Package ‘metavcov’

June 24, 2023

Type Package

Title Computing Variances and Covariances, Visualization and Missing Data Solution for Multivariate Meta-Analysis

Version 2.1.5

Date 2023-06-22

Author Min Lu <m.lu6@umiami.edu>

Maintainer Min Lu <m.lu6@umiami.edu>

BugReports https://github.com/luminwin/metavcov/issues/new

Depends R (>= 3.5.0),

Imports ggplot2

Suggests mixmeta, metaSEM, mvmeta, mice

Description Collection of functions to compute within-study covariances for different effect sizes, data visualization, and single and multiple imputations for missing data. Effect sizes include correlation (r), mean difference (MD), standardized mean difference (SMD), log odds ratio (logOR), log risk ratio (logRR), and risk difference (RD).

License GPL (>= 2)

LazyData TRUE

URL https://github.com/luminwin/metavcov

NeedsCompilation no

Repository CRAN

Date/Publication 2023-06-24 04:00:02 UTC

R topics documented:

metavcov-package *Computing Variances and Covariances, Visualization and Missing Data Solution for Multivariate Meta Analysis*

Description

The package metavcov computes variances and covariances for effect sizes as preparations for multivariate meta-analysis. Effect sizes include correlation (r), mean difference (MD), standardized mean difference (SMD), log odds ratio (logOR), log risk ratio (logRR), and risk difference (RD). Functions for plotting confidence intervals and multiple imputation for missing data are offered. It can fit a fixed-effect model for demonstration purposes; users are highly encouraged to use packages mixmeta and metaSEM for random-effect models.

Author(s)

Min Lu (Maintainer,<m.lu6@umiami.edu>)

References

Lu, M. (2023). Computing within-study covariances, data visualization, and missing data solutions for multivariate meta-analysis with metavcov. *Frontiers in Psychology*, 14:1185012.

Examples

```
################################################
# Effect size: correlation coefficients
################################################
data(Craft2003)
```

```
# extract correlation from the dataset (craft)
corflat <- subset(Craft2003, select=C1:C6)
# transform correlations to z and compute variance-covariance matrix.
computvcov <- r.vcov(n = Craft2003$N, corflat = corflat, method = "average")
# name transformed z scores as y
y <- computvcov$ef
# name variance-covariance matrix of trnasformed z scores as S
S <- computvcov$matrix.vcov
S[1, ]
## fixed-effect model
MMA_FE <- summary(metafixed(y = y, Slist = computvcov$list.vcov))
MMA_FE
# Restricted maximum likelihood (REML) estimator from the mixmeta package
#library(mixmeta)
#mvmeta_RE <- summary(mixmeta(cbind(C1, C2, C3, C4, C5, C6)~1,
#                            S = S, data = y, method = "reml"))
#mvmeta_RE

# maximum likelihood estimators from the metaSEM package
# library(metaSEM)
# metaSEM_RE <- summary(meta(y = y, v = S))
# metaSEM_RE
# Plotting the result:
obj <- MMA_FE
# obj <- mvmeta_RE
# obj <- metaSEM_RE
# pdf("CI.pdf", width = 4, height = 7)
plotCI(y = computvcov$ef, v = computvcov$list.vcov,
        name.y = NULL, name.study = Craft2003$ID,
        y.all = obj$coefficients[,1],
        y.all.se = obj$coefficients[,2])
# dev.off()
##########################################################################
# Other effect sizes of the same or different type
# Choose variable SBP, DBP, DD, D with effect sizes "MD","MD","RD","lgOR"
##########################################################################
data(Geeganage2010)
## set the correlation coefficients list r
r12 <- 0.71
r13 <- 0.5
r14 <- 0.25
r23 <- 0.6
r24 <- 0.16
r34 <- 0.16
r <- vecTosm(c(r12, r13, r14, r23, r24, r34))
diag(r) <- 1
mix.r <- lapply(1:nrow(Geeganage2010), function(i){r})
attach(Geeganage2010)
## compute variance co-variances
computvcov <- mix.vcov(type = c("MD", "MD", "RD", "lgOR"),
                       d = cbind(MD_SBP, MD_DBP, NA, NA),
                       sdt = cbind(sdt_SBP, sdt_DBP, NA, NA),
                       sdc = cbind(sdc_SBP, sdc_DBP, NA, NA),
                       nt = cbind(nt_SBP, nt_DBP, nt_DD, nt_D),
                       nc = cbind(nc_SBP, nc_DBP, nc_DD, nc_D),
                       st = cbind(NA, NA, st_DD, st_D),
                       sc = cbind(NA, NA, sc_DD, sc_D),
                       r = mix.r,
                       name = c("MD.SBP", "MD.DBP", "RD.DD", "lgOR.D"))
# save different effect sizes in y
y <- computvcov$ef
head(y)
# save variances and covariances of all the effect sizes in a matrix S
S <- computvcov$matrix.vcov
S[1, ]
## fixed-effect model
MMA_FE <- summary(metafixed(y = y, Slist = computvcov$list.vcov))
# Restricted maximum likelihood (REML) estimator from the mixmeta package
# library(mixmeta)
# mvmeta_RE <- summary(mixmeta(cbind(MD.SBP, MD.DBP, RD.DD, lgOR.D) ~1,
#                             S = S, data = y, method = "reml"))
# mvmeta_RE

# maximum likelihood estimators from the metaSEM package
# library(metaSEM)
# metaSEM_RE <- summary(meta(y = y, v = S))
# metaSEM_RE
# Plotting the result:
obj <- MMA_FE
# obj <- mvmeta_RE
# obj <- metaSEM_RE
# pdf("CI.pdf", width = 4, height = 7)
plotCI(y = computvcov$ef, v = computvcov$list.vcov,
        name.y = NULL, name.study = Geeganage2010$studyID,
        y.all = obj$coefficients[,1],
        y.all.se = obj$coefficients[,2],
        hline = c(0, 0, 0, 1))
# dev.off()
#####################################################################################
# Multiple Imputation for missing data
#####################################################################################
# prepare a dataset with missing values and input arguments for meta.mi
Craft2003.mnar <- Craft2003[, c(2, 4:10)]
Craft2003.mnar[sample(which(Craft2003$C4 < 0), 6), "C4"] <- NA
dat <- Craft2003.mnar
n.name <- "N"
ef.name <- c("C1", "C2", "C3", "C4", "C5", "C6")
# fixed-effect model
obj <- metami(dat, M = 2, vcov = "r.vcov",
             n.name, ef.name,
             func = "metafixed")
# Plotting the result
computvcov <- r.vcov(n = Craft2003$N,
                     corflat = subset(Craft2003.mnar, select = C1:C6),
                     method = "average")
plotCI(y = computvcov$ef, v = computvcov$list.vcov,
        name.y = NULL, name.study = Craft2003$ID,
        y.all = obj$coefficients[,1],
        y.all.se = obj$coefficients[,2])
```

Craft2003 *Correlation Coefficients Data from the Craft et al. Meta-Analysis*

Description

This dataset includes 18 studies of correlation coefficients reported by Craft, Magyar, Becker, and Feltz (2003).

Usage

```
data(Craft2003)
```

Details

The primary purpose of Craft and colleagues' meta-analysis was to examine the interrelationships between athletic performance and three subscales, cognitive anxiety, somatic anxiety, and self-concept, of the Competitive State Anxiety Inventory (CSAI 2; CITATION). In this meta-analysis, the correlation coefficient was the primary effect size measure. For the purpose of demonstration, I use a subset of the data, i.e., six correlation coefficients among cognitive anxiety, somatic anxiety, self-concept, and sport performance in athletes.

ID	ID for each study included
N	Sample size from each study included
gender	Gender
p_male	Percentage of male
C1	Correlation coefficient between cognitive anxiety and somatic anxiety
C2	Correlation coefficient between cognitive anxiety and self concept
C3	Correlation coefficient between cognitive anxiety and athletic performance
C4	Correlation coefficient between somatic anxiety and self concept
C5	Correlation coefficient between somatic anxiety and athletic performance
C6	Correlation coefficient between self concept and athletic performance

Source

Craft, L. L., Magyar, T. M., Becker, B. J., & Feltz, D. L. (2003). The relationship between the competitive state anxiety inventory-2 and sport performance: a meta-analysis. *Journal of Sport and Exercise Psychology, 25*(1), 44-65.

Examples

```
data(Craft2003)
```

Geeganage2010 *Multivariate Effect Sizes of Different Types from the Geeganage et al. Meta-Analysis*

Description

This dataset includes 17 studies of multivariate effect sizes with four different outcomes reported by Geeganage and Bath (2010).

Usage

```
data(Geeganage2010)
```

Details

In a meta-analysis, Geeganage and Bath (2010) studied whether blood pressure (BP) should be actively altered during the acute phase of stroke, and assessed the effect of multiple vasoactive drugs on BP in acute stroke. Selection criteria included: Randomized trials of interventions that would be expected, on pharmacological grounds, to alter BP in patients within one week of the onset of acute stroke. There were four outcomes: systolic blood pressure (SBP, in mHg), diastolic blood pressure (DBP, in mHg), death (D), and death or disability (DD).

ID:	ID for each study included
ft_D	Number of early death within 1 month (D) in "1 Drug" Group
fc_D	Number of D in "control" Group
nt_D	Number of people in "1 Drug" Group reporting D status
nc_D	Number of people in "control" Group reporting D status
OR_D	Odds ratio of D for "1 Drug" versus "control" group
ft_DD	Number of early death or deterioration within 1 month (DD) in "1 Drug" Group
fc_DD	Number of early DD in "control" Group
nt_DD	Number of people in "1 Drug" Group reporting DD status
nc_DD	Number of people in "control" Group reporting DD status
OR_DD	Odds ratio of DD for "1 Drug" versus "control" group
nt_SBP	Number of people in "1 Drug" Group reporting Systolic blood pressure (SBP) status
nc_SBP	Number of people in "control" Group reporting SBP status
MD_SBP	Mean difference of SBP for "1 Drug" versus "control" group
sdt_SBP	Standard deviation of SBP in "1 Drug" Group
sdc_SBP	Standard deviation of SBP in "control" Group
nt_DBP	Number of people in "1 Drug" Group reporting Diastolic blood pressure (DBP) status
nc_DBP	Number of people in "control" Group reporting DBP status
MD_DBP	Mean difference of DBP for "1 Drug" versus "control" group
sdt_DBP	Standard deviation of DBP in "1 Drug" Group
sdc_DBP	Standard deviation of DBP in "control" Group
SMD_SBP	Standardized mean difference of SBP for "1 Drug" versus "control" group
SMD_DBP	Standardized mean difference of DBP for "1 Drug" versus "control" group

Source

Geeganage, C., & Bath, P. M. (2010). Vasoactive drugs for acute stroke. *Cochrane Database of Systematic Reviews* 2010(7).

