, 200, 300, 200, 350)
cor <- c(.14, .29, .16, .21, .23)
x1 <- c(18, 25, 23, 19, 24)
X <- matrix(x1, 5, 1)
meta.lm.spear(.05, n, cor, X)
```

# Should return:

<table>
<thead>
<tr>
<th></th>
<th>Estimate</th>
<th>SE</th>
<th>z</th>
<th>p</th>
<th>LL</th>
<th>UL</th>
</tr>
</thead>
<tbody>
<tr>
<td>b0</td>
<td>-0.08920088</td>
<td>0.26686388</td>
<td>-0.3342561</td>
<td>0.738</td>
<td>-0.612244475</td>
<td>0.43384271</td>
</tr>
<tr>
<td>b1</td>
<td>0.01370866</td>
<td>0.01190212</td>
<td>1.1517825</td>
<td>0.249</td>
<td>-0.009619077</td>
<td>0.03703639</td>
</tr>
</tbody>
</table>

**meta.lm.stdmean.ps**  
*Meta-regression analysis for paired-samples standardized mean differences*

### Description

This function estimates the intercept and slope coefficients in a meta-regression model where the dependent variable is a paired-samples standardized mean difference. The estimates are OLS estimates with standard errors that accommodate residual heteroscedasticity.

### Usage

```
meta.lm.stdmean.ps(alpha, m1, m2, sd1, sd2, cor, n, X, stdzr)
```

### Arguments

- **alpha**: alpha level for 1-alpha confidence
- **m1**: vector of sample means for group 1
- **m2**: vector of sample means for group 2
- **sd1**: vector of sample SDs for group 1
- **sd2**: vector of sample SDs for group 2
- **cor**: vector of correlations
- **n**: vector of sample sizes
- **X**: matrix of predictor values
- **stdzr**:  
  - set to 0 for square root average variance standardizer
  - set to 1 for group 1 SD standardizer
  - set to 2 for group 2 SD standardizer
meta.lm.stdmean2

**Value**

Returns a matrix. The first row is for the intercept with one additional row per predictor. The matrix has the following columns:

- Estimate - OLS estimate
- SE - standard error
- t - t-value
- p - p-value
- LL - lower limit of the confidence interval
- UL - upper limit of the confidence interval

**References**


**Examples**

```r
n <- c(65, 30, 29, 45, 50)
cor <- c(.87, .92, .85, .90, .88)
m1 <- c(20.1, 20.5, 19.3, 21.5, 19.4)
m2 <- c(10.4, 10.2, 8.5, 10.3, 7.8)
sd1 <- c(9.3, 9.9, 10.1, 10.5, 9.8)
sd2 <- c(7.8, 8.0, 8.4, 8.1, 8.7)
x1 <- c(2, 3, 3, 4, 4)
X <- matrix(x1, 5, 1)
meta.lm.stdmean.ps(.05, m1, m2, sd1, sd2, cor, n, X, 0)
```

# Should return:

<table>
<thead>
<tr>
<th>Estimate</th>
<th>SE</th>
<th>z</th>
<th>p</th>
<th>LL</th>
<th>UL</th>
</tr>
</thead>
<tbody>
<tr>
<td>b0</td>
<td>1.01740253</td>
<td>0.25361725</td>
<td>4.0115667</td>
<td>0.000</td>
<td>0.5203218</td>
</tr>
<tr>
<td>b1</td>
<td>0.04977943</td>
<td>0.07755455</td>
<td>0.6418635</td>
<td>0.521</td>
<td>-0.1022247</td>
</tr>
</tbody>
</table>

**meta.lm.stdmean2**

*Meta-regression analysis for 2-group standardized mean differences*

**Description**

This function estimates the intercept and slope coefficients in a meta-regression model where the dependent variable is a 2-group standardized mean difference. The estimates are OLS estimates with standard errors that accommodate residual heteroscedasticity.
Usage

meta.lm.stdmean2(alpha, m1, m2, sd1, sd2, n1, n2, X, stdzr)

Arguments

alpha   alpha level for 1-alpha confidence
m1      vector of sample means for group 1
m2      vector of sample means for group 2
sd1     vector of sample SDs for group 1
sd2     vector of sample SDs for group 2
n1      vector of group 1 sample sizes
n2      vector of group 2 sample sizes
X       matrix of predictor values
stdzr   • set to 0 for square root average variance standardizer
         • set to 1 for group 1 SD standardizer
         • set to 2 for group 2 SD standardizer
         • set to 3 for square root weighted variance standardizer

Value

Returns a matrix. The first row is for the intercept with one additional row per predictor. The matrix has the following columns:

- Estimate - OLS estimate
- SE - standard error
- z - z-value
- p - p-value
- LL - lower limit of the confidence interval
- UL - upper limit of the confidence interval

References


Examples

n1 <- c(65, 30, 29, 45, 50)
n2 <- c(67, 32, 31, 20, 52)
m1 <- c(31.1, 32.3, 31.9, 29.7, 33.0)
m2 <- c(34.1, 33.2, 30.6, 28.7, 26.5)
sd1 <- c(7.1, 8.1, 7.8, 6.8, 7.6)
sd2 <- c(7.8, 7.3, 7.5, 7.2, 6.8)
x1 <- c(4, 6, 7, 7, 8)
\[
X \leftarrow \text{matrix}(x1, 5, 1)
\]
\[
\text{meta.lm.stdmean2(.05, m1, m2, sd1, sd2, n1, n2, X, 0)}
\]

# Should return:

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Estimate</td>
<td>SE</td>
<td>z</td>
<td>p</td>
<td>LL</td>
</tr>
<tr>
<td>b0</td>
<td>-1.6988257</td>
<td>0.4108035</td>
<td>-4.135373</td>
<td>0</td>
</tr>
<tr>
<td>b1</td>
<td>0.2871641</td>
<td>0.0649815</td>
<td>4.419167</td>
<td>0</td>
</tr>
</tbody>
</table>

---

**meta.sub.cor**

*Confidence interval for a difference in average Pearson or partial correlations for two sets of studies*

**Description**

Computes the estimate, standard error, and confidence interval for a difference in average Pearson or partial correlations for two mutually exclusive sets of studies. Each set can have one or more studies. All of the correlations must be either Pearson correlations or partial correlations.

**Usage**

`meta.sub.cor(alpha, n, cor, q, group)`

**Arguments**

- `alpha` - alpha level for 1-alpha confidence
- `n` - vector of sample sizes
- `cor` - vector of sample Pearson correlations
- `q` - number of control variables (set to 0 for Pearson)
- `group` - vector of group indicators:
  - 1 for set A
  - 2 for set B
  - 0 to ignore

**Value**

Returns a matrix with three rows:

- Row 1 - Estimate for Set A
- Row 2 - Estimate for Set B
- Row 3 - Estimate for difference, Set A - Set B

The columns are:

- Estimate - estimate of average correlation or difference
- SE - standard error
- LL - lower limit of the confidence interval
- UL - upper limit of the confidence interval
References


Examples

n <- c(55, 190, 65, 35)
cor <- c(.40, .65, .60, .45)
group <- c(1, 1, 2, 0)
meta.sub.cronbach(.05, n, cor, 0, group)

# Should return:
# Estimate SE LL UL
# Set A: 0.525 0.06195298 0.3932082 0.6356531
# Set B: 0.600 0.08128008 0.4171458 0.7361686
# Set A - Set B: -0.075 0.10219894 -0.2645019 0.1387283

meta.sub.cronbach

Confidence interval for a difference in average Cronbach reliabilities for two sets of studies

Description

Computes the estimate, standard error, and confidence interval for a difference in average Cronbach reliability coefficients for two mutually exclusive sets of studies. Each set can have one or more studies. The number of measurements used to compute the sample reliability coefficient is assumed to be the same for all studies.

Usage

meta.sub.cronbach(alpha, n, rel, q, group)

Arguments

alpha alpha level for 1-alpha confidence
n vector of sample sizes
rel vector of sample Cronbach reliabilities
q number of measurements (e.g., items)
group vector of group indicators:
  • 1 for set A
  • 2 for set B
  • 0 to ignore
Value

Returns a matrix with three rows:

- Row 1 - Estimate for Set A
- Row 2 - Estimate for Set B
- Row 3 - Estimate for difference, Set A - Set B

The columns are:

- Estimate - estimate of average correlation or difference
- SE - standard error
- LL - lower limit of the confidence interval
- UL - upper limit of the confidence interval

References


Examples

n <- c(120, 170, 150, 135)
rel <- c(.89, .87, .73, .71)
group <- c(1, 1, 2, 2)
q <- 10
meta.sub.cronbach(.05, n, rel, q, group)

# Should return:
# Estimate SE LL UL
# Set A: 0.88 0.01068845 0.8581268 0.8999386
# Set B: 0.72 0.02515130 0.6684484 0.7668524
# Set A - Set B: 0.16 0.02732821 0.1082933 0.2152731

Description

Computes the estimate, standard error, and confidence interval for a difference in average point-biserial correlations for two mutually exclusive sets of studies. Each set can have one or more studies. Two types of point-biserial correlations can be analyzed. One type uses an unweighted variance and is appropriate for 2-group experimental designs. The other type uses a weighted variance and is appropriate for 2-group nonexperimental designs with simple random sampling. Equal variances within or across studies is not assumed.
**Usage**

