Package: EValue (via r-universe)

Type Package

Title Sensitivity Analyses for Unmeasured Confounding and Other Biases
in Observational Studies and Meta-Analyses

Version 4.1.3

Description Conducts sensitivity analyses for unmeasured confounding,
selection bias, and measurement error (individually or in
combination; VanderWeele & Ding (2017) <doi:10.7326/M16-2607>;
Smith & VanderWeele (2019) <doi:10.1097/EDE.0000000000001032>;
VanderWeele & Li (2019) <doi:10.1093/aje/kwz133>; Smith &
VanderWeele (2021) <arXiv:2005.02908>). Also conducts
sensitivity analyses for unmeasured confounding in
meta-analyses (Mathur & VanderWeele (2020a)
<doi:10.1080/01621459.2018.1529598>; Mathur & VanderWeele
(2020b) <doi:10.1097/EDE.0000000000001180>) and for additive
measures of effect modification (Mathur et al., under review).

LazyData true

License GPL-2

Imports stats, graphics, ggplot2 (>= 2.2.1), metafor, metadat,
methods, boot, MetaUtility, dplyr

RoxygenNote 7.1.1

Encoding UTF-8

Suggests testthat, knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation no

Author Maya B. Mathur [cre, aut], Louisa H. Smith [aut], Peng Ding
[aut], Tyler J. VanderWeele [aut]

Maintainer Maya B. Mathur <mmathur@stanford.edu>

Date/Publication 2021-10-28 04:10:02 UTC

Repository https://mayamathur.r-universe.dev

RemoteUrl https://github.com/cran/EValue

RemoteRef HEAD

RemoteSha 739353c4c943398c47cda798275f2c95ace6d6c0

Contents

---

bias_plot   *Plot bias factor as function of confounding relative risks*

Description

Plots the bias factor required to explain away a provided relative risk.

Usage

```
bias_plot(RR, xmax)
```

Arguments

RR	The relative risk
xmax	Upper limit of x-axis.

Examples

```
# recreate the plot in VanderWeele and Ding (2017)
bias_plot(RR=3.9, xmax=20)
```

---

confounded_meta	*Sensitivity analysis for unmeasured confounding in meta-analyses*

Description

This function implements the sensitivity analyses of Mathur & VanderWeele (2020a, 2020b). It computes point estimates, standard errors, and confidence intervals for (1) Prop, the proportion of studies with true causal effect sizes above or below a chosen threshold q as a function of the bias parameters; (2) the minimum bias factor on the relative risk scale (Tmin) required to reduce to less than r the proportion of studies with true causal effect sizes more extreme than q; and (3) the counterpart to (2) in which bias is parameterized as the minimum relative risk for both confounding associations (Gmin).

Usage

```
confounded_meta(
  method = "calibrated",
  q,
  r = NA,
  tail = NA,
  CI.level = 0.95,
  give.CI = TRUE,
  R = 1000,
  muB = NA,
  muB.toward.null = FALSE,
  dat = NA,
  yi.name = NA,
  vi.name = NA,
  sigB = NA,
  yr = NA,
  vyr = NA,
  t2 = NA,
  vt2 = NA,
  ...
)
```

Arguments

method	"calibrated" or "parametric". See Details.
q	True causal effect size chosen as the threshold for a meaningfully large effect.
r	For Tmin and Gmin, value to which the proportion of meaningfully strong effect sizes is to be reduced.

tail	"above" for the proportion of effects above q; "below" for the proportion of effects below q. By default, is set to "above" if the pooled point estimate (method = "parametric") or median of the calibrated estimates (method = "calibrated") is above 1 on the relative risk scale and is set to "below" otherwise.
CI.level	Confidence level as a proportion (e.g., 0.95).
give.CI	Logical. If TRUE, confidence intervals are provided. Otherwise, only point estimates are provided.
R	Number of bootstrap iterates for confidence interval estimation. Only used if method = "calibrated" and give.CI = TRUE.
muB	Mean bias factor on the log scale across studies (greater than 0). When considering bias that is of homogeneous strength across studies (i.e., method = "calibrated" or method = "parametric" with sigB = 0), muB represents the log-bias factor in each study. If muB is not specified, then only Tmin and Gmin will be returned, not Prop.
muB.toward.null	
	Whether you want to consider bias that has on average shifted studies' point estimates away from the null (FALSE; the default) or that has on average shifted studies' point estimates toward the null (TRUE). See Details.
dat	Dataframe containing studies' point estimates and variances. Only used if method = "calibrated".
yi.name	Name of variable in dat containing studies' point estimates on the log-relative risk scale. Only used if method = "calibrated".
vi.name	Name of variable in dat containing studies' variance estimates. Only used if method = "calibrated".
sigB	Standard deviation of log bias factor across studies. Only used if method = "parametric".
yr	Pooled point estimate (on log-relative risk scale) from confounded meta-analysis. Only used if method = "parametric".
vyr	Estimated variance of pooled point estimate from confounded meta-analysis. Only used if method = "parametric".
t2	Estimated heterogeneity (τ^2) from confounded meta-analysis. Only used if method = "parametric".
vt2	Estimated variance of τ^2 from confounded meta-analysis. Only used if method = "parametric".
...	Additional arguments passed to confounded_meta.

Details

Specifying the sensitivity parameters on the bias:

By convention, the average log-bias factor, muB, is taken to be greater than 0 (Mathur & Vander-Weele, 2020a; Ding & VanderWeele, 2017). Confounding can operate on average either away from or toward the null, a choice specified via muB.toward.null. The most common choice for sensitivity analysis is to consider bias that operates on average away from the null, which is confounded_meta's default. In such an analysis, correcting for the bias involves shifting studies'

estimates back toward the null by muB (i.e., if yr > 0, the estimates will be corrected downward; if yr < 0, they will be corrected upward). Alternatively, to consider bias that operates on average away from the null, you would still specify muB > 0 but would also specify muB.toward.null = TRUE. For detailed guidance on choosing the sensitivity parameters muB and sigB, see Section 5 of Mathur & VanderWeele (2020a).

Specifying the threshold q**:**
For detailed guidance on choosing the threshold q, see the Supplement of Mathur & VanderWeele (2020a).

Specifying the estimation method:
By default, confounded_meta performs estimation using a **calibrated method** (Mathur & VanderWeele, 2020b) that extends work by Wang et al. (2019). This method makes no assumptions about the distribution of population effects and performs well in meta-analyses with as few as 10 studies, and performs well even when the proportion being estimated is close to 0 or 1. However, it only accommodates bias whose strength is the same in all studies (homogeneous bias). When using this method, the following arguments need to be specified:

- q
- r (if you want to estimate Tmin and Gmin)
- muB
- dat
- yi.name
- vi.name

The **parametric method** assumes that the population effects are approximately normal and that the number of studies is large. Parametric confidence intervals should only be used when the proportion estimate is between 0.15 and 0.85 (and confounded_meta will issue a warning otherwise). Unlike the calibrated method, the parametric method can accommodate bias that is heterogeneous across studies (specifically, bias that is log-normal across studies). When using this method, the following arguments need to be specified:

- q
- r (if you want to estimate Tmin and Gmin)
- muB
- sigB
- yr
- vyr (if you want confidence intervals)
- t2
- vt2 (if you want confidence intervals)

Effect size measures other than log-relative risks:
If your meta-analysis uses effect sizes other than log-relative risks, you should first approximately convert them to log-relative risks, for example via convert_measures() and then pass the converted point estimates or meta-analysis estimates to confounded_meta.

Interpreting Tmin **and** Gmin**:**
Tmin is defined as the minimum average bias factor on the relative risk scale that would be required to reduce to less than r the proportion of studies with true causal effect sizes stronger than the

threshold q, assuming that the bias factors are log-normal across studies with standard deviation sigB. Gmin is defined as the minimum confounding strength on the relative risk scale – that is, the relative risk relating unmeasured confounder(s) to both the exposure and the outcome – on average among the meta-analyzed studies, that would be required to reduce to less than r the proportion of studies with true causal effect sizes stronger than the threshold q, again assuming that bias factors are log-normal across studies with standard deviation sigB. Gmin is a one-to-one transformation of Tmin given by $Gmin = Tmin + \sqrt{Tmin * (Tmin - 1)}$. If the estimated proportion of meaningfully strong effect sizes is already less than r even without the introduction of any bias, Tmin and Gmin will be set to 1. (These definitions of Tmin and Gmin are generalizations of those given in Mathur & VanderWeele, 2020a, who defined these quantities in terms of bias that is homogeneous across studies. You can conduct analyses with homogeneous bias by setting sigB = 0.)