Examples

```
data(Geeganage2010)
```

`lgOR.vcov` *Computing Variance-Covariance Matrices for Log Odds Ratios*

Description

The function `lgOR.vcov` computes effect sizes and variance-covariance matrix for multivariate meta-analysis when the effect sizes of interest are all measured by log odds ratio. See `mix.vcov` for effect sizes of the same or different types.

Usage

```
lgOR.vcov(r, nt, nc, st, sc, n_rt = NA, n_rc = NA)
```

Arguments

`r` A N-dimensional list of $p \times p$ correlation matrices for the p outcomes from the N studies. `r[[k]][i,j]` is the correlation coefficient between outcome i and outcome j from study k.

`nt` A $N \times p$ matrix storing sample sizes in the treatment group reporting the p outcomes. `nt[i,j]` is the sample size from study i reporting outcome j.

`nc` A matrix defined in a similar way as `nt` for the control group.

`st` A $N \times p$ matrix recording number of participants with event for all outcomes (dichotomous) in treatment group. `st[i,j]` reports number of participants with event for outcome j in treatment group for study i. If outcome j is not dichotomous, NA has to be imputed in column j.

`sc` Defined in a similar way as `st` for the control group.

`n_rt` A N-dimensional list of $p \times p$ matrices storing sample sizes in the treatment group reporting pairwise outcomes in the off-diagonal elements. `n_rt[[k]][i,j]` is the sample size reporting both outcome i and outcome j from study k. Diagonal elements of these matrices are discarded. The default value is `NA`, which means that the smaller sample size reporting the corresponding two outcomes is imputed: i.e. `n_rt[[k]][i,j]=min(nt[k,i],nt[k,j])`.

`n_rc` A list defined in a similar way as `n_rt` for the control group.

Value

`ef` A $N \times p$ data frame whose columns are computed log odds ratios.

`list.vcov` A N-dimensional list of $p(p+1)/2 \times p(p+1)/2$ matrices of computed variance-covariance matrices.

`matrix.vcov` A $N \times p(p+1)/2$ matrix whose rows are computed variance-covariance vectors.

Author(s)

Min Lu

References

Lu, M. (2023). Computing within-study covariances, data visualization, and missing data solutions for multivariate meta-analysis with metavcov. *Frontiers in Psychology*, 14:1185012.

Examples

```
##########################################################################
# Example: Geeganage2010 data
# Preparing log odds ratios and covariances for multivariate meta-analysis
##########################################################################
data(Geeganage2010)
## set the correlation coefficients list r
r12 <- 0.71
r.Gee <- lapply(1:nrow(Geeganage2010), function(i){matrix(c(1, r12, r12, 1), 2, 2)})


computvcov <- lgOR.vcov(nt = subset(Geeganage2010, select = c(nt_DD, nt_D)),
                     nc = subset(Geeganage2010, select = c(nc_DD, nc_D)),
                     st = subset(Geeganage2010, select = c(st_DD, st_D)),
                     sc = subset(Geeganage2010, select = c(sc_DD, sc_D)),
                     r = r.Gee)
# name computed log odds ratio as y
y <- computvcov$ef
colnames(y) <- c("lgOR.DD", "lgOR.D")
# name variance-covariance matrix of trnasformed z scores as S
S <- computvcov$matrix.vcov
## fixed-effect model
MMA_FE <- summary(metafixed(y = y, Slist = computvcov$list.vcov))
MMA_FE
#######################################################################
# Running random-effects model using package "mixmeta" or "metaSEM"
#######################################################################
# Restricted maximum likelihood (REML) estimator from the mixmeta package
#library(mixmeta)
#mvmeta_RE <- summary(mixmeta(cbind(lgOR.DD, lgOR.D)~1,
#                            S = S,
#                            data = as.data.frame(y),
#                            method = "reml"))
#mvmeta_RE
```

```
# maximum likelihood estimators from the metaSEM package
# library(metaSEM)
# metaSEM_RE <- summary(meta(y = y, v = S))
# metaSEM_RE
###############################################################
# Plotting the result:
###############################################################
obj <- MMA_FE
# obj <- mvmeta_RE
# obj <- metaSEM_RE
# pdf("CI.pdf", width = 4, height = 7)
plotCI(y = computvcov$ef, v = computvcov$list.vcov,
        name.y = c("lgOR.DD", "lgOR.D"),
        name.study = Geeganage2010$studyID,
        y.all = obj$coefficients[,1],
        y.all.se = obj$coefficients[,2],
        hline = 1)
# dev.off()
```

lgor_lgrr *Computing Covariance between Log Odds Ratio and Log Risk Ratio*

Description

The function lgor_lgrr computes covariance between log odds ratio and log risk ratio, when the two outcomes are binary. See mix.vcov for effect sizes of the same or different types.

Usage

```
lgor_lgrr(r, n1c, n2c, n1t, n2t,
           n12c = min(n1c, n2c),
           n12t = min(n1t, n2t),
           s2c, s2t, f2c, f2t, s1c, s1t, f1t, f1c)
```

Arguments

r Correlation coefficient of the two outcomes.

n1c Number of participants reporting outcome 1 in the control group.

n2c Number of participants reporting outcome 2 in the control group.

n1t Number of participants reporting outcome 1 in the treatment group.

n2t Number of participants reporting outcome 2 in the treatment group.

n12c Number of participants reporting both outcome 1 and outcome 2 in the control group. By default, it is equal to the smaller number between n1c and n2c.

n12t Defined in a similar way as n12c for the treatment group.

s2c Number of participants with event for outcome 2 (dichotomous) in the control group.

`s2t` Defined in a similar way as `s2c` for the treatment group.

`f2c` Number of participants without event for outcome 2 (dichotomous) in the control group.

`f2t` Defined in a similar way as `f2c` for the treatment group.

`s1c` Number of participants with event for outcome 1 (dichotomous) in the control group.

`s1t` Defined in a similar way as `s1c` for the treatment group.

`f1c` Number of participants without event for outcome 1 (dichotomous) in the control group.

`f1t` Defined in a similar way as `f1c` for the treatment group.

Value

`lgor` Log odds ratio for outcome 1.

`lgrr` Log risk ratio for outcome 2.

`v` Computed covariance.

Author(s)

Min Lu

References

Lu, M. (2023). Computing within-study covariances, data visualization, and missing data solutions for multivariate meta-analysis with metavcov. *Frontiers in Psychology*, 14:1185012.

Examples

```
lgor_lgrr(r = 0.71,
          n1c = 30, n2c = 35, n1t = 28, n2t = 32,
          s2c = 5, s2t = 8, f2c = 30, f2t = 24,
          s1c = 5, s1t = 8, f1c = 25, f1t = 20)
## calculate covariances for variable D and DD in Geeganage2010 data
attach(Geeganage2010)
D_DD <- unlist(lapply(1:nrow(Geeganage2010),
              function(i){lgor_lgrr(r = 0.71, n1c = nc_SBP[i], n2c = nc_DD[i],
                 n1t = nt_SBP[i], n2t = nt_DD[i], s2t = st_DD[i], s2c = sc_DD[i],
                 f2c = nc_DD[i] - sc_DD[i], f2t = nt_DD[i] - st_DD[i],
                 s1t = st_D[i], s1c = sc_D[i],
                 f1c = nc_D[i] - sc_D[i], f1t = nt_D[i] - st_D[i])$v}))
D_DD
## the function mix.vcov() should be used for dataset
```

`lgor_rd` *Computing Covariance between Log Odds Ratio and Risk Difference*

Description

The function `lgor_rd` computes covariance between log odds ratio and risk difference, when the two outcomes are binary. See `mix.vcov` for effect sizes of the same or different types.

Usage

```
lgor_rd(r, n1c, n2c, n1t, n2t,
        n12c = min(n1c, n2c), n12t = min(n1t, n2t),
        s2c, s2t, f2c, f2t, s1c, s1t, f1t, f1c)
```

Arguments

`r` Correlation coefficient of the two outcomes.

`n1c` Number of participants reporting outcome 1 in the control group.

`n2c` Number of participants reporting outcome 2 in the control group.

`n1t` Number of participants reporting outcome 1 in the treatment group.

`n2t` Number of participants reporting outcome 2 in the treatment group.

`n12c` Number of participants reporting both outcome 1 and outcome 2 in the control group. By default, it is equal to the smaller number between `n1c` and `n2c`.

`n12t` Number defined in a similar way as `n12c` for treatment group.

`s2c` Number of participants with event for outcome 2 (dichotomous) in the control group.

`s2t` Defined in a similar way as `s2c` for treatment group.

`f2c` Number of participants without event for outcome 2 (dichotomous) in the control group.

`f2t` Defined in a similar way as `f2c` for treatment group

`s1c` Number of participants with event for outcome 1 (dichotomous) in the control group.

`s1t` Defined in a similar way as `s1c` for treatment group.

`f1c` Number of participants without event for outcome 1 (dichotomous) in the control group.

`f1t` Defined in a similar way as `f1c` for treatment group.

Value

`lgor` Log odds ratio for outcome 1.

`rd` Risk difference for outcome 2.

`v` Computed covariance.

Author(s)

Min Lu

References

Lu, M. (2023). Computing within-study covariances, data visualization, and missing data solutions for multivariate meta-analysis with metavcov. *Frontiers in Psychology*, 14:1185012.

Examples

```
## simple example
lgor_rd(r = 0.71, n1c = 30, n2c = 35, n1t = 28, n2t = 32,
        s2c = 5, s2t = 8, f2c = 30, f2t = 24,
        s1c = 5, s1t = 8, f1c = 25, f1t = 20)
## calculate covariances for variable D and DD in Geeganage2010 data
attach(Geeganage2010)
D_DD <- unlist(lapply(1:nrow(Geeganage2010), function(i){lgor_rd(r = 0.71,
              n1c = nc_SBP[i], n2c = nc_DD[i],
              n1t = nt_SBP[i], n2t = nt_DD[i], s2t = st_DD[i], s2c = sc_DD[i],
              f2c = nc_DD[i] - sc_DD[i], f2t = nt_DD[i] - st_DD[i],
              s1t = st_D[i], s1c = sc_D[i],
              f1c = nc_D[i] - sc_D[i], f1t = nt_D[i] - st_D[i])$v}))
D_DD
## the function mix.vcov() should be used for dataset
```

lgRR.vcov *Computing Variance-Covariance Matrices for Log Risk Ratios*

Description

The function `lgOR.vcov` computes effect sizes and variance-covariance matrix for multivariate meta-analysis when the effect sizes of interest are all measured by log risk ratio (or log relative risk). See `mix.vcov` for effect sizes of the same or different types.

Usage

```
lgRR.vcov(r, nt, nc, st, sc, n_rt = NA, n_rc = NA)
```

Arguments

`r` A N-dimensional list of $p \times p$ correlation matrices for the p outcomes from the N studies. `r[[k]][i,j]` is the correlation coefficient between outcome i and outcome j from study k.