```r
meta.sub.pbcor(alpha, m1, m2, sd1, sd2, n1, n2, type, group)
```

**Arguments**

- **alpha**: alpha level for 1-alpha confidence
- **m1**: vector of sample means for group 1
- **m2**: vector of sample means for group 2
- **sd1**: vector of sample SDs for group 1
- **sd2**: vector of sample SDs for group 2
- **n1**: vector of group 1 sample sizes
- **n2**: vector of group 2 sample sizes
- **type**: • set to 1 for weighted variance
  • set to 2 for unweighted variance
- **group**: vector of group indicators:
  • 1 for set A
  • 2 for set B
  • 0 to ignore

**Value**

Returns a matrix with three rows:

- Row 1 - Estimate for Set A
- Row 2 - Estimate for Set B
- Row 3 - Estimate for difference, Set A - Set B

The columns are:

- Estimate - estimate of average correlation or difference
- SE - standard error
- LL - lower limit of the confidence interval
- UL - upper limit of the confidence interval

**References**

Examples

```r
m1 <- c(45.1, 39.2, 36.3, 34.5)
 m2 <- c(30.0, 35.1, 35.3, 36.2)
 sd1 <- c(10.7, 10.5, 9.4, 11.5)
 sd2 <- c(12.3, 12.0, 10.4, 9.6)
 n1 <- c(40, 20, 50, 25)
 n2 <- c(40, 20, 48, 26)
 group <- c(1, 1, 2, 2)
 meta.sub.pbcor(.05,  m1, m2, sd1, sd2, n1, n2, 2, group)

# Should return:
   Estimate     SE    LL     UL
# Set A:  0.36338772 0.08552728 0.1854777 0.5182304
# Set B:  -0.01480511 0.08741322 -0.1840491 0.1552914
# Set A - Set B:  0.37819284 0.12229467 0.1320530 0.6075828
```

---

**meta.sub.semipart**  
*Confidence interval for a difference in average semipartial correlations for two sets of studies*

**Description**

Computes the estimate, standard error, and confidence interval for a difference in average semipartial correlations for two sets of mutually exclusive studies. Each set can have one or more studies.

**Usage**

```r
meta.sub.semipart(alpha, n, cor, r2, group)
```

**Arguments**

- `alpha`  
  alpha level for 1-alpha confidence
- `n`  
  vector of sample sizes
- `cor`  
  vector of sample semi-partial correlations
- `r2`  
  vector of squared multiple correlations for a model that includes the IV and all control variables
- `group`  
  vector of group indicators:
  - 1 for set A
  - 2 for set B
  - 0 to ignore
Value

Returns a matrix with three rows:

- Row 1 - Estimate for Set A
- Row 2 - Estimate for Set B
- Row 3 - Estimate for difference, Set A - Set B

The columns are:

- Estimate - estimate of average correlation or difference
- SE - standard error
- LL - lower limit of the confidence interval
- UL - upper limit of the confidence interval

Examples