The direction of bias represented by Tmin and Gmin is dependent on the argument tail: when tail = "above", these metrics consider bias that had operated to *increase* studies' point estimates, and when tail = "below", these metrics consider bias that had operated to *decrease* studies' point estimates. Such bias could operate toward or away from the null depending on whether the pooled point estimate yr happens to fall above or below the null. As such, the direction of bias represented by Tmin and Gmin may or may not match that specified by the argument muB.toward.null (which is used only for estimation of Prop).

When these methods should be used:

These methods perform well only in meta-analyses with at least 10 studies; we do not recommend reporting them in smaller meta-analyses. Additionally, it only makes sense to consider proportions of effects stronger than a threshold when the heterogeneity estimate t2 is greater than 0. For meta-analyses with fewer than 10 studies or with a heterogeneity estimate of 0, you can simply report E-values for the point estimate via [evalue()](VanderWeele & Ding, 2017; see Mathur & VanderWeele (2020a), Section 7.2 for interpretation in the meta-analysis context).

References

Mathur MB & VanderWeele TJ (2020a). Sensitivity analysis for unmeasured confounding in meta-analyses. *Journal of the American Statistical Association*.

Mathur MB & VanderWeele TJ (2020b). Robust metrics and sensitivity analyses for meta-analyses of heterogeneous effects. *Epidemiology*.

Mathur MB & VanderWeele TJ (2019). New statistical metrics for meta-analyses of heterogeneous effects. *Statistics in Medicine*.

Ding P & VanderWeele TJ (2016). Sensitivity analysis without assumptions. *Epidemiology*.

VanderWeele TJ & Ding P (2017). Introducing the E-value. *Annals of Internal Medicine*.

Wang C-C & Lee W-C (2019). A simple method to estimate prediction intervals and predictive distributions: Summarizing meta-analyses beyond means and confidence intervals. *Research Synthesis Methods*.

Examples

```
##### Using Calibrated Method #####
d = metafor::escalc(measure="RR", ai=tpos, bi=tneg,
                    ci=cpos, di=cneg, data=metadat::dat.bcg)
```

```
# obtaining all three estimators and inference
# number of bootstrap iterates
# should be larger in practice
R = 100
confounded_meta( method="calibrated",  # for both methods
                 q = log(0.90),
                 r = 0.20,
                 tail="below",
                 muB = log(1.5),
                 dat = d,
                 yi.name = "yi",
                 vi.name = "vi",
                 R = 100 )

# passing only arguments needed for prop point estimate
confounded_meta( method="calibrated",
                 q = log(0.90),
                 tail="below",
                 muB = log(1.5),
                 give.CI = FALSE,
                 dat = d,
                 yi.name = "yi",
                 vi.name = "vi" )

# passing only arguments needed for Tmin, Gmin point estimates
confounded_meta( method="calibrated",
                 q = log(0.90),
                 r = 0.10,
                 tail="below",
                 give.CI = FALSE,
                 dat = d,
                 yi.name = "yi",
                 vi.name = "vi" )

##### Using Parametric Method #####
# fit random-effects meta-analysis
m = metafor::rma.uni(yi= d$yi,
                     vi=d$vi,
                     knha=TRUE,
                     measure="RR",
                     method="REML" )

yr = as.numeric(m$b)  # metafor returns on log scale
vyr = as.numeric(m$vb)
t2 = m$tau2
vt2 = m$se.tau2^2

# obtaining all three estimators and inference
# now the proportion considers heterogeneous bias
confounded_meta( method = "parametric",
                 q=log(0.90),
                 r=0.20,
                 tail = "below",
```

```
                      muB=log(1.5),
                      sigB=0.1,
                      yr=yr,
                      vyr=vyr,
                      t2=t2,
                      vt2=vt2,
                      CI.level=0.95 )

# passing only arguments needed for prop point estimate
confounded_meta( method = "parametric",
                 q=log(0.90),
                 tail = "below",
                 muB=log(1.5),
                 sigB = 0,
                 yr=yr,
                 t2=t2,
                 CI.level=0.95 )

# passing only arguments needed for Tmin, Gmin point estimates
confounded_meta( method = "parametric",
                 q = log(0.90),
                 sigB = 0,
                 r = 0.10,
                 tail = "below",
                 yr=yr,
                 t2=t2,
                 CI.level=0.95 )
```

---

confounding   *Unmeasured confounding*

Description

A type of bias. Declares that unmeasured confounding will be a component of interest in the multi-bias sensitivity analysis. Generally used within other functions, its output is returned invisibly.

Usage

```
confounding(..., verbose = FALSE)
```

Arguments

`...`	Other arguments. Not currently used for this function.
`verbose`	Logical. If `TRUE`, returns warnings and messages immediately. Defaults to `FALSE` because it is generally used within the [multi_bias()](#) function, which will print the same messages/warnings.

Value

Invisibly returns a list with components n (2, the degree of the polynomial in the numerator), d (1, the degree of the polynomial in the denominator), mess (any messages/warnings that should be printed for the user), and bias ("confounding").

Examples

```
# returns invisibly without print()
print(confounding())

# Calculate an E-value for unmeasured confounding only
multi_evalue(est = RR(4), biases = confounding())
```

---

convert_measures   *Convert an effect measure*

Description

These helper functions are mostly used internally to convert effect measures for the calculation of E-values. The approximate conversion of odds and hazard ratios to risk ratios depends on whether the rare outcome assumption is made.

Usage

```
toRR(est, rare, delta = 1, ...)

toMD(est, delta = 1, ...)
```

Arguments

est	The effect estimate; constructed with one of RR(), OR(), HR(), MD(), OLS().
rare	When converting a OR() or HR() estimate, a logical indicating whether the outcome is sufficiently rare to approximate a risk ratio.
delta	When converting an OLS() estimate, the contrast of interest in the exposure. Defaults to 1 (a 1-unit contrast in the exposure).
...	Arguments passed to other methods.

Details

Uses the conversions listed in Table 2 of VanderWeele TJ, Ding P. *Sensitivity Analysis in Observational Research: Introducing the E-Value.* Annals of Internal Medicine. 2017;167(4):268–75.

See references.

Regarding the continuous outcome, the function uses the effect-size conversions in Chinn (2000) and VanderWeele (2017) to approximately convert the mean difference between these exposure "groups" to the odds ratio that would arise from dichotomizing the continuous outcome.

Value

An object of class "estimate" and the desired effect measure. Also includes as an attribute its conversion history.

References

Chinn, S (2000). A simple method for converting an odds ratio to effect size for use in meta-analysis. *Statistics in Medicine*, 19(22), 3127-3131.

VanderWeele, TJ (2017). On a square-root transformation of the odds ratio for a common outcome. *Epidemiology*, 28(6), e58.

VanderWeele TJ (2020). *Optimal approximate conversions of odds ratios and hazard ratios to risk ratios.* Biometrics.

Examples

```
# Both odds ratios are 3, but will be treated differently
# depending on whether rare outcome assumption is reasonable
OR(3, rare = FALSE)
OR(3, rare = TRUE)
toRR(OR(3, rare = FALSE))
toRR(OR(3, rare = TRUE))
attributes(toRR(toMD(OLS(3, sd = 1.2), delta = 1)))
```

---

    effect_measures   *Declare an effect measure*

Description

These functions allow the user to declare that an estimate is a certain type of effect measure: risk ratio (RR), odds ratio (OR), hazard ratio (HR), risk difference (RD), linear regression coefficient (OLS), or mean standardized difference (MD).

Usage

```
RR(est)

OR(est, rare)

HR(est, rare)

RD(est)

OLS(est, sd)

MD(est)
```

Arguments

est	The effect estimate (numeric).
rare	Logical. Whether the outcome is sufficiently rare for use of risk ratio approximates; if not, approximate conversions are used. Used only for HR() and OR(); see Details.
sd	The standard deviation of the outcome (or residual standard deviation). Used only for OLS(); see Details.

Details

The conversion functions use these objects to convert between effect measures when necessary to calculate E-values. Read more about the conversions in Table 2 of VanderWeele TJ, Ding P. *Sensitivity Analysis in Observational Research: Introducing the E-Value.* Annals of Internal Medicine. 2017;167(4):268–75.

See also VanderWeele TJ. *Optimal approximate conversions of odds ratios and hazard ratios to risk ratios.* Biometrics. 2019 Jan 6;(September 2018):1–7.

For OLS(), sd must be specified. A true standardized mean difference for linear regression would use sd = SD(Y | X, C), where Y is the outcome, X is the exposure of interest, and C are any adjusted covariates. See Examples for how to extract this from lm. A conservative approximation would instead use sd = SD(Y). Regardless, the reported E-value for the confidence interval treats sd as known, not estimated.

Value

An object of classes "estimate" and the measure of interest, containing the effect estimate and any other attributes to be used in future calculations.