`nt` A $N \times p$ matrix storing sample sizes in the treatment group reporting the p outcomes. `nt[i,j]` is the sample size from study i reporting outcome j.

`nc` A matrix defined in a similar way as `nt` for the control group.

`st` A $N \times p$ matrix recording number of participants with event for all outcomes (dichotomous) in treatment group. `st[i,j]` reports number of participants with event for outcome j in treatment group for study i. If outcome j is not dichotomous, NA has to be imputed in column j.

`sc` Defined in a similar way as `st` for the control group.

`n_rt` A N-dimensional list of $p \times p$ matrices storing sample sizes in the treatment group reporting pairwise outcomes in the off-diagonal elements. `n_rt[[k]][i,j]` is the sample size reporting both outcome i and outcome j from study k. Diagonal elements of these matrices are discarded. The default value is `NA`, which means that the smaller sample size reporting the corresponding two outcomes is imputed: i.e. `n_rt[[k]][i,j]=min(nt[k,i],nt[k,j])`.

`n_rc` A list defined in a similar way as `n_rt` for the control group.

Value

`ef` A $N \times p$ data frame whose columns are computed log risk ratios.

`list.vcov` A N-dimensional list of $p(p+1)/2 \times p(p+1)/2$ matrices of computed variance-covariance matrices.

`matrix.vcov` A $N \times p(p+1)/2$ matrix whose rows are computed variance-covariance vectors.

Author(s)

Min Lu

References

Lu, M. (2023). Computing within-study covariances, data visualization, and missing data solutions for multivariate meta-analysis with metavcov. *Frontiers in Psychology*, 14:1185012.

Examples

```
##########################################################################
# Example: Geeganage2010 data
# Preparing log risk ratios and covariances for multivariate meta-analysis
##########################################################################
data(Geeganage2010)
## set the correlation coefficients list r
r12 <- 0.71
r.Gee <- lapply(1:nrow(Geeganage2010), function(i){matrix(c(1, r12, r12, 1), 2, 2)})


computvcov <- lgRR.vcov(nt = subset(Geeganage2010, select = c(nt_DD, nt_D)),
                      nc = subset(Geeganage2010, select = c(nc_DD, nc_D)),
                      st = subset(Geeganage2010, select=c(st_DD, st_D)),
                      sc = subset(Geeganage2010, select=c(sc_DD, sc_D)),
                      r = r.Gee)
# name computed log risk ratio as y
y <- computvcov$ef
colnames(y) = c("lgRR.DD", "lgRR.D")
```

```
# name variance-covariance matrix of trnasformed z scores as covars
S <- computvcov$matrix.vcov
## fixed-effect model
MMA_FE <- summary(metafixed(y = y, Slist = computvcov$list.vcov))
MMA_FE
#######################################################################
# Running random-effects model using package "mixmeta" or "metaSEM"
#######################################################################
#library(mixmeta)
#mvmeta_RE = summary(mixmeta(cbind(lgRR.DD, lgRR.D)~1,
#                            S = S, data = as.data.frame(y),
#                            method = "reml"))
#mvmeta_RE

# maximum likelihood estimators from the metaSEM package
# library(metaSEM)
# metaSEM_RE <- summary(meta(y = y, v = S))
# metaSEM_RE
##############################################################
# Plotting the result:
##############################################################
obj <- MMA_FE
# obj <- mvmeta_RE
# obj <- metaSEM_RE
# pdf("CI.pdf", width = 4, height = 7)
plotCI(y = computvcov$ef, v = computvcov$list.vcov,
        name.y = c("lgRR.DD", "lgRR.D"),
        name.study = Geeganage2010$studyID,
        y.all = obj$coefficients[,1],
        y.all.se = obj$coefficients[,2],
        hline = 1)
# dev.off()
```

lgrr_rd *Computing Covariance between Log Risk Ratio and Risk Difference*

Description

The function `lgrr_rd` compute covariance between log risk ratio and risk difference, when the two outcomes are binary. See `mix.vcov` for effect sizes of the same or different types.

Usage

```
lgrr_rd(r, n1c, n2c, n1t, n2t,
        n12c = min(n1c, n2c),
        n12t = min(n1t, n2t),
        s2c, s2t, f2c, f2t,
        s1c, s1t, f1c, f1t)
```

Arguments

r Correlation coefficient of the two outcomes.

n1c Number of participants reporting outcome 1 in the control group.

n2c Number of participants reporting outcome 2 in the control group.

n1t Number of participants reporting outcome 1 in the treatment group.

n2t Number of participants reporting outcome 2 in the treatment group.

n12c Number of participants reporting both outcome 1 and outcome 2 in the control group. By default, it is equal to the smaller number between n1c and n2c.

n12t Defined in a similar way as n12c for the treatment group.

s2c Number of participants with event for outcome 2 (dichotomous) in the control group.

s2t Defined in a similar way as s2c for the treatment group.

f2c Number of participants without event for outcome 2 (dichotomous) in the control group.

f2t Defined in a similar way as f2c for the treatment group.

s1c Number of participants with event for outcome 1 (dichotomous) in the control group.

s1t Defined in a similar way as s1c for the treatment group.

f1c Number of participants without event for outcome 1 (dichotomous) in the control group.

f1t Defined in a similar way as f1c for the treatment group.

Value

lgrr Log risk ratio for outcome 1.

rd Risk difference for outcome 1.

v Computed covariance.

Author(s)

Min Lu

References

Lu, M. (2023). Computing within-study covariances, data visualization, and missing data solutions for multivariate meta-analysis with metavcov. *Frontiers in Psychology*, 14:1185012.

Examples

```
## simple example
lgrr_rd(r = 0.71, n1c = 30, n2c = 35, n1t = 28, n2t = 32,
        s2c = 5, s2t = 8, f2c = 30, f2t = 24,
        s1c = 5, s1t = 8, f1c = 25, f1t = 20)
## calculate covariances for variable D and DD in Geeganage2010 data
```

```
attach(Geeganage2010)
D_DD <- unlist(lapply(1:nrow(Geeganage2010), function(i){lgrr_rd(r = 0.71,
        n1c = nc_SBP[i], n2c = nc_DD[i],
        n1t = nt_SBP[i], n2t = nt_DD[i], s2t = st_DD[i], s2c = sc_DD[i],
        f2c = nc_DD[i] - sc_DD[i], f2t = nt_DD[i] - st_DD[i],
      s1t = st_D[i], s1c = sc_D[i], f1c = nc_D[i] - sc_D[i], f1t = nt_D[i] - st_D[i])$v}))
D_DD
## the function mix.vcov() should be used for dataset
```

md.vcov *Computing Variance-Covariance Matrices for Mean Differences*

Description

The function `md.vcov` computes effect sizes and variance-covariance matrix for multivariate meta-analysis when the effect sizes of interest are all measured by mean difference. See `mix.vcov` for effect sizes of the same or different types.

Usage

```
md.vcov(r, nt, nc, n_rt = NA, n_rc = NA, sdt, sdc)
```

Arguments

`r` A N-dimensional list of $p \times p$ correlation matrices for the p outcomes from the N studies. `r[[k]][i,j]` is the correlation coefficient between outcome i and outcome j from study k.

`nt` A $N \times p$ matrix storing sample sizes in the treatment group reporting the p outcomes. `nt[i,j]` is the sample size from study i reporting outcome j.

`nc` A matrix defined in a similar way as `nt` for the control group.

`n_rt` A N-dimensional list of $p \times p$ matrices storing sample sizes in the treatment group reporting pairwise outcomes in the off-diagonal elements. `n_rt[[k]][i,j]` is the sample size reporting both outcome i and outcome j from study k. Diagonal elements of these matrices are discarded. The default value is `NA`, which means that the smaller sample size reporting the corresponding two outcomes is imputed: i.e. `n_rt[[k]][i,j]=min(nt[k,i],nt[k,j])`.

`n_rc` A list defined in a similar way as `n_rt` for the control group.

`sdt` A $N \times p$ matrix storing sample standard deviations for each outcome from treatment group. `sdt[i,j]` is the sample standard deviation from study i for outcome j. If outcome j is not continuous such as MD or SMD, NA has to be imputed in the jth column.

`sdc` A matrix defined in a similar way as `sdt` for the control group.

Value

`list.vcov` A N-dimensional list of $p(p+1)/2 \times p(p+1)/2$ matrices of computed variance-covariance matrices.

`matrix.vcov` A $N \times p(p+1)/2$ matrix whose rows are computed variance-covariance vectors.

Author(s)

Min Lu

References

Lu, M. (2023). Computing within-study covariances, data visualization, and missing data solutions for multivariate meta-analysis with metavcov. *Frontiers in Psychology*, 14:1185012.

Examples

```
######################################################
# Example: Geeganage2010 data
# Preparing covariances for multivariate meta-analysis
######################################################
## set the correlation coefficients list r
r12 <- 0.71
r.Gee <- lapply(1:nrow(Geeganage2010), function(i){matrix(c(1, r12, r12, 1), 2, 2)})

computvcov <- md.vcov(nt = subset(Geeganage2010, select = c(nt_SBP, nt_DBP)),
                    nc = subset(Geeganage2010, select = c(nc_SBP, nc_DBP)),
                    sdt = subset(Geeganage2010, select=c(sdt_SBP, sdt_DBP)),
                    sdc = subset(Geeganage2010, select=c(sdc_SBP, sdc_DBP)),
                    r = r.Gee)
# name variance-covariance matrix as S
S <- computvcov$matrix.vcov
## fixed-effect model
y <- as.data.frame(subset(Geeganage2010, select = c(MD_SBP, MD_DBP)))
MMA_FE <- summary(metafixed(y = y, Slist = computvcov$list.vcov))
MMA_FE
#######################################################################
# Running random-effects model using package ”mixmeta” or ”metaSEM”
#######################################################################
# Restricted maximum likelihood (REML) estimator from the mixmeta package
#library(mixmeta)
#mvmeta_RE <- summary(mixmeta(cbind(MD_SBP, MD_DBP)~1, S = S,
#                         data = subset(Geeganage2010, select = c(MD_SBP, MD_DBP)),
#                         method = ”reml”))
#mvmeta_RE

# maximum likelihood estimators from the metaSEM package
# library(metaSEM)
# metaSEM_RE <- summary(meta(y = y, v = S))
# metaSEM_RE
##############################################################
# Plotting the result:
##############################################################
# obj <- MMA_FE
# obj <- mvmeta_RE
# obj <- metaSEM_RE

# plotCI(y = y, v = computvcov$list.vcov,
#         name.y = c(”MD_SBP”, ”MD_DBP”), name.study = Geeganage2010$studyID,
#        y.all = obj$coefficients[,1],
#        y.all.se = obj$coefficients[,2])
```

md_lgor *Computing Covariance between Mean Difference and Log Odds Ratio*

Description

The function lgor_rd computes covariance between mean difference and log odds ratio. See mix.vcov for effect sizes of the same or different types.