```r
n <- c(55, 190, 65, 35)
cor <- c(.40, .65, .60, .45)
r2 <- c(.25, .41, .43, .39)
group <- c(1, 1, 2, 0)
meta.sub.semipart(.05, n, cor, r2, group)
```

# Should return:

<table>
<thead>
<tr>
<th></th>
<th>Estimate</th>
<th>SE</th>
<th>LL</th>
<th>UL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Set A:</td>
<td>0.525</td>
<td>0.05955276</td>
<td>0.3986844</td>
<td>0.6317669</td>
</tr>
<tr>
<td>Set B:</td>
<td>0.600</td>
<td>0.07931155</td>
<td>0.4221127</td>
<td>0.7333949</td>
</tr>
<tr>
<td>Set A - Set B:</td>
<td>-0.075</td>
<td>0.09918091</td>
<td>-0.2587113</td>
<td>0.1324682</td>
</tr>
</tbody>
</table>

Description

Computes the estimate, standard error, and confidence interval for a difference in average Spearman correlations for two mutually exclusive sets of studies. Each set can have one or more studies.

Usage

```r
meta.sub.spear(alpha, n, cor, group)
```
Arguments

alpha   alpha level for 1-alpha confidence
n       vector of sample sizes
cor     vector of sample Spearman correlations
group   vector of group indicators:
        • 1 for set A
        • 2 for set B
        • 0 to ignore

Value

Returns a matrix with three rows:

• Row 1 - Estimate for Set A
• Row 2 - Estimate for Set B
• Row 3 - Estimate for difference, Set A - Set B

The columns are:

• Estimate - estimate of average correlation or difference
• SE - standard error
• LL - lower limit of the confidence interval
• UL - upper limit of the confidence interval

References


Examples

n <- c(55, 190, 65, 35)
cor <- c(.40, .65, .60, .45)
group <- c(1, 1, 2, 0)
meta.sub.spear(.05, n, cor, group)

# Should return:
# Estimate   SE   LL   UL
# Set A:     0.525 0.06483629 0.3865928 0.6402793
# Set B:     0.600 0.08829277 0.3992493 0.7458512
# Set A - Set B: -0.075 0.10954158 -0.2760700 0.1564955
replicate.cor

Description

This function can be used to compare and combine Pearson or partial correlations from the original study and the follow-up study. The confidence level for the difference is 1 – 2alpha.

Usage

replicate.cor(alpha, cor1, n1, cor2, n2, s)

Arguments

alpha  
alpha level for 1-alpha confidence

cor1  
sample Pearson correlation between y and x in original study

n1  
sample size in original study

cor2  
sample Pearson correlation between y and x in follow-up study

n2  
sample size in follow-up study

s  
number of control variables in each study (0 for Pearson)

Value

A 4-row matrix. The rows are:

• Row 1 summarizes the Original study
• Row 2 summarizes the Follow-up study
• Row 3 estimates the difference between studies
• Row 4 estimates the average effect size between the two studies

The columns are:

• Estimate - Pearson or partial correlation estimate
• SE - standard error
• z - z-value
• p - p-value
• LL - lower limit of the confidence interval
• UL - upper limit of the confidence interval

References

Examples

```r
replicate.cor(.05, .598, 80, .324, 200, 0)
```

# Should return:

```
# Estimate    SE     z       p      LL       UL
# Original: 0.598 0.11396058 6.589418 4.708045e-09 0.4355043 0.7227538
# Follow-up: 0.324 0.07124705 4.819037 2.865955e-06 0.1939787 0.4428347
# Original - Follow-up: 0.274 0.09708614 2.633335 8.455096e-03 0.1065496 0.4265016
# Average: 0.461 0.04854307 7.634998 2.264855e-14 0.3725367 0.5411607
```

---

```
replicate.gen(alpha, est1, se1, est2, se2)
```

**Description**

This function can be used to compare and combine any effect size (e.g., odds ratio, proportion ratio, proportion difference, slope coefficient, etc.) using the effect size estimate and its standard error from the original study and the follow-up study. The same results can be obtained using the `meta.lc.gen` function with appropriate contrast coefficients. The confidence level for the difference is $1 - 2\alpha$.

**Usage**

```r
replicate.gen(alpha, est1, se1, est2, se2)
```

**Arguments**

- `alpha` - alpha level for 1-alpha confidence
- `est1` - sample effect size in original study
- `se1` - effect size standard error in original study
- `est2` - sample effect size in follow-up study
- `se2` - effect size standard error in follow-up study

**Value**

A 4-row matrix. The rows are:

- Row 1 summarizes the Original study
- Row 2 summarizes the Follow-up study
- Row 3 estimates the difference between studies
- Row 4 estimates the average effect size between the two studies

Columns are:

- Estimate - effect size estimate
replicate.mean.ps

- SE - standard error
- z - z-value
- p - p-value
- LL - lower limit of the confidence interval
- UL - upper limit of the confidence interval

References

Examples
replicate.gen(.05, .782, .210, .650, .154)

# Should return:
# Estimate SE z p LL UL
# Original: 0.782 0.2100000 3.7238095 1.962390e-04 0.3704076 1.1935924
# Follow-up: 0.650 0.1540000 4.2207792 2.434593e-05 0.3481655 0.9518345
# Original - Follow-up: 0.132 0.2604151 0.5068831 6.122368e-01 -0.2963446 0.5603446
# Average: 0.716 0.1302075 5.4989141 3.821373e-08 0.4607979 0.9712021

replicate.mean.ps  Compares paired-samples mean differences in original and follow-up studies

Description
This function computes confidence intervals for a paired-samples mean difference from an original study and a follow-up study. Confidence intervals for the difference and average effect size also are computed. A Satterthwaite adjustment to the degrees of freedom is used to improve the accuracy of the confidence intervals for the difference and average. The same results can be obtained using the meta.lc.mean.ps function with appropriate contrast coefficients. The confidence level for the difference is 1 – 2alpha.

Usage
replicate.mean.ps(
  alpha,
  m11,
  m12,
  sd11,
  sd12,
  cor1,
  n1,
  ...)
m21,  
m22,  
sd21,  
sd22,  
cor2,  
n2  
)

Arguments

alpha      alpha level for 1-alpha confidence
m11        sample mean for group 1 in original study
m12        sample mean for group 2 in original study
sd11       sample SD for group 1 in original study
sd12       sample SD for group 2 in original study
cor1       sample correlation of paired observations in original study
n1         sample size in original study
m21        sample mean for group 1 in follow-up study
m22        sample mean for group 2 in follow-up study
sd21       sample SD for group 1 in follow-up study
sd22       sample SD for group 2 in follow-up study
cor2       sample correlation of paired observations in follow-up study
n2         sample size in follow-up study

Value

A 4-row matrix. The rows are:

- Row 1 summarizes the Original study
- Row 2 summarizes the Follow-up study
- Row 3 estimates the difference between studies
- Row 4 estimates the average effect size between the two studies

The columns are:

- Estimate - effect size estimate
- SE - standard error
- df - degrees of freedom
- t - t-value
- p - p-value
- LL - lower limit of the confidence interval
- UL - upper limit of the confidence interval
References


Examples