Examples

```
# Both odds ratios are 3, but will be treated differently in E-value calculations
# depending on whether rare outcome assumption is reasonable
OR(3, rare = FALSE)
OR(3, rare = TRUE)
evalue(OR(3, rare = FALSE))
evalue(OR(3, rare = TRUE))
attributes(OR(3, rare = FALSE))

# If an estimate was constructed via conversion from another effect measure,
# we can see the history of a conversion using the summary() function
summary(toRR(OR(3, rare = FALSE)))
summary(toRR(OLS(3, sd = 1)))

# Estimating sd for an OLS estimate
# first standardizing conservatively by SD(Y)
data(lead)
ols = lm(age ~ income, data = lead)
est = ols$coefficients[2]
sd = sd(lead$age)
summary(evalue(OLS(est, sd)))
```

```
# now use residual SD to avoid conservatism
# here makes very little difference because income and age are
# not highly correlated
sd = summary(ols)$sigma
summary(evalue(OLS(est, sd)))
```

---

evalue   *Compute an E-value for unmeasured confounding*

Description

Returns a data frame containing point estimates, the lower confidence limit, and the upper confidence limit on the risk ratio scale (possibly through an approximate conversion) as well as E-values for the point estimate and the confidence interval limit closer to the null.

Usage

```
evalue(est, lo = NA, hi = NA, se = NA, delta = 1, true = c(0, 1), ...)
```

Arguments

est	The effect estimate that was observed but which is suspected to be biased. A number of class "estimate" (constructed with RR(), OR(), HR(), OLS(), or MD(); for E-values for risk differences, see evalues.RD()).
lo	Optional. Lower bound of the confidence interval. If not an object of class "estimate", assumed to be on the same scale as est.
hi	Optional. Upper bound of the confidence interval. If not an object of class "estimate", assumed to be on the same scale as est.
se	The standard error of the point estimate, for est of class "OLS"
delta	The contrast of interest in the exposure, for est of class "OLS"
true	A number to which to shift the observed estimate to. Defaults to 1 for ratio measures (RR(), OR(), HR()) and 0 for additive measures (OLS(), MD()).
...	Arguments passed to other methods.

Details

An E-value for unmeasured confounding is minimum strength of association, on the risk ratio scale, that unmeasured confounder(s) would need to have with both the treatment and the outcome to fully explain away a specific treatment–outcome association, conditional on the measured covariates.

The estimate is converted appropriately before the E-value is calculated. See conversion functions for more details. The point estimate and confidence limits after conversion are returned, as is the E-value for the point estimate and the confidence limit closest to the proposed "true" value (by default, the null value.)

For an OLS() estimate, the E-value is for linear regression with a continuous exposure and outcome. Regarding the continuous exposure, the choice of delta defines essentially a dichotomization in

the exposure between hypothetical groups of subjects with exposures equal to an arbitrary value *c* versus to another hypothetical group with exposures equal to *c* + delta.

For example, if resulting E-value is 2, this means that unmeasured confounder(s) would need to double the probability of a subject's having exposure equal to *c* + delta instead of *c*, and would also need to double the probability of being high versus low on the outcome, in which the cutoff for "high" versus "low" is arbitrary subject to some distributional assumptions (Chinn, 2000).

References

1. Ding & VanderWeele (2016). Sensitivity analysis without assumptions. *Epidemiology.* 27(3), 368.

2. VanderWeele & Ding (2017). Sensitivity analysis in observational research: Introducing the E-value. *Annals of Internal Medicine.* 27(3), 368.

Examples

```
# compute E-value for leukemia example in VanderWeele and Ding (2017)
evalue(RR(0.80), 0.71, 0.91)

# you can also pass just the point estimate
# and return just the E-value for the point estimate with summary()
summary(evalue(RR(0.80)))

# demonstrate symmetry of E-value
# this apparently causative association has same E-value as the above
summary(evalue(RR(1 / 0.80)))

# E-value for a non-null true value
summary(evalue(RR(2), true = 1.5))

## Hsu and Small (2013 Biometrics) Data
## sensitivity analysis after log-linear or logistic regression
head(lead)

## log linear model -- obtain the conditional risk ratio
lead.loglinear = glm(lead ~ ., family = binomial(link = "log"),
                        data = lead[,-1])
est_se = summary(lead.loglinear)$coef["smoking", c(1, 2)]

est     = RR(exp(est_se[1]))
lowerRR = exp(est_se[1] - 1.96*est_se[2])
upperRR = exp(est_se[1] + 1.96*est_se[2])
evalue(est, lowerRR, upperRR)

## logistic regression -- obtain the conditional odds ratio
lead.logistic = glm(lead ~ ., family = binomial(link = "logit"),
                        data = lead[,-1])
est_se = summary(lead.logistic)$coef["smoking", c(1, 2)]

est     = OR(exp(est_se[1]), rare = FALSE)
lowerOR = exp(est_se[1] - 1.96*est_se[2])
```

```
          upperOR  = exp(est_se[1] + 1.96*est_se[2])
          evalue(est, lowerOR, upperOR)

          ## linear regression
          # standardizing conservatively by SD(Y)
          ols = lm(age ~ income, data = lead)
          est = OLS(ols$coefficients[2], sd = sd(lead$age))

          # for a 1-unit increase in income
          evalue(est = est,
                 se = summary(ols)$coefficients['income', 'Std. Error'])

          # for a 0.5-unit increase in income
          evalue(est = est,
                 se = summary(ols)$coefficients['income', 'Std. Error'],
                 delta = 0.5)

          # E-value for Cohen's d = 0.5 with SE = 0.25
          evalue(est = MD(.5), se = .25)

          # compute E-value for HR = 0.56 with CI: [0.46, 0.69]
          # for a common outcome
          evalue(HR(0.56, rare = FALSE), lo = 0.46, hi = 0.69)
          # for a rare outcome
          evalue(HR(0.56, rare = TRUE), lo = 0.46, hi = 0.69)
```

---

evalues.HR   *Compute E-value for a hazard ratio and its confidence interval limits*

Description

Returns a data frame containing point estimates, the lower confidence limit, and the upper confidence limit on the risk ratio scale (through an approximate conversion if needed when outcome is common) as well as E-values for the point estimate and the confidence interval limit closer to the null.

Usage

```
evalues.HR(est, lo = NA, hi = NA, rare = NA, true = 1, ...)
```

Arguments

est	The point estimate
lo	The lower limit of the confidence interval
hi	The upper limit of the confidence interval
rare	1 if outcome is rare (<15 percent at end of follow-up); 0 if outcome is not rare (>15 percent at end of follow-up)
true	The true HR to which to shift the observed point estimate. Typically set to 1 to consider a null true effect.
...	Arguments passed to other methods.

Examples

```
# compute E-value for HR = 0.56 with CI: [0.46, 0.69]
# for a common outcome
evalues.HR(0.56, 0.46, 0.69, rare = FALSE)
```

---

evalues.IC	*Compute an E-value for unmeasured confounding for an additive interaction contrast*

Description

Computes the E-value for an additive interaction contrast, representing the difference between stratum Z=1 and stratum Z=0 in the causal risk differences for a binary treatment X.

Usage

```
evalues.IC(
  stat,
  true = 0,
  unidirBias = FALSE,
  unidirBiasDirection = NA,
  p1_1,
  p1_0,
  n1_1,
  n1_0,
  f1,
  p0_1,
  p0_0,
  n0_1,
  n0_0,
  f0,
  alpha = 0.05
)
```

Arguments

stat	The statistic for which to compute the E-value ("est" for the interaction contrast point estimate or "CI" for its lower confidence interval limit)
true	The true (unconfounded) value to which to shift the specified statistic (point estimate or confidence interval limit). Should be smaller than the confounded statistic.
unidirBias	Whether the direction of confounding bias is assumed to be the same in both strata of Z (TRUE or FALSE); see Details
unidirBiasDirection	
	If bias is assumed to be unidirectional, its assumed direction ("positive", "negative", or "unknown"; see Details). If bias is not assumed to be unidirectional, this argument should be NA.

p1_1	The probability of the outcome in stratum Z=1 with treatment X=1
p1_0	The probability of the outcome in stratum Z=1 with treatment X=0
n1_1	The sample size in stratum Z=1 with treatment X=1
n1_0	The sample size in stratum Z=1 with treatment X=0
f1	The probability in stratum Z=1 of having treatment X=1
p0_1	The probability of the outcome in stratum Z=0 with treatment X=1
p0_0	The probability of the outcome in stratum Z=0 with treatment X=0
n0_1	The sample size in stratum Z=0 with treatment X=1
n0_0	The sample size in stratum Z=0 with treatment X=0
f0	The probability in stratum Z=0 of treatment X=1
alpha	The alpha-level to be used for p-values and confidence intervals

Details

E-values for additive effect modification:

The interaction contrast is a measure of additive effect modification that represents the difference between stratum Z=1 versus stratum Z=0 of the causal risk differences relating a treatment X to an outcome Y. The estimated interaction contrast is given by:

(p1_1 - p1_0) - (p0_1 - p0_0)

To use this function, the strata (Z) should be coded such that the confounded interaction contrast is positive rather than negative.