Usage

```
md_lgor(r, n1c, n2c, n1t, n2t,
        n12c = min(n1c, n2c), n12t = min(n1t, n2t),
        s2c, s2t, f2c, f2t, sd1c, sd1t)
```

Arguments

r	Correlation coefficient of the two outcomes.
n1c	Number of participants reporting outcome 1 in the control group.
n2c	Number of participants reporting outcome 2 in the control group.
n1t	Number of participants reporting outcome 1 in the treatment group.
n2t	Number of participants reporting outcome 2 in the treatment group.
n12c	Number of participants reporting both outcome 1 and outcome 2 in the control group. By default, it is equal to the smaller number between n1c and n2c.
n12t	Number defined in a similar way as n12c for the treatment group.
s2c	Number of participants with event for outcome 2 (dichotomous) in the control group.
s2t	Defined in a similar way as s2c for the treatment group.
f2c	Number of participants without event for outcome 2 (dichotomous) in the control group.
f2t	Defined in a similar way as f2c for the treatment group.
sd1c	Sample standard deviation of outcome 1 for the control group.
sd1t	Defined in a similar way as sd1c for the treatment group.

Value

lgor	Log odds ratio for outcome 2.
v	Computed covariance.

Author(s)

Min Lu

References

Lu, M. (2023). Computing within-study covariances, data visualization, and missing data solutions for multivariate meta-analysis with metavcov. *Frontiers in Psychology*, 14:1185012.

Examples

```
## a simple example
md_lgor(r = 0.71, n1c = 34, n2c = 35, n1t = 25, n2t = 32,
        s2c = 5, s2t = 8, f2c = 30, f2t = 24, sd1t = 0.4, sd1c = 8)
## calculate covariances for variable SBP and DD in Geeganage2010 data
attach(Geeganage2010)
SBP_DD <- unlist(lapply(1:nrow(Geeganage2010), function(i){md_lgor(r = 0.71,
               n1c = nc_SBP[i], n2c = nc_DD[i], n1t = nt_SBP[i], n2t = nt_DD[i],
               sd1t = sdt_SBP[i], s2t = st_DD[i], sd1c = sdc_SBP[i], s2c = sc_DD[i],
               f2c = nc_DD[i] - sc_DD[i], f2t = nt_DD[i] - st_DD[i])$v}))
SBP_DD
## the function mix.vcov() should be used for dataset
```

md_lgrr *Computing Covariance between Mean Difference and Log Risk Ratio*

Description

The function `md_lgrr` computes covariance between mean difference and log risk ratio. See `mix.vcov` for effect sizes of the same or different types.

Usage

```
md_lgrr(r, n1c, n2c, n1t, n2t,
        n12c = min(n1c, n2c), n12t = min(n1t, n2t),
        s2c, s2t, f2c, f2t, sd1c, sd1t)
```

Arguments

`r` Correlation coefficient of the two outcomes.

`n1c` Number of participants reporting outcome 1 in the control group.

`n2c` Number of participants reporting outcome 2 in the control group.

`n1t` Number of participants reporting outcome 1 in the treatment group.

`n2t` Number of participants reporting outcome 2 in the treatment group.

`n12c` Number of participants reporting both outcome 1 and outcome 2 in the control group. By default, it is equal to the smaller number between `n1c` and `n2c`.

`n12t` Number defined in a similar way as `n12c` for the treatment group.

`s2c` Number of participants with event for outcome 2 (dichotomous) in the control group.

`s2t` Defined in a similar way as `s2c` for the treatment group.

`f2c` Number of participants without event for outcome 2 (dichotomous) in the control group.

`f2t` Defined in a similar way as `f2c` for the treatment group.

`sd1c` Sample standard deviation of outcome 1 for the control group.

`sd1t` Defined in a similar way as `sd1c` for the treatment group.

Value

`lgrr` Log risk ratio for outcome 2.

`v` Computed covariance.

Author(s)

Min Lu

References

Lu, M. (2023). Computing within-study covariances, data visualization, and missing data solutions for multivariate meta-analysis with metavcov. *Frontiers in Psychology*, 14:1185012.

Examples

```
## a simple example
md_lgrr(r = 0.71, n1c = 34, n2c = 35, n1t = 25, n2t = 32,
        s2c = 5, s2t = 8, f2c = 30, f2t = 24, sd1t = 0.4, sd1c = 8)
## calculate covariances for variable SBP and DD in Geeganage2010 data
attach(Geeganage2010)
SBP_DD <- unlist(lapply(1:nrow(Geeganage2010), function(i){md_lgrr(r = 0.71,
                n1c = nc_SBP[i], n2c = nc_DD[i], n1t = nt_SBP[i], n2t = nt_DD[i],
                sd1t = sdt_SBP[i], s2t = st_DD[i], sd1c = sdc_SBP[i], s2c = sc_DD[i],
                f2c = nc_DD[i] - sc_DD[i], f2t = nt_DD[i] - st_DD[i])$v}))
SBP_DD
## the function mix.vcov() should be used for dataset
```

md_rd *Computing Covariance between Mean Difference and Risk Difference*

Description

The function `lgor_rd` computes covariance between mean difference and risk difference. See `mix.vcov` for effect sizes of the same or different types.

Usage

```
md_rd(r, n1c, n2c, n1t, n2t,
      n12c = min(n1c, n2c), n12t = min(n1t, n2t),
      s2c, s2t, f2c, f2t, sd1c, sd1t)
```

Arguments

`r` Correlation coefficient of the two outcomes.

`n1c` Number of participants reporting outcome 1 in the control group.

`n2c` Number of participants reporting outcome 2 in the control group.

`n1t` Number of participants reporting outcome 1 in the treatment group.

`n2t` Number of participants reporting outcome 2 in the treatment group.

`n12c` Number of participants reporting both outcome 1 and outcome 2 in the control group. By default, it is equal to the smaller number between `n1c` and `n2c`.

`n12t` Number defined in a similar way as `n12c` for the treatment group.

`s2c` Number of participants with event for outcome 2 (dichotomous) in the control group.

`s2t` Defined in a similar way as `s2c` for the treatment group.

`f2c` Number of participants without event for outcome 2 (dichotomous) in the control group.

`f2t` Defined in a similar way as `f2c` for the treatment group.

`sd1c` Sample standard deviation of outcome 1 for the control group.

`sd1t` Defined in a similar way as `sd1c` for the treatment group.

Value

`rd` Computed risk difference for outcome 2.

`v` Computed covariance.

Author(s)

Min Lu

References

Lu, M. (2023). Computing within-study covariances, data visualization, and missing data solutions for multivariate meta-analysis with metavcov. *Frontiers in Psychology*, 14:1185012.

Examples

```
## a simple example
md_rd(r = 0.71, n1c = 34, n2c = 35, n1t = 25, n2t = 32,
      s2c = 5, s2t = 8, f2c = 30, f2t = 24, sd1t = 0.4, sd1c = 8)
## calculate covariances for variable SBP and DD in Geeganage2010 data
attach(Geeganage2010)
SBP_DD <- unlist(lapply(1:nrow(Geeganage2010), function(i){md_rd(r = 0.71,
               n1c = nc_SBP[i], n2c = nc_DD[i], n1t = nt_SBP[i], n2t = nt_DD[i],
               sd1t = sdt_SBP[i], s2t = st_DD[i], sd1c = sdc_SBP[i], s2c = sc_DD[i],
               f2c = nc_DD[i] - sc_DD[i], f2t = nt_DD[i] - st_DD[i])$v}))
SBP_DD
## the function mix.vcov() should be used for dataset
```

md_smd *Computing Covariance between Mean Difference and Standardized Mean Difference*

Description

The function `lgor_rd` computes covariance between mean difference and standardized mean difference. See `mix.vcov` for effect sizes of the same or different types.

Usage

```
md_smd(smd, r, n1c, n2c, n1t, n2t,
       n12c = min(n1c, n2c), n12t = min(n1t, n2t),
       sd1t, sd2t, sd1c, sd2c)
```

Arguments

`smd`	Standardized mean difference for outcome 2.
`r`	Correlation coefficient of the two outcomes.
`n1c`	Number of participants reporting outcome 1 in the control group.
`n2c`	Number of participants reporting outcome 2 in the control group.
`n1t`	Number of participants reporting outcome 1 in the treatment group.
`n2t`	Number of participants reporting outcome 2 in the treatment group.
`n12c`	Number of participants reporting both outcome 1 and outcome 2 in the control group. By default, it is equal to the smaller number between `n1c` and `n2c`.
`n12t`	Number defined in a similar way as `n12c` for treatment group.
`sd1t`	Sample standard deviation of outcome 1 for the treatment group.
`sd2t`	Sample standard deviation of outcome 2 for the treatment group.
`sd1c`	Defined in a similar way as `sd1t` for the control group.
`sd2c`	Defined in a similar way as `sd2t` for the control group.

Value

`g`	Computed Hedge's g from the input argument smd for outcome 2.
`v`	Computed covariance.

Author(s)

Min Lu

References

Lu, M. (2023). Computing within-study covariances, data visualization, and missing data solutions for multivariate meta-analysis with metavcov. *Frontiers in Psychology*, 14:1185012.

Examples

```
## a simple example
md_smd(smd = 1, r = 0.71, n1c = 34, n2c = 35, n1t = 25, n2t = 32,
       sd1t = 0.6, sd2t = 0.4, sd1c = 8, sd2c = 0.9)
## calculate covariances for variable SBP and DBP in Geeganage2010 data
attach(Geeganage2010)
SBP_DBP <- unlist(lapply(1:nrow(Geeganage2010), function(i){md_smd(smd = SMD_DBP, r = 0.71,
                n1c = nc_SBP[i], n2c = nc_DBP[i], n1t = nt_SBP[i], n2t = nt_DBP[i],
                sd1t = sdt_SBP[i], sd2t = sdt_DBP[i],
                sd1c = sdc_SBP[i], sd2c = sdc_SBP[i])$v}))
SBP_DBP
## the function mix.vcov() should be used for dataset
```

metafixed *Fitting Fixed-Effect Meta-Analysis Models*

Description

The function metafixed performs fixed-effects multivariate meta-analysis with the generalized least squares (GLS) method.

Usage

```
metafixed(y, Slist)
```

Arguments

y — A $N \times p$ matrix or data frame that stores effect sizes from N primary studies. Usually the output value ef produced by r.vcov for correlation coefficients or mix.vcov for other types of effect sizes.

Slist — A N-dimensional list of $p(p+1)/2 \times p(p+1)/2$ matrices that stores within-study (co)variance matrices of the estimated effect sizes for each one of the N studies. Usually the output value list.vcov produced by r.vcov for correlation coefficients or mix.vcov for other types of effect sizes.

Details

Estimators were calculated from the generalized least squares approach.

Value

The metafixed function typically returns a list object of class "metafixed" representing the meta-analytical model. Use the summary function to check the analysis results.