```r
replicate.mean.ps(.05, 86.22, 70.93, 14.89, 12.32, .765, 20, 
84.81, 77.24, 15.68, 16.95, .702, 75)
```

# Should return:

<table>
<thead>
<tr>
<th>Estimate</th>
<th>SE</th>
<th>t</th>
<th>p</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original:</td>
<td>15.29</td>
<td>2.154344</td>
<td>7.097288</td>
</tr>
<tr>
<td>Follow-up:</td>
<td>7.57</td>
<td>1.460664</td>
<td>5.182575</td>
</tr>
<tr>
<td>Original - Follow-up:</td>
<td>7.72</td>
<td>2.602832</td>
<td>2.966000</td>
</tr>
<tr>
<td>Average:</td>
<td>11.43</td>
<td>1.301416</td>
<td>8.782740</td>
</tr>
<tr>
<td>LL</td>
<td>UL</td>
<td>df</td>
<td></td>
</tr>
<tr>
<td>Original:</td>
<td>10.780906</td>
<td>19.79909</td>
<td>19.00000</td>
</tr>
<tr>
<td>Follow-up:</td>
<td>4.659564</td>
<td>10.48044</td>
<td>74.00000</td>
</tr>
<tr>
<td>Original - Follow-up:</td>
<td>3.332885</td>
<td>12.10712</td>
<td>38.40002</td>
</tr>
<tr>
<td>Average:</td>
<td>8.796322</td>
<td>14.06368</td>
<td>38.40002</td>
</tr>
</tbody>
</table>

replicate.mean2

*Compares 2-group mean differences in original and follow-up studies*

Description

This function computes confidence intervals for a 2-group mean difference from an original study and a follow-up study. Confidence intervals for the difference and average effect size also are computed. A Satterthwaite adjustment to the degrees of freedom is used to improve the accuracy of the confidence intervals. The same results can be obtained using the *meta.lc.mean2* function with appropriate contrast coefficients. The confidence level for the difference is $1 - 2\alpha$.

Usage

```r
replicate.mean2(
  alpha, 
  m11,  
  m12, 
  sd11, 
  sd12, 
  n11, 
  n12, 
  m21, 
  m22, 
  sd21,
)```
replicate.mean2

sd22, n21, n22
)

Arguments

alpha alpha level for 1-alpha confidence
m11 sample mean for group 1 in original study
m12 sample mean for group 2 in original study
sd11 sample SD for group 1 in original study
sd12 sample SD for group 2 in original study
n11 sample size for group 1 in original study
n12 sample size for group 2 in original study
m21 sample mean for group 1 in follow-up study
m22 sample mean for group 2 in follow-up study
sd21 sample SD for group 1 in follow-up study
sd22 sample SD for group 2 in follow-up study
n21 sample size for group 1 in follow-up study
n22 sample size for group 2 in follow-up study

Value

A 4-row matrix. The rows are:

• Row 1 summarizes the Original study
• Row 2 summarizes the Follow-up study
• Row 3 estimates the difference between studies
• Row 4 estimates the average effect size between the two studies

The columns are:

• Estimate - effect size estimate
• SE - standard error
• t - t-value
• p - p-value
• LL - lower limit of the confidence interval
• UL - upper limit of the confidence interval
• df - degrees of freedom

References

replicate.plot

Examples

replicate.mean2(.05, 21.9, 16.1, 3.82, 3.21, 40, 40,
                   25.2, 19.1, 3.98, 3.79, 75, 75)

# Should return:
# Estimate       SE       t       p
# Original:     5.80  0.7889312  7.3517180  1.927969e-10
# Follow-up:    6.10  0.6346075  9.6122408  0.000000e+00
# Original - Follow-up:  -0.30  1.0124916  -0.2962988  7.673654e-01
# Average:      5.95  0.5062458  11.7531843  0.000000e+00
#               LL       UL       df
# Original:     4.228624  7.371376  75.75255
# Follow-up:    4.845913  7.354087  147.64728
# Original - Follow-up:  -1.974571  1.374571  169.16137
# Average:      4.950627  6.949373  169.16137

replicate.plot

Plot to compare estimates from an original and follow-up study

Description

Generates a basic plot using ggplot2 to visualize the estimates from and original and follow-up study

Usage

replicate.plot(
  result,
  focus = c("Both", "Difference", "Average"),
  reference_line = NULL,
  diamond_height = 0.2,
  difference_axis_ticks = 5,
  ggtheme = ggplot2::theme_classic()
)

Arguments

result • a result matrix from any of the replicate functions in vcmeta
focus • Optional specification of the focus of the plot; defaults to 'Both'
  • Both - a bit busy; plots each estimate, difference, and average
  • Difference - plot each estimate and difference between them
  • Average - plot each estimate and the average effect size
reference_line Optional x-value for a reference line. Only applies if focus is 'Difference' or 'Both'. Defaults to NULL, in which case a reference line is not drawn.
diamond_height  • Optional height of the diamond representing average effect size. Only applies if focus is 'Average' or 'Both'. Defaults to 0.2

difference_axis_ticks  • Optional requested number of ticks on the difference axis. Only applies if focus is 'Difference' or 'Both'. Defaults to 5.

ggtheme  • optional ggplot2 theme object; defaults to theme_classic()