If, in one or both strata of Z, there are unmeasured confounders of the treatment-outcome association, then the interaction contrast may be biased as well (Mathur et al., 2021). The E-value for the interaction contrast represents the minimum strength of association, on the risk ratio scale, that unmeasured confounder(s) would need to have with both the treatment (X) and the outcome (Y) in both strata of Z to fully explain away the observed interaction contrast, conditional on the measured covariates. This bound is attained when the strata have confounding bias in opposite directions ("potentially multidirectional bias"). Alternatively, if one assumes that the direction of confounding is the same in each stratum of Z ("unidirectional bias"), then the E-value for the interaction contrast is defined as the minimum strength of association, on the risk ratio scale, that unmeasured confounder(s) would need to have with both the treatment (X) and the outcome (Y) in *at least one* stratum of Z to fully explain away the observed interaction contrast, conditional on the measured covariates. This bound under unidirectional confounding arises when one stratum is unbiased. See Mathur et al. (2021) for details.

As for the standard E-value for main effects (Ding & VanderWeele, 2016; VanderWeele & Ding, 2017), the E-value for the interaction contrast can be computed for both the point estimate and the lower confidence interval limit, and it can be also be calculated for shifting the estimate or confidence interval to a non-null value via the argument true.

Specifying the bias direction:

The argument unidirBias indicates whether you are assuming unidirectional bias (unidirBias = TRUE) or not (unidirBias = FALSE). The latter is the default because it is more conservative and requires the fewest assumptions. When setting unidirBias = FALSE, there is no need to specify the direction of bias via unidirBiasDir. However, when setting unidirBias = TRUE, the direction of bias does need to be specified via unidirBiasDir, whose options are:

- `unidirBiasDir = "positive"`: Assumes that the risk differences in both strata of Z are positively biased.
- `unidirBiasDir = "negative"`: Assumes that the risk differences in both strata of Z are negatively biased.
- `unidirBiasDir = "unknown"`: Assumes that the risk differences in both strata of Z are biased in the same direction, but that the direction could be either positive or negative.

Adjusted interaction contrasts:

If your estimated interaction contrast has been adjusted for covariates, then you can use covariate-adjusted probabilities for p1_1, p1_0, p0_1, and p0_0. For example, these could be fitted probabilities from a covariate-adjusted regression model.

Multiplicative effect modification:

For multiplicative measures of effect modification (e.g., the ratio of risk ratios between the two strata of Z), you can simply use the function `evalue`. To allow the bias to be potentially multidirectional, you would pass the square-root of your multiplicative effect modification estimate on the risk ratio scale to `evalue` rather than the estimate itself. To assume unidirectional bias, regardless of direction, you would pass the multiplicative effect modification estimate itself to `evalue`. See Mathur et al. (2021) for details.

Value

Returns a list containing two dataframes (`evalues` and `RDt`). The E-value itself can be accessed as `evalues$evalue`.

The dataframe `evalues` contains the E-value, the corresponding bias factor, the bound on the interaction contrast if confounding were to attain that bias factor (this bound will be close to `true`, by construction), and the direction of bias when the bias factor is attained. If you specify that the bias is potentially multidirectional, is unidirectional and positive, or is unidirectional and negative, the returned direction of bias will simply be what you requested. If you specify unidirectional bias of unknown direction, the bias direction will be either positive or negative depending on which direction produces the maximum bias.

The dataframe `RDt` contains, for each stratum and for the interaction contrast, bias-corrected estimates (risk differences for the strata and the interaction contrast for `stratum = effectMod`), their standard errors, their confidence intervals, and their p-values. These estimates are bias-corrected for the worst-case bias that could arise for confounder(s) whose strength of association are no more severe than the requested E-value for either the estimate or the confidence interval (i.e., the bias factor indicated by `evalues$biasFactor`). The bias-corrected risk differences for the two strata (`stratum = "1"` and `stratum = "0"`) are corrected in the direction(s) indicated by `evalues$biasDir`.

If you specify unidirectional bias of unknown direction, the E-value is calculated by taking the minimum of the E-value under positive unidirectional bias and the E-value under negative unidirectional bias. With this specification, a third dataframe (`candidates`) will be returned. This is similar to `evalues`, but contains the results for positive unidirectional bias and negative unidirectional bias (the two "candidate" E-values that were considered).

References

1. Mathur MB, Smith LH, Yoshida K, Ding P, VanderWeele TJ (2021). E-values for effect modification and approximations for causal interaction. Under review.

2. Ding P & VanderWeele TJ (2016). Sensitivity analysis without assumptions. *Epidemiology.* 27(3), 368.

3. VanderWeele TJ & Ding P (2017). Sensitivity analysis in observational research: Introducing the E-value. *Annals of Internal Medicine.* 27(3), 368.

Examples

```
### Letenneur et al. (2000) example data
# this is the example given in Mathur et al. (2021)
# Z: sex (w = women, m = male; males are the reference category)
# Y: dementia (1 = developed dementia, 0 = did not develop dementia )
# X: low education (1 = up to 7 years, 0 = at least 12 years)
# n: sample size

# data for women
nw_1 = 2988
nw_0 = 364
dw = data.frame(  Y = c(1, 1, 0, 0),
                  X = c(1, 0, 1, 0),
                  n = c( 158, 6, nw_1-158, nw_0-6 ) )

# data for men
nm_1 = 1790
nm_0 = 605
dm = data.frame(  Y = c(1, 1, 0, 0),
                  X = c(1, 0, 1, 0),
                  n = c( 64, 17, nm_1-64, nm_0-17 ) )

# P(Y = 1 | X = 1) and P(Y = 1 | X = 0) for women and for men
( pw_1 = dw$n[ dw$X == 1 & dw$Y == 1 ] / sum(dw$n[ dw$X == 1 ]) )
( pw_0 = dw$n[ dw$X == 0 & dw$Y == 1 ] / sum(dw$n[ dw$X == 0 ]) )
( pm_1 = dm$n[ dm$X == 1 & dm$Y == 1 ] / sum(dm$n[ dm$X == 1 ]) )
( pm_0 = dm$n[ dm$X == 0 & dm$Y == 1 ] / sum(dm$n[ dm$X == 0 ]) )

# prevalence of low education among women and among men
fw = nw_1 / (nw_1 + nw_0)
fm = nm_1 / (nm_1 + nm_0)

# confounded interaction contrast estimate
( pw_1 - pw_0 ) - ( pm_1 - pm_0 )

### E-values without making assumptions on direction of confounding bias
# for interaction contrast point estimate
evalues.IC( stat = "est",

            p1_1 = pw_1,
            p1_0 = pw_0,
            n1_1 = nw_1,
            n1_0 = nw_0,
            f1 = fw,

            p0_1 = pm_1,
```

```
                 p0_0 = pm_0,
                 n0_1 = nm_1,
                 n0_0 = nm_0,
                 f0 = fm )

# and for its lower CI limit
evalues.IC( stat = "CI",

                 p1_1 = pw_1,
                 p1_0 = pw_0,
                 n1_1 = nw_1,
                 n1_0 = nw_0,
                 f1 = fw,

                 p0_1 = pm_1,
                 p0_0 = pm_0,
                 n0_1 = nm_1,
                 n0_0 = nm_0,
                 f0 = fm )

### E-values assuming unidirectonal confounding of unknown direction
# for interaction contrast point estimate
evalues.IC( stat = "est",
                 unidirBias = TRUE,
                 unidirBiasDirection = "unknown",

                 p1_1 = pw_1,
                 p1_0 = pw_0,
                 n1_1 = nw_1,
                 n1_0 = nw_0,
                 f1 = fw,

                 p0_1 = pm_1,
                 p0_0 = pm_0,
                 n0_1 = nm_1,
                 n0_0 = nm_0,
                 f0 = fm )

# and for its lower CI limit
evalues.IC( stat = "CI",
                 unidirBias = TRUE,
                 unidirBiasDirection = "unknown",

                 p1_1 = pw_1,
                 p1_0 = pw_0,
                 n1_1 = nw_1,
                 n1_0 = nw_0,
                 f1 = fw,

                 p0_1 = pm_1,
                 p0_0 = pm_0,
                 n0_1 = nm_1,
                 n0_0 = nm_0,
```

```
        f0 = fm )
```

---

evalues.MD   *Compute E-value for a difference of means and its confidence interval*
                                    limits

Description

Returns a data frame containing point estimates, the lower confidence limit, and the upper confidence limit on the risk ratio scale (through an approximate conversion) as well as E-values for the point estimate and the confidence interval limit closer to the null.