Author(s)

Min Lu

References

Lu, M. (2023). Computing within-study covariances, data visualization, and missing data solutions for multivariate meta-analysis with metavcov. *Frontiers in Psychology*, 14:1185012.

Cooper, H., Hedges, L.V., & Valentine, J.C. (Eds.) (2009). *The handbook of research synthesis and meta-analysis.* New York: Russell Sage Foundation.

Examples

```
######################################################
# Example: Craft2003 data
# Preparing covariances for multivariate meta-analysis
######################################################
data(Craft2003)
computvcov <- r.vcov(n = Craft2003$N,
                     corflat = subset(Craft2003, select = C1:C6),
                     method = "average")
y <- computvcov$ef
Slist <- computvcov$list.vcov
#####################################################
# Running fixed-effects model using "metafixed"
#####################################################
MMA_FE <- summary(metafixed(y = y, Slist = Slist))
MMA_FE$coefficients
##############################################################
# Plotting the result:
##############################################################
plotCI(y = computvcov$ef, v = computvcov$list.vcov,
        name.y = NULL, name.study = Craft2003$ID,
        y.all = MMA_FE$coefficients[,1],
        y.all.se = MMA_FE$coefficients[,2],
        up.bound = Inf, low.bound = -Inf)
```

metami — *Multiple Imputation for Missing Data in Meta-Analysis*

Description

Multiple imputation allows for the uncertainty about the missing data by generating several different plausible imputed data sets and appropriately combining results obtained from each of them. Let $\hat{\theta}_{*m}$ be the estimated coefficient from the mth imputed dataset for one of the p dimensions in the multivariate outcome, where $m = 1, \dots, M$. The coefficient from MI $\bar{\theta}$ is simply just an arithmetic mean of the individual coefficients estimated from each of the M meta-analysis. We have

$$\bar{\theta} = \frac{\sum_{m=1}^{M} \hat{\theta}_{*m}}{M}.$$

Estimation of the standard error for each variable is little more complicated. Let V_W be the within imputation variance, which is the average of the variance of the estimated coefficient from each imputed dateset:

$$V_W = \frac{\sum_{m=1}^{M} V(\hat{\theta}_{*m})}{M},$$

where $V(\hat{\theta}_{*m})$ is the variance of the estimator calculated from generalized least squares methods using the imputed dataset. Let V_B be the between imputation variance, which is calculated as

$$V_B = \frac{\sum_{m=1}^{M} (\hat{\theta}_{*m} - \bar{\theta})^2}{M - 1}.$$

From V_W and V_B, the variance of the pooled coefficients is calculated as

$$V(\bar{\theta}) = V_W + V_B + \frac{V_B}{M}$$

The above variance is statistically principled since V_W reflects the sampling variance and V_B reflects the extra variance due to the missing data.

Usage

```
metami(data, M = 20, vcov = "r.vcov",
        r.n.name, ef.name, x.name = NULL,
        rvcov.method = "average", rvcov.zscore = TRUE,
        type = NULL,
        d = NULL, sdt = NULL, sdc = NULL,
        nt = NULL, nc = NULL,
        st = NULL, sc = NULL,
        n_rt = NA, n_rc = NA,
        r = NULL,
        func = "mixmeta",
        formula = NULL,
        method = "fixed",
        pool.seq = NULL,
        return.mi = FALSE,
        ci.level = 0.95)
```

Arguments

`data`	A $N \times p$ data frame that contains effect sizes and predictors for meta-regression, if any.
`M`	Number of imputed data sets.
`vcov`	Method for computing effect sizes; options including `vcov = "r.vcov"` for correlation coefficients and `vcov = "mix.vcov"` for other types of effect sizes. See `r.vcov` and `mix.vcov` for details.
`r.n.name`	A string defining the column name for sample sizes in `data` when the effect sizes are correlation coefficients (`vcov = "r.vcov"`).
`ef.name`	A p-dimensional vector that stores the column names for sample sizes in `data` when the effect sizes are correlation coefficients (`vcov = "r.vcov"`).
`x.name`	A vector that stores the column names in `data` for predictors for meta-regression.

`rvcov.method` Method used for `r.vcov`; options including `"average"` and `"each"`.

`rvcov.zscore` Whether the correlation coefficients in `data` are already transformed into Fisher's z scores.

`type` A p-dimensional vector indicating types of effect sizes for the argument `vcov = "mix.vcov"`. `"MD"` stands for mean difference, `"SMD"` stands for standardized mean difference, `"logOR"` stands for log odds ratio, `"logRR"` stands for log risk ratio, and `"RD"` stands for risk difference.

`d` A p-dimensional vector that stores the column names in `data` for continuous effect sizes such as MD or SMD. If outcome j is dichotomous, NA has to be imputed in for $d[j]$.

`sdt` A p-dimensional vector that stores the column names in `data` for the sample standard deviations of each outcome from the treatment group. If outcome j is dichotomous, NA has to be imputed in for $d[j]$.

`sdc` A vector defined in a similar way as `sdt` for the control group.

`nt` A p-dimensional vector that stores the column names in `data` for sample sizes of p outcomes from treatment group.

`nc` A vector defined in a similar way as `nt` for the control group.

`st` A p-dimensional vector that stores the column names in `data` for the number of participants with event for all outcomes (dichotomous) in the treatment group. If outcome j is dichotomous, NA has to be imputed in for $st[j]$.

`sc` A vector defined in a similar way as `st` for the control group.

`n_rt` A N-dimensional list of $p \times p$ correlation matrices storing sample sizes in the treatment group reporting pairwise outcomes in the off-diagonal elements. See `mix.vcov` for details.

`n_rc` A list defined in a similar way as `n_rt` for the control group.

`r` A N-dimensional list of $p \times p$ correlation matrices for the p outcomes from the N studies. See `mix.vcov` for details.

`func` A string defining the function to be used for fitting the meta-analysis. Options include `func = "metafixed"` for fixed-effect meta-analysis (see `metafixed` for details). `func = "mixmeta"`, for which the `mixmeta` package must be installed beforehand, and `func = "meta"`, for which the `metaSEM` package must be installed beforehand.

`formula` Formula used for the function `func = "mixmeta"` from the `mixmeta` package when `func = "mixmeta"`.

`method` Method used for the function `func = "mixmeta"` from the `mixmeta` package when `func = "mixmeta"`.

`pool.seq` A numeric vector indicating if the results are pooled from subsets of the `M` data sets. By default, the results are only pooled from all `M` data sets.

`return.mi` Should the `M` imputed data sets be returned?

`ci.level` Significant level for the pooled confidence intervals. The default is 0.05.

Details

For the imputation phase, this function imports the `mice` package that imputes incomplete multivariate data by chained equations. The pooling phase is performed via the Rubin's rules.

Value

`coefficients` A data.frame that contains the pooled results from the `M` imputed data sets.

`results.mi` A M-dimensional list of results from each imputed data set.

`data.mi` A M-dimensional list of imputed data sets if the argument `return.mi` = `TRUE`.

`result.seq` A list of results from the pooled results from the subsets of the `M` imputed data sets if the argument `pool.seq` = `TRUE`.

Author(s)

Min Lu

References

Lu, M. (2023). Computing within-study covariances, data visualization, and missing data solutions for multivariate meta-analysis with metavcov. *Frontiers in Psychology*, 14:1185012.

Van Buuren, S. and Groothuis-Oudshoorn, K., 2011. mice: Multivariate imputation by chained equations in R. *Journal of statistical software*, 45(1), pp.1-67.

Examples

```
#####################################################################################
# Example: Craft2003 data
# Preparing input arguments for meta.mi() and fixed-effect model
#####################################################################################
# prepare a dataset with missing values and input arguments for meta.mi
Craft2003.mnar <- Craft2003[, c(2, 4:10)]
Craft2003.mnar[sample(which(Craft2003$C4 < 0), 6), "C4"] <- NA
dat <- Craft2003.mnar
n.name <- "N"
ef.name <- c("C1", "C2", "C3", "C4", "C5", "C6")
# fixed-effect model
obj <- metami(dat, M = 2, vcov = "r.vcov",
             n.name, ef.name,
             func = "metafixed")

########################
# Plotting the result
########################
computvcov <- r.vcov(n = Craft2003$N,
                     corflat = subset(Craft2003.mnar, select = C1:C6),
                     method = "average")
plotCI(y = computvcov$ef, v = computvcov$list.vcov,
        name.y = NULL, name.study = Craft2003$ID,
        y.all = obj$coefficients[,1],
        y.all.se = obj$coefficients[,2])

########################
# Pooling from subsets
########################
```

```
# o1 <- metami(dat, M = 10, vcov = "r.vcov",
#               n.name, ef.name,
#               func = "metafixed",
#               pool.seq = c(5, 10))
# pooled results from M = 5 imputed data sets
# o1$result.seq$M5$coefficients
# pooled results from M = 10 imputed data sets
# o1$result.seq$M10$coefficients
#########################################################################################
# Running random-effects and meta-regression model using packages "mixmeta" or "metaSEM"
#########################################################################################
# Restricted maximum likelihood (REML) estimator from the mixmeta package
# library(mixmeta)
# o2 <- metami(dat, M = 10, vcov = "r.vcov",
#                     n.name, ef.name,
#                     formula = as.formula(cbind(C1, C2, C3, C4, C5, C6) ~ 1),
#                     func = "mixmeta",
#                     method = "reml")
# maximum likelihood estimators from the metaSEM package
# library(metaSEM)
# o3 <- metami(dat, M = 10, vcov = "r.vcov",
#               n.name, ef.name,
#               func = "meta")

# meta-regression
# library(metaSEM)
# o4 <- metami(dat, M = 10, vcov = "r.vcov",
#                     n.name, ef.name, x.name = "p_male",
#                     func = "meta")
# library(mixmeta)
# o5 <- metami(dat, M = 20, vcov = "r.vcov",
#               n.name, ef.name, x.name = "p_male",
#               formula = as.formula(cbind(C1, C2, C3, C4, C5, C6) ~ p_male ),
#               func = "mixmeta",
#               method = "reml")
#####################################################################################
# Example: Geeganage2010 data
# Preparing input arguments for meta.mi() and fixed-effect model
#####################################################################################
# Geeganage2010.mnar <- Geeganage2010
# Geeganage2010.mnar$MD_SBP[sample(1:nrow(Geeganage2010),7)] <- NA
# r12 <- 0.71
# r13 <- 0.5
# r14 <- 0.25
# r23 <- 0.6
# r24 <- 0.16
# r34 <- 0.16
# r <- vecTosm(c(r12, r13, r14, r23, r24, r34))
# diag(r) <- 1
# mix.r <- lapply(1:nrow(Geeganage2010), function(i){r})
# o <- metami(data = Geeganage2010.mnar, M = 10, vcov = "mix.vcov",
#               ef.name = c("MD_SBP", "MD_DBP", "RD_DD", "lgOR_D"),
#               type = c("MD", "MD", "RD", "lgOR"),
#              d = c("MD_SBP", "MD_DBP", NA, NA),
#              sdt = c("sdt_SBP", "sdt_DBP", NA, NA),
#              sdc = c("sdc_SBP", "sdc_DBP", NA, NA),
#              nt = c("nt_SBP", "nt_DBP", "nt_DD", "nt_D"),
#              nc = c("nc_SBP", "nc_DBP", "nc_DD", "nc_D"),
#              st = c(NA, NA, "st_DD", "st_D"),
#              sc = c(NA, NA, "sc_DD", "sc_D"),
#              r = mix.r,
#              func = "metafixed")
```

mix.vcov *Computing Variance-Covariance Matrices for Effect Sizes of the Same or Different Types*

Description

The function `r.vcov` computes effect sizes and variance-covariance matrices between effect sizes of the same of different types. Effect sizes include mean difference (MD), standardized mean difference (SMD), log odds ratio (logOR), log risk ratio (logRR), and risk difference (RD). Formulas are in Table I of Wei et al.'s paper (2013).