Value

Returns a ggplot object. If stored, can be further customized via the ggplot API

Examples

# Compare Damisch et al., 2010 to Calin-Jageman & Caldwell 2014
# Damisch et al., 2010, Exp 1, German participants made 10 mini-golf putts
# Half were told they had a 'lucky' golf ball; half were not
# Found a large but uncertain improvement in shots made in the luck condition
# Calin-Jageman & Caldwell, 2014, Exp 1, was a pre-registered replication with
# input from Damisch, though with English-speaking participants
#
# Here we compare the effect sizes, in original units, for the two studies
# Use the replicate.mean2 function because the design is 2-group between-subs

library(ggplot2)
damisch_v_calinjageman_raw <- replicate.mean2(
  alpha = 0.05,
  m11 = 6.42,
  m12 = 4.75,
  sd11 = 1.88,
  sd12 = 2.15,
  n11 = 14,
  n12 = 14,
  m21 = 4.73,
  m22 = 4.62,
  sd21 = 1.958,
  sd22 = 2.12,
  n21 = 66,
  n22 = 58
)

# View the comparison:
damisch_v_calinjageman_raw

# Now plot the comparison, focusing on the difference
replicate.plot(damisch_v_calinjageman_raw, focus = "Difference")

# Plot the comparison, focusing on the average
replicate.plot(damisch_v_calinjageman_raw,
               focus = "Average",
               reference_line = 0,
diamond_height = 0.1

# Kind of busy, but plot the comparison with both difference and average
# In this case, store the plot for manipulation
myplot <- replicate.plot(
  damisch_v_calinjageman_raw,
  focus = "Both",
  reference_line = 0
)

# View the stored plot
myplot

# Change x-labels and study labels
myplot <- myplot + xlab("Difference in Putts Made, Lucky - Control")
myplot <- myplot + scale_y_discrete(
  labels = c(
    "Average",
    "Difference",
    "Calin-Jageman & Caldwell, 2014",
    "Damisch et al., 2010"
  )
)

# View the updated plot
myplot

---

**replicate.stdmean.ps**  
*Compares paired-samples standardized mean differences in original and follow-up studies*

**Description**

This function computes confidence intervals for a paired-samples standardized mean difference from an original study and a follow-up study. Confidence intervals for the difference and average effect size also are computed. The same results can be obtained using the `meta.lc.stdmean.ps` function with appropriate contrast coefficients. The confidence level for the difference is $1 - 2\alpha$.