Usage

```
evalues.MD(est, se = NA, true = 0, ...)
```

Arguments

est	The point estimate as a standardized difference (i.e., Cohen's d)
se	The standard error of the point estimate
true	The true standardized mean difference to which to shift the observed point estimate. Typically set to 0 to consider a null true effect.
...	Arguments passed to other methods.

Details

Regarding the continuous outcome, the function uses the effect-size conversions in Chinn (2000) and VanderWeele (2017) to approximately convert the mean difference between the exposed versus unexposed groups to the odds ratio that would arise from dichotomizing the continuous outcome.

For example, if resulting E-value is 2, this means that unmeasured confounder(s) would need to double the probability of a subject's being exposed versus not being exposed, and would also need to double the probability of being high versus low on the outcome, in which the cutoff for "high" versus "low" is arbitrary subject to some distributional assumptions (Chinn, 2000).

References

Chinn, S (2000). A simple method for converting an odds ratio to effect size for use in meta-analysis. *Statistics in Medicine*, 19(22), 3127-3131.

VanderWeele, TJ (2017). On a square-root transformation of the odds ratio for a common outcome. *Epidemiology*, 28(6), e58.

Examples

```
# compute E-value if Cohen's d = 0.5 with SE = 0.25
evalues.MD(.5, .25)
```

---

evalues.OLS   *Compute E-value for a linear regression coefficient estimate*

Description

Returns a data frame containing point estimates, the lower confidence limit, and the upper confidence limit on the risk ratio scale (through an approximate conversion) as well as E-values for the point estimate and the confidence interval limit closer to the null.

Usage

```
evalues.OLS(est, se = NA, sd, delta = 1, true = 0, ...)
```

Arguments

est	The linear regression coefficient estimate (standardized or unstandardized)
se	The standard error of the point estimate
sd	The standard deviation of the outcome (or residual standard deviation); see Details
delta	The contrast of interest in the exposure
true	The true standardized mean difference to which to shift the observed point estimate. Typically set to 0 to consider a null true effect.
...	Arguments passed to other methods.

Details

This function is for linear regression with a continuous exposure and outcome. Regarding the continuous exposure, the choice of delta defines essentially a dichotomization in the exposure between hypothetical groups of subjects with exposures equal to an arbitrary value *c* versus to another hypothetical group with exposures equal to *c* + delta. Regarding the continuous outcome, the function uses the effect-size conversions in Chinn (2000) and VanderWeele (2017) to approximately convert the mean difference between these exposure "groups" to the odds ratio that would arise from dichotomizing the continuous outcome.

For example, if resulting E-value is 2, this means that unmeasured confounder(s) would need to double the probability of a subject's having exposure equal to *c* + delta instead of *c*, and would also need to double the probability of being high versus low on the outcome, in which the cutoff for "high" versus "low" is arbitrary subject to some distributional assumptions (Chinn, 2000).

A true standardized mean difference for linear regression would use sd = SD(Y | X, C), where Y is the outcome, X is the exposure of interest, and C are any adjusted covariates. See Examples for how to extract this from lm. A conservative approximation would instead use sd = SD(Y). Regardless, the reported E-value for the confidence interval treats sd as known, not estimated.

References

Chinn, S (2000). A simple method for converting an odds ratio to effect size for use in meta-analysis. *Statistics in Medicine*, 19(22), 3127-3131.

VanderWeele, TJ (2017). On a square-root transformation of the odds ratio for a common outcome. *Epidemiology*, 28(6), e58.

Examples

```
# first standardizing conservatively by SD(Y)
data(lead)
ols = lm(age ~ income, data = lead)

# for a 1-unit increase in income
evalues.OLS(est = ols$coefficients[2],
            se = summary(ols)$coefficients['income', 'Std. Error'],
            sd = sd(lead$age))

# for a 0.5-unit increase in income
evalues.OLS(est = ols$coefficients[2],
            se = summary(ols)$coefficients['income', 'Std. Error'],
            sd = sd(lead$age),
            delta = 0.5)

# now use residual SD to avoid conservatism
# here makes very little difference because income and age are
# not highly correlated
evalues.OLS(est = ols$coefficients[2],
            se = summary(ols)$coefficients['income', 'Std. Error'],
            sd = summary(ols)$sigma)
```

---

evalues.OR   *Compute E-value for an odds ratio and its confidence interval limits*

Description

Returns a data frame containing point estimates, the lower confidence limit, and the upper confidence limit on the risk ratio scale (through an approximate conversion if needed when outcome is common) as well as E-values for the point estimate and the confidence interval limit closer to the null.

Usage

```
evalues.OR(est, lo = NA, hi = NA, rare = NA, true = 1, ...)
```

Arguments

est	The point estimate
lo	The lower limit of the confidence interval
hi	The upper limit of the confidence interval
rare	1 if outcome is rare (<15 percent at end of follow-up); 0 if outcome is not rare (>15 percent at end of follow-up)
true	The true OR to which to shift the observed point estimate. Typically set to 1 to consider a null true effect.
...	Arguments passed to other methods.

Examples

```
# compute E-values for OR = 0.86 with CI: [0.75, 0.99]
# for a common outcome
evalues.OR(0.86, 0.75, 0.99, rare = FALSE)

## Example 2
## Hsu and Small (2013 Biometrics) Data
## sensitivity analysis after log-linear or logistic regression

head(lead)

## log linear model -- obtain the conditional risk ratio
lead.loglinear = glm(lead ~ ., family = binomial(link = "log"),
                          data = lead[,-1])
est = summary(lead.loglinear)$coef["smoking", c(1, 2)]

RR      = exp(est[1])
lowerRR = exp(est[1] - 1.96*est[2])
upperRR = exp(est[1] + 1.96*est[2])
evalues.RR(RR, lowerRR, upperRR)

## logistic regression -- obtain the conditional odds ratio
lead.logistic = glm(lead ~ ., family = binomial(link = "logit"),
                          data = lead[,-1])
est = summary(lead.logistic)$coef["smoking", c(1, 2)]

OR      = exp(est[1])
lowerOR = exp(est[1] - 1.96*est[2])
upperOR = exp(est[1] + 1.96*est[2])
evalues.OR(OR, lowerOR, upperOR, rare=FALSE)
```

---

evalues.RD	*Compute E-value for a population-standardized risk difference and its confidence interval limits*

Description

Returns E-values for the point estimate and the lower confidence interval limit for a positive risk difference. If the risk difference is negative, the exposure coding should be first be reversed to yield a positive risk difference.

Usage

```
evalues.RD(n11, n10, n01, n00, true = 0, alpha = 0.05, grid = 1e-04, ...)
```

Arguments

n11	Number of exposed, diseased individuals
n10	Number of exposed, non-diseased individuals
n01	Number of unexposed, diseased individuals
n00	Number of unexposed, non-diseased individuals
true	True value of risk difference to which to shift the point estimate. Usually set to 0 to consider the null.
alpha	Alpha level
grid	Spacing for grid search of E-value
...	Arguments passed to other methods.

Examples

```
## example 1
## Hammond and Holl (1958 JAMA) Data
## Two by Two Table
##           Lung Cancer    No Lung Cancer
##Smoker    397            78557
##Nonsmoker 51             108778

# E-value to shift observed risk difference to 0
evalues.RD(397, 78557, 51, 108778)

# E-values to shift observed risk difference to other null values
evalues.RD(397, 78557, 51, 108778, true = 0.001)
```

---

evalues.RR	*Compute E-value for a risk ratio or rate ratio and its confidence interval limits*

Description

Returns a data frame containing point estimates, the lower confidence limit, and the upper confidence limit for the risk ratio (as provided by the user) as well as E-values for the point estimate and the confidence interval limit closer to the null.

Usage

```
evalues.RR(est, lo = NA, hi = NA, true = 1, ...)
```

Arguments

est	The point estimate
lo	The lower limit of the confidence interval
hi	The upper limit of the confidence interval
true	The true RR to which to shift the observed point estimate. Typically set to 1 to consider a null true effect.
...	Arguments passed to other methods.

Examples

```
# compute E-value for leukemia example in VanderWeele and Ding (2017)
evalues.RR(0.80, 0.71, 0.91)

# you can also pass just the point estimate
evalues.RR(0.80)

# demonstrate symmetry of E-value
# this apparently causative association has same E-value as the above
evalues.RR(1 / 0.80)
```

---

lead	*An example dataset*

Description

An example dataset from Hsu and Small (Biometrics, 2013).

Usage

```
lead
```

Format

An object of class `data.frame` with 3340 rows and 18 columns.

---

misclassification   *Misclassification*

Description

A type of bias. Declares that (differential) misclassification will be a component of interest in the multi-bias sensitivity analysis. Generally used within other functions; its output is returned invisibly.