Usage

```
mix.vcov(d, r, nt, nc,
         st, sc, n_rt = NA, n_rc = NA,
         sdt, sdc, type,
         name = NULL, na.impute = NA)
```

Arguments

`d` A $N \times p$ matrix or data frame with mean differences (MD) and/or standard mean differences (SMD) from the N studies. `d[i,j]` is the value from study i for outcome j. If outcome j is not MD or SMD, NA has to be imputed in column j.

`r` A N-dimensional list of $p \times p$ correlation matrices for the p outcomes from the N studies. `r[[k]][i,j]` is the correlation coefficient between outcome i and outcome j from study k.

`nt` A $N \times p$ matrix storing sample sizes in the treatment group reporting the p outcomes. `nt[i,j]` is the sample size from study i reporting outcome j.

`nc` A matrix defined in a similar way as `nt` for the control group.

`st` A $N \times p$ matrix recording number of participants with event for all outcomes (dichotomous) in treatment group. `st[i,j]` reports number of participants with event for outcome j in treatment group for study i. If outcome j is not dichotomous, NA has to be imputed in column j.

`sc` Defined in a similar way as `st` for the control group.

`n_rt` A N-dimensional list of $p \times p$ matrices storing sample sizes in the treatment group reporting pairwise outcomes in the off-diagonal elements. `n_rt[[k]][i,j]` is the sample size reporting both outcome i and outcome j from study k. Diagonal elements of these matrices are discarded. The default value is `NA`, which means that the smaller sample size reporting the corresponding two outcomes is imputed: i.e. `n_rt[[k]][i,j]=min(nt[k,i],nt[k,j])`.

`n_rc` A list defined in a similar way as `n_rt` for the control group.

`sdt` A $N \times p$ matrix storing sample standard deviations for each outcome from treatment group. `sdt[i,j]` is the sample standard deviation from study i for outcome j. If outcome j is not continuous such as MD or SMD, NA has to be imputed in the jth column.

`sdc` A matrix defined in a similar way as `sdt` for the control group.

`type` A p-dimensional vector indicating types of effect sizes. `"MD"` stands for mean difference, `"SMD"` stands for standardized mean difference, `"logOR"` stands for log odds ratio, `"logRR"` stands for log risk ratio, and `"RD"` stands for risk difference.

`name` A p-dimensional vector storing names for the effect sizes.

`na.impute` Missing values in `d` can be imputed by a numeric value, such as zero by setting `na.impute = 0`. With the default setting `na.impute = NA`, missing values are not imputed. If specifying `na.impute = "average"`, missing values are imputed by the mean of `d` that is sample-size weighted from the complete records.

Value

`ef` A $N \times p$ data frame whose columns are computed effect sizes according to the input argument `"type"`. SMD will be converted to Hedges's g (Wei and Higgins, 2013).

`list.vcov` A N-dimensional list of $p(p+1)/2 \times p(p+1)/2$ matrices of computed variance-covariance matrices.

`matrix.vcov` A $N \times p(p+1)/2$ matrix whose rows are computed variance-covariance vectors.

Author(s)

Min Lu

References

Lu, M. (2023). Computing within-study covariances, data visualization, and missing data solutions for multivariate meta-analysis with metavcov. *Frontiers in Psychology*, 14:1185012.

Wei, Y., & Higgins, J. (2013). Estimating within study covariances in multivariate meta-analysis with multiple outcomes. *Statistics in Medicine, 32*(7), 119-1205.

Examples

```
#########################################################################
# Example: Geeganage2010 data
# Preparing covariances for a multivariate meta-analysis
# Choose variable SBP, DBP, DD, D with effect sizes "MD","MD","RD","lgOR"
#########################################################################
data(Geeganage2010)
## set the correlation coefficients list r
r12 <- 0.71
r13 <- 0.5
r14 <- 0.25
r23 <- 0.6
r24 <- 0.16
r34 <- 0.16
r <- vecTosm(c(r12, r13, r14, r23, r24, r34))
diag(r) <- 1
mix.r <- lapply(1:nrow(Geeganage2010), function(i){r})
attach(Geeganage2010)
## compute variance co-variances
computvcov <- mix.vcov(type = c("MD", "MD", "RD", "lgOR"),
                       d = cbind(MD_SBP, MD_DBP, NA, NA),
                       sdt = cbind(sdt_SBP, sdt_DBP, NA, NA),
                       sdc = cbind(sdc_SBP, sdc_DBP, NA, NA),
                       nt = cbind(nt_SBP, nt_DBP, nt_DD, nt_D),
                       nc = cbind(nc_SBP, nc_DBP, nc_DD, nc_D),
                       st = cbind(NA, NA, st_DD, st_D),
                       sc = cbind(NA, NA, sc_DD, sc_D),
                       r = mix.r,
                       name = c("MD.SBP", "MD.DBP", "RD.DD", "lgOR.D"))
# save different effect sizes in y
y <- computvcov$ef
head(y)
# save variances and covariances of all the effect sizes in a matrix S
S <- computvcov$matrix.vcov
S[1, ]
## fixed-effect model
MMA_FE <- summary(metafixed(y = y, Slist = computvcov$list.vcov))
#######################################################################
# Running random-effects model using package "mixmeta" or "metaSEM"
#######################################################################
# Restricted maximum likelihood (REML) estimator from the mixmeta package
# library(mixmeta)
# mvmeta_RE <- summary(mixmeta(cbind(MD.SBP, MD.DBP, RD.DD, lgOR.D) ~ 1,
#                             S = S, data = y, method = "reml"))
# mvmeta_RE

# maximum likelihood estimators from the metaSEM package
# library(metaSEM)
# metaSEM_RE <- summary(meta(y = y, v = S))
# metaSEM_RE
##############################################################
# Plotting the result:
##############################################################
obj <- MMA_FE
# obj <- mvmeta_RE
# obj <- metaSEM_RE
# pdf("CI.pdf", width = 4, height = 7)
```

```
plotCI(y = computvcov$ef, v = computvcov$list.vcov,
        name.y = NULL, name.study = Geeganage2010$studyID,
        y.all = obj$coefficients[,1],
        y.all.se = obj$coefficients[,2],
        hline = c(0, 0, 0, 1))
# dev.off()
```

plotCI *Plot Confidence Intervals for a Meta-Analysis*

Description

The function `plot.CI` generates confidence interval figures for effect sizes from each study and the estimated effect sizes across studies.

Usage

```
plotCI(y, v,
        name.y = NULL,
        name.study = NULL,
        y.all, y.all.se,
        hline = 0,
        up.bound = Inf, low.bound = -Inf,
        return.data = FALSE)
```

Arguments

`y` A $N \times p$ matrix or data frame that stores effect sizes from N primary studies. Usually the output value `ef` produced by `r.vcov` for correlation coefficients or `mix.vcov` for other types of effect sizes.

`v` A N-dimensional list of $p \times p$ matrices that stores within-study (co)variance matrices of the estimated effect sizes for each one of the N studies. Usually the output value `list.vcov` produced by `r.vcov` for correlation coefficients or `mix.vcov` for other types of effect sizes.

`name.y` A p-dimensional vector that stores names for the effect sizes in `y`. By default, it is the columname of `y`.

`name.study` A N-dimensional vector that stores names for the primary in `y`. By default, it is the row name of `y`.

`y.all` A p-dimensional vector that stores the estimated effect sizes across studies.

`y.all.se` A p-dimensional vector that stores the standard errors for the estimated effect sizes in `y.all`.

`hline` A p-dimensional vector that specifies the position of the dash line in the figures to compare the coefficients across studies. If its length is one instead of p, this number will be adopted for all the p effect sizes.

`up.bound` A p-dimensional vector that specifies the upper bound in the figures. If its length is one instead of p, this number will be adopted for all the p effect sizes.

`low.bound` A p-dimensional vector that specifies the lower bound in the figures. If its length is one instead of p, this number will be adopted for all the p effect sizes.

`return.data` Should the data for the confidence interval plots be returned?

Details

The difference between a forest plot and a confidence interval plot is that a forest plot requires a symbol on each confidence interval that is proportional to the weight for each study. Because the weighting mechanism in multivariate meta-analysis is too complex to be visualized, such a propositional symbol is omitted for multivariate meta-analysis.

Author(s)

Min Lu

References

Lu, M. (2023). Computing within-study covariances, data visualization, and missing data solutions for multivariate meta-analysis with metavcov. *Frontiers in Psychology*, 14:1185012.

Examples

```
######################################################
# Example: Craft2003 data
######################################################
data(Craft2003)
computvcov <- r.vcov(n = Craft2003$N,
                     corflat = subset(Craft2003, select = C1:C6),
                     method = "average")
y <- computvcov$ef
Slist <- computvcov$list.vcov
MMA_FE <- summary(metafixed(y = y, Slist = Slist))
obj <- MMA_FE
# pdf("CI.pdf", width = 4, height = 7)
plotCI(y = computvcov$ef, v = computvcov$list.vcov,
        name.y = NULL, name.study = Craft2003$ID,
        y.all = obj$coefficients[,1],
        y.all.se = obj$coefficients[,2])
# dev.off()

######################################################
# Substitute obj for Random-effect model
######################################################
# library(mvmeta)
# S <- computvcov$matrix.vcov
# MMA_RE <- summary(mvmeta(cbind(C1, C2, C3, C4, C5, C6),
#                            S = S, data = y, method = "reml"))
# obj <- MMA_RE
```

r.vcov *Computing Variance-Covariance Matrices for Correlation Coefficients*

Description

The function r.vcov computes variance-covariance matrix for multivariate meta-analysis when the effect size is measured by correlation coefficient.