**Usage**

```r
replicate.stdmean.ps(
  alpha,  
  m11,  
  m12,  
  sd11,  
  sd12,
)```
cor1, n1, m21, m22, sd21, sd22, cor2, n2
)

Arguments

alpha alpha level for 1-alpha confidence
m11 sample mean for group 1 in original study
m12 sample mean for group 2 in original study
sd11 sample SD for group 1 in original study
sd12 sample SD for group 2 in original study
cor1 sample correlation of paired observations in original study
n1 sample size in original study
m21 sample mean for group 1 in follow-up study
m22 sample mean for group 2 in follow-up study
sd21 sample SD for group 1 in follow-up study
sd22 sample SD for group 2 in follow-up study
cor2 sample correlation of paired observations in follow-up study
n2 sample size in follow-up study

Value

A 4-row matrix. The rows are:

- Row 1 summarizes the Original study
- Row 2 summarizes the Follow-up study
- Row 3 estimates the difference between studies
- Row 4 estimates the average effect size between the two studies

The columns are:

- Estimate - effect size estimate
- SE - standard error
- LL - lower limit of the confidence interval
- UL - upper limit of the confidence interval
replicate.stdmean2

References


Examples

replicate.stdmean2(ps(
  alpha = .05,
  m11 = 86.22, m12 = 70.93, sd11 = 14.89, sd12 = 12.32, cor1 = .765, n1 = 20,
  m21 = 84.81, m22 = 77.24, sd21 = 15.68, sd22 = 16.95, cor2 = .702, n2 = 75
)

# Should return:
# Estimate  SE   LL    UL
# Orginal:  1.089030 0.229156 0.669735 1.568009
# Follow-up: 0.460496 0.095905 0.275669 0.651601
# Original - Follow-up: 0.625532 0.248416 0.246626 1.063840
# Average:  0.774763 0.124208 0.531320 1.018205

replicate.stdmean2 Compares 2-group standardized mean differences in original and follow-up studies

Description

This function computes confidence intervals for a 2-group standardized mean difference from an original study and a follow-up study. Confidence intervals for the difference and average effect size also are computed. The same results can be obtained using the meta.lc.stdmean2 function with appropriate contrast coefficients. The confidence level for the difference is 1 – 2alpha.

Usage

replicate.stdmean2(
  alpha,
  m11, m12, sd11, sd12, n11, n12, m21, m22, sd21, sd22, n21,
Arguments

alpha alpha level for 1-alpha confidence
m11 sample mean for group 1 in original study
m12 sample mean for group 2 in original study
sd11 sample SD for group 1 in original study
sd12 sample SD for group 2 in original study
n11 sample size for group 1 in original study
n12 sample size for group 2 in original study
m21 sample mean for group 1 in follow-up study
m22 sample mean for group 2 in follow-up study
sd21 sample SD for group 1 in follow-up study
sd22 sample SD for group 2 in follow-up study
n21 sample size for group 1 in follow-up study
n22 sample size for group 2 in follow-up study

Value

A 4-row matrix. The rows are:

- Row 1 summarizes the Original study
- Row 2 summarizes the Follow-up study
- Row 3 estimates the difference between studies
- Row 4 estimates the average effect size between the two studies

The columns are:

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

References

Examples

replicate.stdmean2(.05, 21.9, 16.1, 3.82, 3.21, 40, 40,
  25.2, 19.1, 3.98, 3.79, 75, 75)

# Should return:
# Estimate    SE     LL     UL
# Original:   1.62803662 0.2594668 1.1353486 2.1524396
# Follow-up:  1.56170447 0.1870576 1.2030461 1.9362986
# Original - Follow-up: 0.07422178 0.3198649 -0.4519092 0.6003527
# Average:    1.59487055 0.1599325 1.2814087 1.9083324

se.cor

Computes the standard error for a Pearson or partial correlation

Description

This function can be used to compute the standard error of a Pearson or partial correlation using the sample correlation, sample size, and number of control variables. The effect size estimate and standard error output from this function can be used as input in the meta.ave.gen, meta.lc.gen, and meta.lm.gen functions in applications where a combination of different types of correlations are used in the meta-analysis.

Usage

se.cor(cor, q, n)

Arguments

cor sample Pearson or partial correlation
q number of control variables (0 for Pearson)
n sample size

Value

Returns a one-row matrix:

- Estimate - estimate of Pearson or partial correlation
- SE - standard error

References

Examples

\[ \text{se.cor}(0.40, 0, 55) \]

# Should return:
# Estimate SE
# Correlation: 0.4 0.116487

se.mean.ps Computes the standard error for a paired-samples mean difference

Description

This function can be used to compute the standard error of a paired-samples mean difference using the two sample means, sample effect size estimate and standard error output from this function can be used as input in the meta.ave.gen, meta.lc.gen, and meta.lm.gen functions in applications where compatible mean differences from a combination of 2-group and paired-samples experiments are used in the meta-analysis.

Usage

\[ \text{se.mean.ps}(m1, m2, sd1, sd2, cor, n) \]

Arguments

- \( m1 \): sample mean for measurement 1
- \( m2 \): sample mean for measurement 2
- \( sd1 \): sample standard deviation for measurement 1
- \( sd2 \): sample standard deviation for measurement 2
- \( cor \): sample correlation for measurements 1 and 2
- \( n \): sample size

Value

Returns a one-row matrix:

- Estimate - estimate of mean difference
- SE - standard error

References

se.mean2

Examples

se.mean.ps(23.9, 25.1, 1.76, 2.01, .78, 25)

# Should return:
# Estimate          SE
# Mean difference: -1.2  0.2544833

Description

This function can be used to compute the standard error of a 2-group mean difference using the two sample means, sample standard deviations, and sample sizes. The effect size estimate and standard error output from this function can be used as input in the meta.ave.gen, meta.lc.gen, and meta.lm.gen functions in applications where compatible mean differences from a combination of 2-group and paired-samples experiments are used in the meta-analysis.

Usage

se.mean2(m1, m2, sd1, sd2, n1, n2)

Arguments

m1 sample mean for group 1
m2 sample mean for group 2
sd1 sample standard deviation for group 1
sd2 sample standard deviation for group 2
n1 group 1 sample size
n2 group 2 sample size

Value

Returns a one-row matrix:

- Estimate - estimate of mean difference
- SE - standard error

References

Examples

se.mean2(21.9, 16.1, 3.82, 3.21, 40, 40)

# Estimate SE
# Mean difference: 5.8 0.7889312

se.meanratio.ps Computes the standard error for a paired-samples log mean ratio

Description

This function can be used to compute the standard error of a paired-samples mean ratio using the two sample means, sample effect size estimate and standard error output from this function can be used as input in the meta.ave.gen, meta.lc.gen, and meta.lm.gen functions in application where compatible mean ratios from a combination of 2-group and paired-samples experiments are used in the meta-analysis.

Usage

se.meanratio.ps(m1, m2, sd1, sd2, cor, n)

Arguments

m1 sample mean for measurement 1
m2 sample mean for measurement 2
sd1 sample standard deviation for measurement 1
sd2 sample standard deviation for measurement 2
cor sample correlation for measurements 1 and 2
n sample size

Value

Returns a one-row matrix:

- Estimate - estimate of log mean ratio
- SE - standard error

References

se.meanratio2

Examples

se.meanratio.ps(21.9, 16.1, 3.82, 3.21, .748, 40)

# Should return:
# Estimate SE
# Log mean ratio: 0.3076674 0.02130161

se.meanratio2  Computes the standard error for a 2-group log mean ratio

Description

This function can be used to compute the standard error of a 2-group mean ratio using the two sample means, sample standard deviations, and sample sizes. The effect size estimate and standard error output from this function can be used as input in the meta.ave.gen, meta.lc.gen, and meta.lm.gen functions in application where compatible mean ratios from a combination of 2-group and paired-samples experiments are used in the meta-analysis.

Usage

se.meanratio2(m1, m2, sd1, sd2, n1, n2)

Arguments

m1  sample mean for group 1
m2  sample mean for group 2
sd1 sample standard deviation for group 1
sd2 sample standard deviation for group 2
n1  group 1 sample size
n2  group 2 sample size

Value

Returns a one-row matrix:

• Estimate - estimate of log mean ratio
• SE - standard error

References

Examples

se.meanratio2(21.9, 16.1, 3.82, 3.21, 40, 40)

# Should return:
# Estimate       SE
# Log mean ratio: 0.3076674 0.041886

se.odds Computes the standard error for a log odds ratio

Description

This function computes a log odds ratio and its standard error using the frequency counts and sample sizes in a 2-group design. These frequency counts and sample sizes can be obtained from a 2x2 contingency table. This function is useful in a meta-analysis of odds ratios where some studies report the sample odds ratio and its standard error and other studies only report the frequency counts (or a 2x2 contingency table. The log odds ratio and standard error output from this function can be used as input in the meta.ave.gen, meta.lc.gen, and meta.lm.gen functions.

Usage

se.odds(f1, n1, f2, n2)

Arguments

f1 number of participants who have the outcome of interest in group 1
n1 group 1 sample size
f2 number of participants who have the outcome of interest in group 2
n2 group 2 sample size

Value

Returns a one-row matrix:

- Estimate - estimate of log odds ratio
- SE - standard error

References

Examples

se.odds(36, 50, 21, 50)

# Should return:
# Estimate       SE
# Log odds ratio: 1.239501 0.4204435

se.pbcor

Computes the standard error for a point-biserial correlation

Description

The function computes a point-biserial correlation and its standard error for two types of point-biserial correlations in 2-group designs using the sample means, sample standard deviations, and samples sizes. One type of point-biserial correlation uses an unweighted average of variances and is appropriate for 2-group experimental designs. The other type of point-biserial correlation uses an weighted average of variances and is appropriate for 2-group nonexperimental designs with simple random sampling. This function is useful in a meta-analysis of compatible point-biserial correlations where some studies used a 2-group experimental design and other studies used a 2-group nonexperimental design. The effect size estimate and standard error output from this function can be used as input in the meta.ave.gen, meta.lc.gen, and meta.lm.gen functions.

Usage

se.pbcor(m1, m2, sd1, sd2, n1, n2, type)

Arguments

m1 sample mean for group 1
m2 sample mean for group 2
sd1 sample standard deviation for group 1
sd2 sample standard deviation for group 2
n1 group 1 sample size
n2 group 2 sample size
type • set to 1 for weighted variance average
       • set to 2 for unweighted variance average

Value

Returns a one-row matrix:

• Estimate - estimate of point-biserial correlation
• SE - standard error
References


Examples