Usage

```
misclassification(
  ...,
  rare_outcome = FALSE,
  rare_exposure = FALSE,
  verbose = FALSE
)
```

Arguments

`...`	Arguments describing the type of misclassification. Currently two options: "outcome" or "exposure".
`rare_outcome`	Logical. Is the outcome rare enough that outcome odds ratios approximate risk ratios? Only needed when considering exposure misclassification. Note that `rare_outcome = FALSE` returns an error, as this option is not currently available.
`rare_exposure`	Logical. Is the exposure rare enough that exposure odds ratios approximate risk ratios? Only needed when considering exposure misclassification.
`verbose`	Logical. If `TRUE`, returns warnings and messages immediately. Defaults to `FALSE` because it is generally used within the [multi_bias()](#) function, which will print the same messages/warnings.

Value

Invisibly returns a list with components whose values depend on the options chosen: `n` (the degree of the polynomial in the numerator), `d` (the degree of the polynomial in the denominator), `m` (the parameters in the bias factor), `mess` (any messages/warnings that should be printed for the user), and `bias("misclassification")`.

Examples

```
# returns invisibly without print()
print(misclassification("outcome"))

# Calculate an E-value for misclassification
multi_evalue(est = RR(4),
        biases = misclassification("exposure",
                  rare_outcome = TRUE, rare_exposure = TRUE))
```

---

multi_bias   *Create a set of biases for a multi-bias sensitivity analysis*

Description

Multiple biases ([confounding()](), [selection()](), and/or [misclassification()]()) can be assessed simultaneously after creating a `multi_bias` object using this function.

Usage

```
multi_bias(..., verbose = TRUE)
```

Arguments

`...`	Biases ([confounding()](), [selection()](), and/or [misclassification()]()), each possibly including arguments specifying more detail about the bias of interest. Selection and confounding should be listed in the order in which they affect the data (see ordering of the biases)
`verbose`	Logical. If `TRUE`, returns warnings and messages immediately. Defaults to `TRUE`.

Value

Invisibly returns a list with components whose values depend on the options chosen: n (the degree of the polynomial in the numerator), d (the degree of the polynomial in the denominator), m (the parameters in the bias factor), mess (any messages/warnings that should be printed for the user), and bias("misclassification").

Examples

```
biases <- multi_bias(confounding(),
                     selection("general"))

# print() lists the arguments for the multi_bound() function
print(biases)

# summary() provides more information
# with parameters in latex notation if latex = TRUE
summary(biases, latex = TRUE)

# Calculate a bound
multi_bound(biases = biases,
            RRAUc = 1.5, RRUcY = 2, RRUsYA1 = 1.25,
            RRSUsA1 = 4, RRUsYA0 = 3, RRSUsA0 = 2)
```

---

multi_bound   *Calculate a bound for the bias*

Description

Function used to calculate the maximum factor by which a risk ratio is biased, given possible values
for each of the parameters that describe the bias factors for each type of bias.

Usage

```
multi_bound(
  biases,
  RRAUc = NULL,
  RRUcY = NULL,
  RRUsYA1 = NULL,
  RRSUsA1 = NULL,
  RRUsYA0 = NULL,
  RRSUsA0 = NULL,
  RRAUscS = NULL,
  RRUscYS = NULL,
  RRAYy = NULL,
  ORYAa = NULL,
  RRYAa = NULL,
  RRAYyS = NULL,
  ORYAaS = NULL,
  RRYAaS = NULL,
  RRAUsS = NULL,
  RRUsYS = NULL
)
```

Arguments

biases   A set of biases (or single bias) to include in the calculation of the bound. A
              single object constructed with the [multi_bias()](#) function, it may include any
              or all of [confounding()](#), [selection()](#), and [misclassification()](#), and any
              of the options described in the documentation for those functions.

RRAUc   Named parameter values with which to calculate a bound. Names must cor-
              respond to the parameters defining the biases provided by biases. Help with
              names can be found by running print(multi_bias(...)) for the biases of
              interest. Unnecessary parameters are ignored with a warning.

RRUcY   See RRAUc

RRUsYA1   See RRAUc

RRSUsA1   See RRAUc

RRUsYA0   See RRAUc

RRSUsA0   See RRAUc

RRAUscS	See RRAUc
RRUscYS	See RRAUc
RRAYy	See RRAUc
ORYAa	See RRAUc
RRYAa	See RRAUc
RRAYyS	See RRAUc
ORYAaS	See RRAUc
RRYAaS	See RRAUc
RRAUsS	See RRAUc
RRUsYS	See RRAUc

Details

The names of the parameters in the bound can be found for a given set of biases with `print(biases)`. Running `summary(biases)` shows the equivalent notation used in the output of the [multi_evalue()](#) function.

Value

Returns the value of the bound formed as a function of the provided parameters.

Examples

```
multi_bound(multi_bias(confounding()),
            RRAUc = 2.2, RRUcY = 1.7)

biases <- multi_bias(confounding(), selection("S = U"),
                     misclassification("exposure",
                     rare_outcome = TRUE, rare_exposure = FALSE))

print(biases)

multi_bound(biases,
            RRAUc = 3, RRUcY = 2, RRSUsA1 = 2.3,
            RRSUsA0 = 1.7, ORYAaS = 5.2)
```

---

multi_evalue	*Calculate a multiple-bias E-value*

Description

Calculate an E-value for a specified set of biases.

Usage

```
multi_evalue(biases, est, ...)

multi_evalues.HR(
  biases,
  est,
  lo = NA,
  hi = NA,
  rare = NULL,
  true = 1,
  verbose = TRUE,
  ...
)

multi_evalues.OR(
  biases,
  est,
  lo = NA,
  hi = NA,
  rare = NULL,
  true = 1,
  verbose = TRUE,
  ...
)

multi_evalues.RR(biases, est, lo = NA, hi = NA, true = 1, verbose = TRUE, ...)
```

Arguments

biases	An object created by [multi_bias()](#) (or a single bias) to include in the calculation of the E-value. May include any or all of [confounding()](#), [selection()](#), and [misclassification()](#), and any of the options described in the documentation for those functions.
est	The effect estimate that was observed but which is suspected to be biased. This may be of class "estimate" (constructed with [RR()](#), [OR()](#), or [HR()](#), or more information can be provided using the other arguments.
...	Arguments passed to other methods.
lo	Optional. Lower bound of the confidence interval. If not an object of class "estimate", assumed to be on the same scale as est.
hi	Optional. Upper bound of the confidence interval. If not an object of class "estimate", assumed to be on the same scale as est.
rare	Logical indicating whether outcome is sufficiently rare for risk ratio approximation to hold.
true	A number to which to shift the observed estimate to. Defaults to 1. If not an object of class "estimate", assumed to be on the same scale as est.
verbose	Logical indicating whether or not to print information about which parameters the multi-bias E-value refers to. Defaults to TRUE.

Value

Returns a multiple bias E-value, of class "multi_evalue", describing the value that each of a number of parameters would have to have for the observed effect measure to be completely explained by bias.

Examples

```
# Calculate an E-value for unmeasured confounding
multi_evalue(est = RR(4), biases = confounding())
# Equivalent to
evalues.RR(4)

# Calculate a multi-bias E-value for selection bias
# and misclassification
multi_evalue(est = RR(2.5),
        biases = multi_bias(selection("selected"),
                    misclassification("outcome")))

# Calculate a multi-bias E-value for all three
# available types of bias
biases <- multi_bias(confounding(),
                    selection("general", "S = U"),
                    misclassification("exposure",
                            rare_outcome = TRUE))
multi_evalue(est = RR(2.5), biases = biases)

# Calculate a multi-bias E-value for a non-rare OR
# using the square root approximation
multi_evalue(est = OR(2.5, rare = FALSE), biases = biases)

# Calculate a non-null multi-bias E-value
multi_evalue(est = RR(2.5), biases = biases, true = 2)
```

---

selection	*Selection bias*

Description

A type of bias. Declares that selection bias will be a component of interest in the multi-bias sensitivity analysis. Generally used within other functions; its output is returned invisibly.

Usage

```
selection(..., verbose = FALSE)
```

Arguments

...	Optional arguments describing the type of potential selection bias. Options are "general" (general selection bias, the default if no options are chosen), "increased risk" and "decreased risk" (assumptions about the direction of risk in the selected population), "S = U" (simplification used if the biasing characteristic is common to the entire selected population), and "selected" (when the target of inference is the selected population only). Errors are produced when incompatible assumptions are chosen.
verbose	Logical. If TRUE, returns warnings and messages immediately. Defaults to FALSE because it is generally used within the multi_bias() function, which will print the same messages/warnings.

Value

Invisibly returns a list with components whose values depend on the options chosen: n (the degree of the polynomial in the numerator), d (the degree of the polynomial in the denominator),mess (any messages/warnings that should be printed for the user), and bias("selection").