Usage

```
r.vcov(n, corflat, zscore = FALSE, name = NULL, method = "average", na.impute = NA)
```

Arguments

n A N-dimensional vector containing sample sizes from N studies.

corflat A $N \times p$ matrix or data frame storing p correlation coefficients from each of the N studies.

zscore Whether the correlation coefficients in corflat are already transformed into Fisher's z scores.

name A p-dimensional vector containing names for the p correlation coefficients.

method Method "average" computes variance and covariances with mean correlation coefficients that are sample-size weighted from all the N studies (missing values are omitted); method "each" computes variance and covariances with each of the corresponding correlation coefficients.

na.impute Missing values can be imputed by a numeric value, such as zero by setting na.impute = 0. With the default setting na.impute = NA, missing values are not imputed. If specifying na.impute = "average", missing values are imputed by mean correlation coefficients that are sample-size weighted from the complete records.

Details

How to arrange correlation coefficients of each study from matrix to vector is in Cooper et al book page 385 to 386. Details for average method are in book of Cooper et al page 388. Let r_{ist} denote the sample correlation coefficient that describes the relationship between variables s and t in study i. We can calculate its variance as $var(r_{ist}) = (1 - \rho_{ist}^2)^2/n_i$, and the covariance between two correlation coefficients is $cov(r_{ist}, r_{iuv}) = [.5\rho_{ist}\rho_{iuv}(\rho_{isu}^2 + \rho_{isv}^2 + \rho_{itu}^2 + \rho_{itv}^2) + \rho_{isu}\rho_{itv} + \rho_{isv}\rho_{itu} - (\rho_{ist}\rho_{isu}\rho_{isv} + \rho_{its}\rho_{itu}\rho_{itv} + \rho_{ius}\rho_{iut}\rho_{iuv} + \rho_{ivs}\rho_{ivt}\rho_{ivu})]/n_i$, where $\rho_{i..}$ represents the population value. In practice, $\rho_{i..}$ can be substituted by the observed sample correlation or sample-size weighted mean correlation coefficients from all studies.

Value

`r` A $N \times p$ data frame that contains the input argument `corflat` with column names and imputed values according to the input argument `na.impute`. If the input argument `zscore=TRUE`, `r` is transformed from Fisher's z score in `corflat`.

`list.rvcov` A N-dimensional list of $p(p+1)/2 \times p(p+1)/2$ matrices of computed variance-covariance matrices.

`matrix.rvcov` A $N \times p(p+1)/2$ matrix whose rows are computed variance-covariance vectors.

`ef` A $N \times p$ data frame that contains Fisher's z transformed correlation coefficients from the input argument `corflat` with column names and imputed values according to the input argument `na.impute`.

`list.vcov` A list in the same format of `list.rvcov` for Fisher's z transformed correlation coefficients.

`matrix.vcov` A matrix `matrix.rvcov` for Fisher's z transformed correlation coefficients.

Author(s)

Min Lu

References

Lu, M. (2023). Computing within-study covariances, data visualization, and missing data solutions for multivariate meta-analysis with metavcov. *Frontiers in Psychology*, 14:1185012.

Examples

```
##############################################################
# A simple example:
# Checking the example in Harris Cooper et al.'s book page 388
##############################################################
r <- matrix(c(-0.074, -0.127, 0.324, 0.523, -0.416, -0.414), 1)
n <- 142
computvcov <- r.vcov(n = n, corflat = r,
                     name = paste("C", c("st", "su", "sv", "tu", "tv", "uv"), sep = ""),
                     method = "each")
round(computvcov$list.rvcov[[1]], 4)
round(computvcov$ef, 4)
round(computvcov$list.vcov[[1]], 4)
round(computvcov$matrix.vcov, 4)
######################################################
# Example: Craft2003 data
# Preparing covariances for multivariate meta-analysis
######################################################
data(Craft2003)
# extract correlation from the dataset (craft)
corflat <- subset(Craft2003, select=C1:C6)
# transform correlations to z and compute variance-covariance matrix.
computvcov <- r.vcov(n = Craft2003$N, corflat = corflat, method = "average")
# name transformed z scores as y
y <- computvcov$ef
```

```
# name variance-covariance matrix of trnasformed z scores as S
S <- computvcov$matrix.vcov
S[1, ]
## fixed-effect model
MMA_FE <- summary(metafixed(y = y, Slist = computvcov$list.vcov))
#######################################################################
# Running random-effects model using package "mixmeta" or "metaSEM"
#######################################################################
# Restricted maximum likelihood (REML) estimator from the mixmeta package
#library(mixmeta)
#mvmeta_RE <- summary(mixmeta(cbind(C1, C2, C3, C4, C5, C6) ~ 1,
#                           S = S, data = y, method = "reml"))
#mvmeta_RE

# maximum likelihood estimators from the metaSEM package
# library(metaSEM)
# metaSEM_RE <- summary(meta(y = y, v = S))
# metaSEM_RE
##############################################################
# Plotting the result:
##############################################################
obj <- MMA_FE
# obj <- mvmeta_RE
# obj <- metaSEM_RE
# pdf("CI.pdf", width = 4, height = 7)
plotCI(y = computvcov$ef, v = computvcov$list.vcov,
        name.y = NULL, name.study = Craft2003$ID,
        y.all = obj$coefficients[,1],
        y.all.se = obj$coefficients[,2])
# dev.off()
```

rd.vcov *Computing Variance-Covariance Matrices for Risk Differences*

Description

The function `lgOR.vcov` computes effect sizes and variance-covariance matrix for multivariate meta-analysis when the effect sizes of interest are all measured by risk difference. See `mix.vcov` for effect sizes of the same or different types.

Usage

```
rd.vcov(r, nt, nc, st, sc, n_rt = NA, n_rc = NA)
```

Arguments

`r` A N-dimensional list of $p \times p$ correlation matrices for the p outcomes from the N studies. `r[[k]][i,j]` is the correlation coefficient between outcome i and outcome j from study k.

`nt` A $N \times p$ matrix storing sample sizes in the treatment group reporting the p outcomes. `nt[i,j]` is the sample size from study i reporting outcome j.

`nc` A matrix defined in a similar way as `nt` for the control group.

`st` A $N \times p$ matrix recording number of participants with event for all outcomes (dichotomous) in treatment group. `st[i,j]` reports number of participants with event for outcome j in treatment group for study i. If outcome j is not dichotomous, NA has to be imputed in column j.

`sc` Defined in a similar way as `st` for the control group.

`n_rt` A N-dimensional list of $p \times p$ matrices storing sample sizes in the treatment group reporting pairwise outcomes in the off-diagonal elements. `n_rt[[k]][i,j]` is the sample size reporting both outcome i and outcome j from study k. Diagonal elements of these matrices are discarded. The default value is `NA`, which means that the smaller sample size reporting the corresponding two outcomes is imputed: i.e. `n_rt[[k]][i,j]=min(nt[k,i],nt[k,j])`.

`n_rc` A list defined in a similar way as `n_rt` for the control group.

Value

`ef` A $N \times p$ data frame whose columns are computed risk differences.

`list.vcov` A N-dimensional list of $p(p+1)/2 \times p(p+1)/2$ matrices of computed variance-covariance matrices.

`matrix.vcov` A $N \times p(p+1)/2$ matrix whose rows are computed variance-covariance vectors.

Author(s)

Min Lu

References

Lu, M. (2023). Computing within-study covariances, data visualization, and missing data solutions for multivariate meta-analysis with metavcov. *Frontiers in Psychology*, 14:1185012.

Examples

```
###########################################################################
# Example: Geeganage2010 data
# Preparing risk differences and covariances for multivariate meta-analysis
###########################################################################
data(Geeganage2010)
## set the correlation coefficients list r
r12 <- 0.71
r.Gee <- lapply(1:nrow(Geeganage2010), function(i){matrix(c(1, r12, r12, 1), 2, 2)})


computvcov <- rd.vcov(nt = subset(Geeganage2010, select = c(nt_DD, nt_D)),
                      nc = subset(Geeganage2010, select = c(nc_DD, nc_D)),
                      st = subset(Geeganage2010, select = c(st_DD, st_D)),
                      sc = subset(Geeganage2010, select = c(sc_DD, sc_D)),
                      r = r.Gee)
# name computed relative risk as y
y <- computvcov$ef
colnames(y) <- c("rd.DD", "rd.D")
# name variance-covariance matrix of trnasformed z scores as covars
S <- computvcov$matrix.vcov
## fixed-effect model
MMA_FE <- summary(metafixed(y = y, Slist = computvcov$list.vcov))
#######################################################################
# Running random-effects model using package "mixmeta" or "metaSEM"
#######################################################################
#library(mixmeta)
#mvmeta_RE <- summary(mixmeta(cbind(rd.DD, rd.D)~1,
#                          S = S, data = as.data.frame(y),
#                          method = "reml"))
#mvmeta_RE

# maximum likelihood estimators from the metaSEM package
# library(metaSEM)
# metaSEM_RE <- summary(meta(y = y, v = S))
# metaSEM_RE
##############################################################
# Plotting the result:
##############################################################
obj <- MMA_FE
# obj <- mvmeta_RE
# obj <- metaSEM_RE
# pdf("CI.pdf", width = 4, height = 7)
plotCI(y = computvcov$ef, v = computvcov$list.vcov,
        name.y = c("rd.DD", "rd.D"),
        name.study = Geeganage2010$studyID,
        y.all = obj$coefficients[,1],
        y.all.se = obj$coefficients[,2])
# dev.off()
```

smd.vcov *Computing Variance-Covariance Matrices for Standardized Mean Differences*

Description

The function lgOR.vcov computes effect sizes and variance-covariance matrix for multivariate meta-analysis when the effect sizes of interest are all measured by standardized mean difference. See mix.vcov for effect sizes of the same or different types.

Usage

```
smd.vcov(nt, nc, d, r, n_rt = NA, n_rc = NA, name = NULL)
```

Arguments

`nt` A $N \times p$ matrix storing sample sizes in the treatment group reporting the p outcomes. `nt[i,j]` is the sample size from study i reporting outcome j.

`nc` A matrix defined in a similar way as `nt` for the control group.

`d` A $N \times p$ matrix or data frame with standard mean differences (SMD) from the N studies. `d[i,j]` is the value from study i for outcome j.

`r` A N-dimensional list of $p \times p$ correlation matrices for the p outcomes from the N studies. `r[[k]][i,j]` is the correlation coefficient between outcome i and outcome j from study k.

`n_rt` A N-dimensional list of $p \times p$ matrices storing sample sizes in the treatment group reporting pairwise outcomes in the off-diagonal elements. `n_rt[[k]][i,j]` is the sample size reporting both outcome i and outcome j from study k. Diagonal elements of these matrices are discarded. The default value is `NA`, which means that the smaller sample size reporting the corresponding two outcomes is imputed: i.e. `n_rt[[k]][i,j]=min(nt[k,i],nt[k,j])`.

`n_rc` A list defined in a similar way as `n_rt` for the control group.

`name` Names for the outcomes.

Value

`ef` A $N \times p$ data frame that transforms the input argument `d` into Hedges's g (Wei and Higgins, 2013).

`list.vcov` A N-dimensional list of $p(p+1)/2 \times p(p+1)/2$ variance-covariance matrices for Hedges's g (Wei and Higgins, 2013).