```r
se.pbcor(21.9, 16.1, 3.82, 3.21, 40, 40, 1)
```

# Should return:
# Estimate SE
# Point-biserial correlation: 0.6349786 0.05981325

---

**se.semipartial**

*Computation of the standard error for a semipartial correlation*

Description

This function can be used to compute the standard error of a semipartial correlation using the sample correlation, sample size, and squared multiple correlation for the full model. The effect size estimate and standard error output from this function can be used as input in the `meta.ave.gen`, `meta.lc.gen`, and `meta.lm.gen` functions in applications where a combination of different types of correlations are used in the meta-analysis.

Usage

```r
se.semipartial(cor, r2, n)
```

Arguments

- `cor` - sample semipartial correlation
- `r2` - squared multiple correlation for full model
- `n` - sample size

Value

Returns a one-row matrix:

- Estimate - estimate of semipartial correlation
- SE - standard error
Examples

```r
se.semipartial(.40, .25, 60)
```

# Should return:
# Estimate         SE
# Semipartial correlation:  0.4 0.1063262

---

**se.slope**

*Computes a slope and standard error*

**Description**

This function can be used to compute a slope and its standard error for a simple linear regression model using the sample Pearson correlation and the standard deviations of response and predictor variables. This function is useful in a meta-analysis of slopes of a simple linear regression model where some studies report the Pearson correlation but not the slope.

**Usage**