Examples

```
# returns invisibly without print()
print(selection("general", "increased risk"))

# Calculate an E-value for selection bias only
multi_evalue(est = RR(4),
         biases = selection("general", "increased risk"))
```

---

selection_evalue	*Compute selection bias E-value for a hazard ratio and its confidence interval limits*

Description

Returns a data frame containing point estimates, the lower confidence limit, and the upper confidence limit on the risk ratio scale (through an approximate conversion if needed when outcome is common) as well as E-values for the point estimate and the confidence interval limit closer to the null.

Usage

```
selection_evalue(
  est,
  lo = NA,
  hi = NA,
  true = 1,
  sel_pop = FALSE,
```

```
  S_eq_U = FALSE,
  risk_inc = FALSE,
  risk_dec = FALSE,
  ...
)
```

Arguments

est	The point estimate: a risk, odds, or hazard ratio. An object of class "estimate", it should be constructed with functions `RR()`, `OR()`, or `HR()`.
lo	The lower limit of the confidence interval
hi	The upper limit of the confidence interval
true	The true value to which to shift the observed point estimate. Typically set to 1 to consider a null true effect.
sel_pop	Whether inference is specific to selected population (TRUE) or entire population (FALSE). Defaults to FALSE.
S_eq_U	Whether the unmeasured factor is assumed to be a defining characteristic of the selected population. Defaults to FALSE.
risk_inc	Whether selection is assumed to be associated with increased risk of the outcome in both exposure groups. Defaults to FALSE.
risk_dec	Whether selection is assumed to be associated with decreased risk of the outcome in both exposure groups. Defaults to FALSE.
...	Arguments passed to other methods.

Details

A selection bias E-value is a summary measure that helps assess susceptibility of a result to selection bias. Each of one or more parameters characterizing the extent of the bias must be greater than or equal to this value to be sufficient to shift an estimate (est) to the null or other true value (true). The parameters, as defined in Smith and VanderWeele 2019, depend on assumptions an investigator is willing to make (see arguments sel_pop, S_eq_U, risk_inc, risk_dec). The function prints a message about which parameters the selection bias E-value refers to given the assumptions made. See the cited article for details.

Examples

```
# Examples from Smith and VanderWeele 2019

# Zika virus example
selection_evalue(OR(73.1, rare = TRUE), lo = 13.0)

# Endometrial cancer example
selection_evalue(OR(2.30, rare = TRUE), true = 11.98, S_eq_U = TRUE, risk_inc = TRUE)

# Obesity paradox example
selection_evalue(RR(1.50), lo = 1.22, sel_pop = TRUE)
```

---

sens_plot *Plots for sensitivity analyses*

Description

Produces line plots (type="line") showing the average bias factor across studies on the relative risk
(RR) scale vs. the estimated proportion of studies with true RRs above or below a chosen threshold
q. The plot secondarily includes a X-axis showing the minimum strength of confounding to produce
the given bias factor. The shaded region represents a pointwise confidence band. Alternatively,
produces distribution plots (type="dist") for a specific bias factor showing the observed and true
distributions of RRs with a red line marking exp(q).

Usage

```
sens_plot(
  method = "calibrated",
  type,
  q,
  CI.level = 0.95,
  tail = NA,
  muB.toward.null = FALSE,
  give.CI = TRUE,
  Bmin = 0,
  Bmax = log(4),
  breaks.x1 = NA,
  breaks.x2 = NA,
  muB,
  sigB,
  yr,
  vyr = NA,
  t2,
  vt2 = NA,
  R = 1000,
  dat = NA,
  yi.name = NA,
  vi.name = NA
)
```

Arguments

method	"calibrated" or "parametric". See Details.
type	dist for distribution plot; line for line plot (see Details)
q	True causal effect size chosen as the threshold for a meaningfully large effect
CI.level	Pointwise confidence level as a proportion (e.g., 0.95).

tail	"above" for the proportion of effects above q; "below" for the proportion of effects below q. By default, is set to "above" if the pooled point estimate (method = "parametric") or median of the calibrated estimates (method = "calibrated") is above 1 on the relative risk scale and is set to "below" otherwise.
muB.toward.null	
	Whether you want to consider bias that has on average shifted studies' point estimates away from the null (FALSE; the default) or that has on average shifted studies' point estimates toward the null (TRUE). See Details.
give.CI	Logical. If TRUE, a pointwise confidence intervals is plotted.
Bmin	Lower limit of lower X-axis on the log scale (only needed if type = "line").
Bmax	Upper limit of lower X-axis on the log scale (only needed if type = "line")
breaks.x1	Breaks for lower X-axis (bias factor) on RR scale. (optional for type = "line"; not used for type = "dist").
breaks.x2	Breaks for upper X-axis (confounding strength) on RR scale (optional for type = "line"; not used for type = "dist")
muB	Single mean bias factor on log scale (only needed if type = "dist")
sigB	Standard deviation of log bias factor across studies (only used if method = "parametric")
yr	Pooled point estimate (on log scale) from confounded meta-analysis (only used if method = "parametric")
vyr	Estimated variance of pooled point estimate from confounded meta-analysis (only used if method = "parametric")
t2	Estimated heterogeneity (τ^2) from confounded meta-analysis (only used if method = "parametric")
vt2	Estimated variance of τ^2 from confounded meta-analysis (only used if method = "parametric")
R	Number of bootstrap iterates for confidence interval estimation. Only used if method = "calibrated" and give.CI = TRUE.
dat	Dataframe containing studies' point estimates and variances. Only used if method = "calibrated".
yi.name	Name of variable in dat containing studies' point estimates. Only used if method = "calibrated".
vi.name	Name of variable in dat containing studies' variance estimates. Only used if method = "calibrated".

Details

This function calls confounded_meta to get the point estimate and confidence interval at each value of the bias factor. See ?confounded_meta for details.

Note that Bmin and Bmax are specified on the log scale for consistency with the muB argument and with the function confounded_meta, whereas breaks.x1 and breaks.x2 are specified on the relative risk scale to facilitate adjustments to the plot appearance.

References

Mathur MB & VanderWeele TJ (2020a). Sensitivity analysis for unmeasured confounding in meta-analyses. *Journal of the American Statistical Association*.

Mathur MB & VanderWeele TJ (2020b). Robust metrics and sensitivity analyses for meta-analyses of heterogeneous effects. *Epidemiology*.

Wang C-C & Lee W-C (2019). A simple method to estimate prediction intervals and predictive distributions: Summarizing meta-analyses beyond means and confidence intervals. *Research Synthesis Methods*.

Examples

```
##### Example 1: Calibrated Line Plots #####

# simulated dataset with exponentially distributed
#  population effects
# we will use the calibrated method to avoid normality assumption
data(toyMeta)

# without confidence band
sens_plot( method = "calibrated",
          type="line",
          q=log(.9),
          tail = "below",
          dat = toyMeta,
          yi.name = "est",
          vi.name = "var",
          give.CI = FALSE )


# # with confidence band and a different threshold, q
# # commented out because takes a while too run
# sens_plot( method = "calibrated",
#          type="line",
#          q=0,
#          tail = "below",
#          dat = toyMeta,
#          yi.name = "est",
#          vi.name = "var",
#          give.CI = TRUE,
#          R = 300 ) # should be higher in practice


##### Example 2: Calibrated and Parametric Line Plots #####

# example dataset
d = metafor::escalc(measure="RR",
                    ai=tpos,
                    bi=tneg,
                    ci=cpos,
                    di=cneg,
                    data=metadat::dat.bcg)
```

```
# without confidence band
sens_plot( method = "calibrated",
           type="line",
           tail = "below",
           q=log(1.1),
           dat = d,
           yi.name = "yi",
           vi.name = "vi",
           give.CI = FALSE )

# # with confidence band
# # commented out because it takes a while
# # this example gives bootstrap warnings because of its small sample size
# sens_plot( method = "calibrated",
#            type="line",
#            q=log(1.1),
#            R = 500,  # should be higher in practice (e.g., 1000)
#            dat = d,
#            yi.name = "yi",
#            vi.name = "vi",
#            give.CI = TRUE )


# now with heterogeneous bias across studies (via sigB) and with confidence band
sens_plot( method = "parametric",
           type="line",
           q=log(1.1),
           yr=log(1.3),
           vyr = .05,
           vt2 = .001,
           t2=0.4,
           sigB = 0.1,
           Bmin=0,
           Bmax=log(4) )

##### Distribution Line Plot #####

# distribution plot: apparently causative
sens_plot( type="dist",
           q=log(1.1),
           muB=log(2),
           sigB = 0.1,
           yr=log(1.3),
           t2=0.4 )

# distribution plot: apparently preventive
sens_plot( type="dist",
           q=log(0.90),
           muB=log(1.5),
           sigB = 0.1,
           yr=log(0.7),
           t2=0.2 )
```

---

soyMeta   *A meta-analysis on soy intake and breast cancer risk (Trock et al.,*
                                 2006)

Description

A meta-analysis of observational studies (12 case-control and six cohort or nested case-control) on
the association of soy-food intake with breast cancer risk. Data are from Trock et al.'s (2006) Table
1. This dataset was used as the applied example in Mathur & VanderWeele (2020a).