`matrix.vcov` A $N \times p(p+1)/2$ whose rows are computed variance-covariance vectors for Hedges's g (Wei and Higgins, 2013).

`list.dvcov` A N-dimensional list of $p(p+1)/2 \times p(p+1)/2$ variance-covariance matrices for SMD (Olkin and Gleser, 2009).

`matrix.dvcov` A $N \times p(p+1)/2$ matrix whose rows are computed variance-covariance vectors for SMD (Olkin and Gleser, 2009).

Author(s)

Min Lu

References

Lu, M. (2023). Computing within-study covariances, data visualization, and missing data solutions for multivariate meta-analysis with metavcov. *Frontiers in Psychology*, 14:1185012.

Examples

```
#####################################################
# Example: Geeganage2010 data
# Preparing covarianceS for multivariate meta-analysis
#####################################################
```

```
data(Geeganage2010)
## set the correlation coefficients list r
r12 <- 0.71
r.Gee <- lapply(1:nrow(Geeganage2010), function(i){matrix(c(1, r12, r12, 1), 2, 2)})

computvcov <- smd.vcov(nt = subset(Geeganage2010, select = c(nt_SBP, nt_DBP)),
               nc = subset(Geeganage2010, select = c(nc_SBP, nc_DBP)),
               d = subset(Geeganage2010, select = c(SMD_SBP, SMD_DBP)), r = r.Gee,
               name = c(”SMD_SBP”, ”SMD_DBP”))
# name variance-covariance matrix as S
S <- computvcov$matrix.vcov
## fixed-effect model
y <- computvcov$ef
MMA_FE <- summary(metafixed(y = y, Slist = computvcov$list.vcov))
#######################################################################
# Running random-effects model using package ”mixmeta” or ”metaSEM”
#######################################################################
# Restricted maximum likelihood (REML) estimator from the mixmeta package
#library(mixmeta)
#mvmeta_RE <- summary(mixmeta(cbind(SMD_SBP, SMD_DBP)~1,
#                     S = S,
#                     data = y,
#                     method = ”reml”))
#mvmeta_RE

# maximum likelihood estimators from the metaSEM package
# library(metaSEM)
# metaSEM_RE <- summary(meta(y = y, v = S))
# metaSEM_RE
##############################################################
# Plotting the result:
##############################################################
obj <- MMA_FE
# obj <- mvmeta_RE
# obj <- metaSEM_RE
# pdf(”CI.pdf”, width = 4, height = 7)
plotCI(y = computvcov$ef, v = computvcov$list.vcov,
        name.y = NULL, name.study = Geeganage2010$studyID,
        y.all = obj$coefficients[,1],
        y.all.se = obj$coefficients[,2])
# dev.off()
```

smd_lgor *Computing Covariance between Standardized Mean Difference and Log Odds Ratio*

Description

The function smd_lgor computes covariance between standardized mean difference and log odds ratio. See mix.vcov for effect sizes of the same or different types.

Usage

```
smd_lgor(d, r, n1c, n2c, n1t, n2t,
         n12c = min(n1c, n2c), n12t = min(n1t, n2t),
         s2c, s2t, f2c, f2t, sd1c, sd1t)
```

Arguments

`d` Standardized mean difference for outcome 1.

`r` Correlation coefficient of the two outcomes.

`n1c` Number of participants reporting outcome 1 in the control group.

`n2c` Number of participants reporting outcome 2 in the control group.

`n1t` Number of participants reporting outcome 1 in the treatment group.

`n2t` Number of participants reporting outcome 2 in the treatment group.

`n12c` Number of participants reporting both outcome 1 and outcome 2 in the control group. By default, it is equal to the smaller number between `n1c` and `n2c`.

`n12t` Number defined in a similar way as `n12c` for the treatment group.

`s2c` Number of participants with event for outcome 2 (dichotomous) in the control group.

`s2t` Defined in a similar way as `s2c` for the treatment group.

`f2c` Number of participants without event for outcome 2 (dichotomous) in the control group.

`f2t` Defined in a similar way as `f2c` for the treatment group.

`sd1c` Sample standard deviation of outcome 1 for the control group.

`sd1t` Defined in a similar way as `sd1c` for the treatment group.

Value

`g` Computed Hedge's `g` from the input argument `d` for outcome 1.

`lgor` Computed log odds ratio for outcome 2.

`v` Computed covariance.

Author(s)

Min Lu

References

Lu, M. (2023). Computing within-study covariances, data visualization, and missing data solutions for multivariate meta-analysis with metavcov. *Frontiers in Psychology*, 14:1185012.

Examples

```
## simple example
smd_lgor(d = 1, r = 0.71, n1c = 34, n2c = 35, n1t = 25, n2t = 32,
         s2c = 5, s2t = 8, f2c = 30, f2t = 24, sd1t = 0.4, sd1c = 8)
## calculate covariances for variable SBP and DD in Geeganage2010 data
attach(Geeganage2010)
SBP_DD <- unlist(lapply(1:nrow(Geeganage2010), function(i){smd_lgor(d = SMD_SBP, r = 0.71,
                n1c = nc_SBP[i], n2c = nc_DD[i], n1t = nt_SBP[i], n2t = nt_DD[i],
                sd1t = sdt_SBP[i], s2t = st_DD[i], sd1c = sdc_SBP[i], s2c = sc_DD[i],
                f2c = nc_DD[i] - sc_DD[i], f2t = nt_DD[i] - st_DD[i])$v}))
SBP_DD
## the function mix.vcov() should be used for dataset
```

---

smd_lgrr — *Computing Covariance between Standardized Mean Difference and Log Risk Ratio*

Description

The function smd_lgrr computes covariance between standardized mean difference and log risk ratio. See mix.vcov for effect sizes of the same or different types.

Usage

```
smd_lgrr(d, r, n1c, n2c, n1t, n2t,
          n12c = min(n1c, n2c), n12t = min(n1t, n2t),
          s2c, s2t, f2c, f2t, sd1c, sd1t)
```

Arguments

d	Standardized mean difference for outcome 1.
r	Correlation coefficient of the two outcomes.
n1c	Number of participants reporting outcome 1 in the control group.
n2c	Number of participants reporting outcome 2 in the control group.
n1t	Number of participants reporting outcome 1 in the treatment group.
n2t	Number of participants reporting outcome 2 in the treatment group.
n12c	Number of participants reporting both outcome 1 and outcome 2 in the control group. By default, it is equal to the smaller number between n1c and n2c.
n12t	Number defined in a similar way as n12c for the treatment group.
s2c	Number of participants with event for outcome 2 (dichotomous) in the control group.
s2t	Defined in a similar way as s2c for the treatment group.
f2c	Number of participants without event for outcome 2 (dichotomous) in the control group.
f2t	Defined in a similar way as f2c for the treatment group.
sd1c	Sample standard deviation of outcome 1 for the control group.
sd1t	Defined in a similar way as sd1c for the treatment group.

Value

`g` Computed Hedge's g from the input argument `d` for outcome 1.

`lgrr` Computed log risk ratio for outcome 2.

`v` Computed covariance.

Author(s)

Min Lu

References

Lu, M. (2023). Computing within-study covariances, data visualization, and missing data solutions for multivariate meta-analysis with metavcov. *Frontiers in Psychology*, 14:1185012.

Examples

```
## simple example
smd_lgrr(d = 1, r = 0.3, n1c = 34, n2c = 35, n1t = 25, n2t = 32,
         s2c = 5, s2t = 8, f2c = 30, f2t = 24, sd1t = 0.4, sd1c = 8)
## calculate covariances for variable SBP and DD in Geeganage2010 data
attach(Geeganage2010)
SBP_DD <- unlist(lapply(1:nrow(Geeganage2010), function(i){smd_lgrr(d = SMD_SBP, r = 0.3,
                 n1c = nc_SBP[i], n2c = nc_DD[i], n1t = nt_SBP[i], n2t = nt_DD[i],
                 sd1t = sdt_SBP[i], s2t = st_DD[i], sd1c = sdc_SBP[i], s2c = sc_DD[i],
                 f2c = nc_DD[i] - sc_DD[i], f2t = nt_DD[i] - st_DD[i])$v}))
SBP_DD
## the function mix.vcov() should be used for dataset
```

`smd_rd` *Computing Covariance between Standardized Mean Difference and Risk Difference*

Description

The function `smd_rd` computes covariance between standardized mean difference and risk difference. See `mix.vcov` for effect sizes of the same or different types.

Usage

```
smd_rd(d, r, n1c, n2c, n1t, n2t,
       n12c = min(n1c, n2c), n12t = min(n1t, n2t),
       s2c, s2t, f2c, f2t, sd1c, sd1t)
```

Arguments

d	Standardized mean difference for outcome 1.
r	Correlation coefficient of the two outcomes.
n1c	Number of participants reporting outcome 1 in the control group.
n2c	Number of participants reporting outcome 2 in the control group.
n1t	Number of participants reporting outcome 1 in the treatment group.
n2t	Number of participants reporting outcome 2 in the treatment group.
n12c	Number of participants reporting both outcome 1 and outcome 2 in the control group. By default, it is equal to the smaller number between n1c and n2c.
n12t	Number defined in a similar way as n12c for the treatment group.
s2c	Number of participants with event for outcome 2 (dichotomous) in the control group.
s2t	Defined in a similar way as s2c for the treatment group.
f2c	Number of participants without event for outcome 2 (dichotomous) in the control group.
f2t	Defined in a similar way as f2c for the treatment group.
sd1c	Sample standard deviation of outcome 1 for the control group.
sd1t	Defined in a similar way as sd1c for the treatment group.

Value

g	Computed Hedge's g from the input argument d for outcome 1.
rd	Computed risk difference for outcome 1.
v	Computed covariance.

Author(s)

Min Lu

References

Lu, M. (2023). Computing within-study covariances, data visualization, and missing data solutions for multivariate meta-analysis with metavcov. *Frontiers in Psychology*, 14:1185012.

Examples

```
## simple example
smd_rd(d = 1, r = 0.71, n1c = 34, n2c = 35, n1t = 25, n2t = 32,
       s2c = 5, s2t = 8, f2c = 30, f2t = 24, sd1t = 0.4, sd1c = 8)
## calculate covariances for variable SBP and DD in Geeganage2010 data
attach(Geeganage2010)
SBP_DD <- unlist(lapply(1:nrow(Geeganage2010), function(i){smd_rd(d = SMD_SBP, r = 0.71,
                n1c = nc_SBP[i], n2c = nc_DD[i], n1t = nt_SBP[i], n2t = nt_DD[i],
                sd1t = sdt_SBP[i], s2t = st_DD[i], sd1c = sdc_SBP[i], s2c = sc_DD[i],
                f2c = nc_DD[i] - sc_DD[i], f2t = nt_DD[i] - st_DD[i])}))
SBP_DD
## the function mix.vcov() should be used for dataset
```

Index