```r
se.slope(cor, sdy, sdx, n)
```

**Arguments**

- `cor` sample Pearson correlation
- `sdy` sample standard deviation of the response variable
- `sdx` sample standard deviation of the predictor variable
- `n` sample size

**Value**

Returns a one-row matrix:

- `Estimate` - estimate of slope
- `SE` - standard error

**References**

se.spear

Computes the standard error for a Spearman correlation

Description

This function can be used to compute the standard error of a Spearman correlation using the sample correlation and sample size. The standard error from this function can be used as input in the meta.ave.gen, meta.lc.gen, and meta.lm.gen functions in applications where a combination of different types of correlations are used in the meta-analysis.

Usage

se.spear(cor, n)

Arguments

cor   sample Spearman correlation
n     sample size

Value

Returns a one-row matrix:

- Estimate - estimate of Spearman correlation
- SE - standard error

References


Examples

se.spear(.40, 55)

# Should return:
# Estimate     SE
# Spearman correlation: 0.4 0.1210569
Se.stdmean.ps

Computes the standard error for a paired-samples standardized mean difference

Description

The effect size estimate and standard error output from this function can be used as input in the meta.ave.gen, meta.lc.gen, and meta.lm.gen functions in applications where compatible standardized mean differences from a combination of 2-group and paired-samples experiments are used in the meta-analysis.

Usage

se.stdmean.ps(m1, m2, sd1, sd2, cor, n, stdzr)

Arguments

m1  sample mean for measurement 1
m2  sample mean for measurement 2
sd1 sample standard deviation for measurement 1
sd2 sample standard deviation for measurement 2
cor sample correlation for measurements 1 and 2
n  sample size
stdzr • set to 0 for square root average variance standardizer
    • set to 1 for group 1 SD standardizer
    • set to 2 for group 2 SD standardizer
    • set to 3 for square root weighted variance standardizer

Value

Returns a one-row matrix:

• Estimate - estimate of standardized mean difference
• SE - standard error

References

Examples

se.stdmean2(23.9, 25.1, 1.76, 2.01, .78, 25, 0)

# Should return:
# Estimate SE
# Standardized mean difference: -0.6352097 0.1602852

---

se.stdmean2 Computes the standard error for a 2-group standardized mean difference

Description

Use the square root average variance standardizer (stdzr = 0) for 2-group experimental designs. Use the square root weighted variance standardizer (stdzr = 3) for 2-group nonexperimental designs with simple random sampling. The single-group standardizers (stdzr = 1 and stdzr = 2) can be used with either 2-group experimental or nonexperimental designs. The effect size estimate and standard error output from this function can be used as input in the meta.ave.gen, meta.lc.gen, and meta.lm.gen functions in applications where compatible standardized mean differences from a combination of 2-group and paired-samples experiments are used in the meta-analysis.

Usage

se.stdmean2(m1, m2, sd1, sd2, n1, n2, stdzr)

Arguments

- m1: sample mean for group 1
- m2: sample mean for group 2
- sd1: sample standard deviation for group 1
- sd2: sample standard deviation for group 2
- n1: group 1 sample size
- n2: group 2 sample size
- stdzr: set to 0 for square root average variance standardizer, set to 1 for group 1 SD standardizer, set to 2 for group 2 SD standardizer, set to 3 for square root weighted variance standardizer

Value

Returns a one-row matrix:

- Estimate: estimate of standardized mean difference
- SE: standard error
References


Examples

se.stdmean2(21.9, 16.1, 3.82, 3.21, 40, 40, 0)

# Should return:
# Estimate SE
# Standardized mean difference: 1.643894 0.2629049
Index

<table>
<thead>
<tr>
<th>Subject</th>
<th>Page(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ci.fisher</td>
<td>4</td>
</tr>
<tr>
<td>cor.from.t</td>
<td>4</td>
</tr>
<tr>
<td>meta.ave.agree</td>
<td>5</td>
</tr>
<tr>
<td>meta.ave.cor</td>
<td>6</td>
</tr>
<tr>
<td>meta.ave.cronbach</td>
<td>7</td>
</tr>
<tr>
<td>meta.ave.gen, 8, 93–100, 102–104</td>
<td>6</td>
</tr>
<tr>
<td>meta.ave.gen.cc</td>
<td>10</td>
</tr>
<tr>
<td>meta.ave.gen.rc</td>
<td>11</td>
</tr>
<tr>
<td>meta.ave.mean.ps</td>
<td>12</td>
</tr>
<tr>
<td>meta.ave.mean2</td>
<td>14</td>
</tr>
<tr>
<td>meta.ave.meanratio.ps</td>
<td>15</td>
</tr>
<tr>
<td>meta.ave.meanratio2</td>
<td>16</td>
</tr>
<tr>
<td>meta.ave.odds</td>
<td>18</td>
</tr>
<tr>
<td>meta.ave.path</td>
<td>19</td>
</tr>
<tr>
<td>meta.ave.pbcor</td>
<td>20</td>
</tr>
<tr>
<td>meta.ave.plot</td>
<td>22</td>
</tr>
<tr>
<td>meta.ave.prop.ps</td>
<td>23</td>
</tr>
<tr>
<td>meta.ave.prop2</td>
<td>24</td>
</tr>
<tr>
<td>meta.ave.propratio2</td>
<td>26</td>
</tr>
<tr>
<td>meta.ave.semipart</td>
<td>27</td>
</tr>
<tr>
<td>meta.ave.slope</td>
<td>28</td>
</tr>
<tr>
<td>meta.ave.spear</td>
<td>29</td>
</tr>
<tr>
<td>meta.ave.stdmean.ps</td>
<td>30</td>
</tr>
<tr>
<td>meta.ave.stdmean2</td>
<td>32</td>
</tr>
<tr>
<td>meta.lc.agree</td>
<td>33</td>
</tr>
<tr>
<td>meta.lc.gen, 34, 82, 93–100, 102–104</td>
<td>33</td>
</tr>
<tr>
<td>meta.lc.mean.ps</td>
<td>35, 83</td>
</tr>
<tr>
<td>meta.lc.mean1, 36</td>
<td></td>
</tr>
<tr>
<td>meta.lc.mean2, 37, 85</td>
<td></td>
</tr>
<tr>
<td>meta.lc.meanratio.ps</td>
<td>39</td>
</tr>
<tr>
<td>meta.lc.meanratio2</td>
<td>40</td>
</tr>
<tr>
<td>meta.lc.odds</td>
<td>41</td>
</tr>
<tr>
<td>meta.lc.prop.ps</td>
<td>43</td>
</tr>
<tr>
<td>meta.lc.prop1, 44</td>
<td></td>
</tr>
<tr>
<td>meta.lc.prop2, 45</td>
<td></td>
</tr>
<tr>
<td>meta.lc.propratio2, 46</td>
<td></td>
</tr>
<tr>
<td>meta.lc.stdmean.ps</td>
<td>47, 89</td>
</tr>
<tr>
<td>meta.lc.stdmean2, 49, 91</td>
<td></td>
</tr>
<tr>
<td>meta.lm.agree, 50</td>
<td></td>
</tr>
<tr>
<td>meta.lm.cor</td>
<td>51</td>
</tr>
<tr>
<td>meta.lm.cor.gen, 52</td>
<td></td>
</tr>
<tr>
<td>meta.lm.cronbach</td>
<td>53</td>
</tr>
<tr>
<td>meta.lm.gen, 55, 93–100, 102–104</td>
<td>55</td>
</tr>
<tr>
<td>meta.lm.mean.ps</td>
<td>56</td>
</tr>
<tr>
<td>meta.lm.mean1, 57</td>
<td></td>
</tr>
<tr>
<td>meta.lm.mean2, 58</td>
<td></td>
</tr>
<tr>
<td>meta.lm.meanratio.ps, 59</td>
<td></td>
</tr>
<tr>
<td>meta.lm.meanratio2, 61</td>
<td></td>
</tr>
<tr>
<td>meta.lm.odds</td>
<td>62</td>
</tr>
<tr>
<td>meta.lm.prop.ps</td>
<td>64</td>
</tr>
<tr>
<td>meta.lm.prop1, 65</td>
<td></td>
</tr>
<tr>
<td>meta.lm.prop2, 66</td>
<td></td>
</tr>
<tr>
<td>meta.lm.propratio2, 67</td>
<td></td>
</tr>
<tr>
<td>meta.lm.semipart</td>
<td>69</td>
</tr>
<tr>
<td>meta.lm.spear</td>
<td>70</td>
</tr>
<tr>
<td>meta.lm.stdmean.ps</td>
<td>71</td>
</tr>
<tr>
<td>meta.lm.stdmean2</td>
<td>72</td>
</tr>
<tr>
<td>meta.sub.cor</td>
<td>74</td>
</tr>
<tr>
<td>meta.sub.cronbach</td>
<td>75</td>
</tr>
<tr>
<td>meta.sub.pbcor</td>
<td>76</td>
</tr>
<tr>
<td>meta.sub.semipart</td>
<td>78</td>
</tr>
<tr>
<td>meta.sub.spear</td>
<td>79</td>
</tr>
<tr>
<td>replicate.cor</td>
<td>81</td>
</tr>
<tr>
<td>replicate.gen</td>
<td>82</td>
</tr>
<tr>
<td>replicate.mean.ps</td>
<td>83</td>
</tr>
<tr>
<td>replicate.mean2, 85</td>
<td></td>
</tr>
<tr>
<td>replicate.plot</td>
<td>87</td>
</tr>
<tr>
<td>replicate.stdmean.ps, 89</td>
<td></td>
</tr>
<tr>
<td>replicate.stdmean2, 91</td>
<td></td>
</tr>
<tr>
<td>se.cor, 93</td>
<td></td>
</tr>
<tr>
<td>se.mean.ps, 94</td>
<td></td>
</tr>
<tr>
<td>se.mean2, 95</td>
<td></td>
</tr>
<tr>
<td>se.meanratio.ps, 96</td>
<td></td>
</tr>
<tr>
<td>se.meanratio2, 97</td>
<td></td>
</tr>
<tr>
<td>se.odds, 98</td>
<td></td>
</tr>
<tr>
<td>se.pbcor, 99</td>
<td></td>
</tr>
<tr>
<td>se.semipartial, 100</td>
<td></td>
</tr>
</tbody>
</table>
INDEX

se.slope, 101
se.spear, 102
se.stdmean.ps, 103
se.stdmean2, 104