Usage

    soyMeta

Format

An object of class data.frame with 20 rows and 3 columns.

Details

The variables are as follows:

- author Last name of the study's first author.
- est Point estimate on the log-relative risk or log-odds ratio scale.
- var Variance of the log-relative risk or log-odds ratio.

References

Trock BJ, Hilakivi-Clarke L, Clark R (2006). Meta-analysis of soy intake and breast cancer risk.
Journal of the National Cancer Institute.

Mathur MB & VanderWeele TJ (2020a). Sensitivity analysis for unmeasured confounding in meta-
analyses. *Journal of the American Statistical Association*.

---

svalues.HR   *Compute selection bias E-value for an estimate and its confidence in-*
                                 terval limits

Description

Returns a data frame containing point estimates, the lower confidence limit, and the upper confi-
dence limit on the risk ratio scale (through an approximate conversion if needed when outcome is
common) as well as selection bias E-values for the point estimate and the confidence interval limit
closer to the null.

Usage

```
svalues.HR(
  est,
  lo = NA,
  hi = NA,
  rare = NA,
  true = 1,
  sel_pop = FALSE,
  S_eq_U = FALSE,
  risk_inc = FALSE,
  risk_dec = FALSE,
  ...
)
```

Arguments

est	The point estimate
lo	The lower limit of the confidence interval
hi	The upper limit of the confidence interval
rare	1 if outcome is rare (<15 percent at end of follow-up); 0 if outcome is not rare (>15 percent at end of follow-up)
true	The true HR to which to shift the observed point estimate. Typically set to 1 to consider a null true effect.
sel_pop	Whether inference is specific to selected population (TRUE) or entire population (FALSE). Defaults to FALSE.
S_eq_U	Whether the unmeasured factor is assumed to be a defining characteristic of the selected population. Defaults to FALSE.
risk_inc	Whether selection is assumed to be associated with increased risk of the outcome in both exposure groups. Defaults to FALSE.
risk_dec	Whether selection is assumed to be associated with decreased risk of the outcome in both exposure groups. Defaults to FALSE.
...	Arguments passed to other methods.

Details

A selection bias E-value is a summary measure that helps assess susceptibility of a result to selection bias. Each of one or more parameters characterizing the extent of the bias must be greater than or equal to this value to be sufficient to shift an estimate (est) to the null or other true value (true). The parameters, as defined in Smith and VanderWeele 2019, depend on assumptions an investigator is willing to make (see arguments sel_pop, S_eq_U, risk_inc, risk_dec). The svalues.XX functions print a message about which parameters the selection bias E-value refers to given the assumptions made. See the cited article for details.

Examples

```
# Examples from Smith and VanderWeele 2019

# Obesity paradox example
svalues.RR(est = 1.50, lo = 1.22, sel_pop = TRUE)
```

---

svalues.OR	*Compute selection bias E-value for an odds ratio and its confidence interval limits*

Description

Returns a data frame containing point estimates, the lower confidence limit, and the upper confidence limit on the risk ratio scale (through an approximate conversion if needed when outcome is common) as well as E-values for the point estimate and the confidence interval limit closer to the null.

Usage

```
svalues.OR(
  est,
  lo = NA,
  hi = NA,
  rare = NA,
  true = 1,
  sel_pop = FALSE,
  S_eq_U = FALSE,
  risk_inc = FALSE,
  risk_dec = FALSE,
  ...
)
```

Arguments

est	The point estimate
lo	The lower limit of the confidence interval
hi	The upper limit of the confidence interval
rare	1 if outcome is rare (<15 percent at end of follow-up); 0 if outcome is not rare (>15 percent at end of follow-up)
true	The true OR to which to shift the observed point estimate. Typically set to 1 to consider a null true effect.
sel_pop	Whether inference is specific to selected population (TRUE) or entire population (FALSE). Defaults to FALSE.
S_eq_U	Whether the unmeasured factor is assumed to be a defining characteristic of the selected population. Defaults to FALSE.

risk_inc	Whether selection is assumed to be associated with increased risk of the outcome in both exposure groups. Defaults to FALSE.
risk_dec	Whether selection is assumed to be associated with decreased risk of the outcome in both exposure groups. Defaults to FALSE.
...	Arguments passed to other methods.

Details

A selection bias E-value is a summary measure that helps assess susceptibility of a result to selection bias. Each of one or more parameters characterizing the extent of the bias must be greater than or equal to this value to be sufficient to shift an estimate (est) to the null or other true value (true). The parameters, as defined in Smith and VanderWeele 2019, depend on assumptions an investigator is willing to make (see arguments sel_pop, S_eq_U, risk_inc, risk_dec). The svalues.XX functions print a message about which parameters the selection bias E-value refers to given the assumptions made. See the cited article for details.

Examples

```
# Examples from Smith and VanderWeele 2019

# Zika virus example
svalues.OR(est = 73.1, rare = TRUE, lo = 13.0)

# Endometrial cancer example
svalues.OR(est = 2.30, rare = TRUE, true = 11.98, S_eq_U = TRUE, risk_inc = TRUE)
```

| svalues.RR | *Compute selection bias E-value for a risk ratio or rate ratio and its confidence interval limits* |

Description

Returns a data frame containing point estimates, the lower confidence limit, and the upper confidence limit for the risk ratio (as provided by the user) as well as selection bias E-values for the point estimate and the confidence interval limit closer to the null.

Usage

```
svalues.RR(
  est,
  lo = NA,
  hi = NA,
  true = 1,
  sel_pop = FALSE,
  S_eq_U = FALSE,
  risk_inc = FALSE,
  risk_dec = FALSE,
  ...
)
```

Arguments

`est`	The point estimate
`lo`	The lower limit of the confidence interval
`hi`	The upper limit of the confidence interval
`true`	The true RR to which to shift the observed point estimate. Typically set to 1 to consider a null true effect.
`sel_pop`	Whether inference is specific to selected population (TRUE) or entire population (FALSE). Defaults to FALSE.
`S_eq_U`	Whether the unmeasured factor is assumed to be a defining characteristic of the selected population. Defaults to FALSE.
`risk_inc`	Whether selection is assumed to be associated with increased risk of the outcome in both exposure groups. Defaults to FALSE.
`risk_dec`	Whether selection is assumed to be associated with decreased risk of the outcome in both exposure groups. Defaults to FALSE.
`...`	Arguments passed to other methods.

Details

A selection bias E-value is a summary measure that helps assess susceptibility of a result to selection bias. Each of one or more parameters characterizing the extent of the bias must be greater than or equal to this value to be sufficient to shift an estimate (`est`) to the null or other true value (`true`). The parameters, as defined in Smith and VanderWeele 2019, depend on assumptions an investigator is willing to make (see arguments `sel_pop`, `S_eq_U`, `risk_inc`, `risk_dec`). The `svalues.XX` functions print a message about which parameters the selection bias E-value refers to given the assumptions made. See the cited article for details.

Examples

```
# Examples from Smith and VanderWeele 2019

# Zika virus example
svalues.RR(est = 73.1, lo = 13.0)

# Endometrial cancer example
svalues.RR(est = 2.30, true = 11.98, S_eq_U = TRUE, risk_inc = TRUE)

# Obesity paradox example
svalues.RR(est = 1.50, lo = 1.22, sel_pop = TRUE)
```

---

  toyMeta   *An example meta-analysis*

Description

A simple simulated meta-analysis of 50 studies with exponentially distributed population effects.

Usage

```
toyMeta
```

Format

An object of class data.frame with 50 rows and 2 columns.

Details

The variables are as follows:

- est Point estimate on the log-relative risk scale.
- var Variance of the log-relative risk.

---

twoXtwoRR   *Estimate risk ratio and compute CI limits from two-by-two table*

Description

Given counts in a two-by-two table, computes risk ratio and confidence interval limits.

Usage

```
twoXtwoRR(n11, n10, n01, n00, alpha = 0.05)
```

Arguments

n11	Number exposed (X=1) and diseased (D=1)
n10	Number exposed (X=1) and not diseased (D=0)
n01	Number unexposed (X=0) and diseased (D=1)
n00	Number unexposed (X=0) and not diseased (D=0)
alpha	Alpha level associated with confidence interval

Examples

```
# Hammond and Holl (1958 JAMA) Data
# Two by Two Table
#          Lung Cancer    No Lung Cancer
# Smoker    397              78557
# Nonsmoker 51               108778

twoXtwoRR(397, 78557, 51, 108778)
```

Index