# Package ‘EValue’

June 16, 2020

**Type** Package

**Title** Sensitivity Analyses for Unmeasured Confounding or Selection Bias in Observational Studies and Meta-Analyses

**Version** 3.0.0

**Description** Conducts sensitivity analyses for unmeasured confounding for either an observational study or a meta-analysis of observational studies. For a single observational study, the package reports E-values, defined as the minimum strength of association on the risk ratio scale that an unmeasured confounder 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. You can use one of the evalues.XX() functions to compute E-values for the relevant outcome types. Outcome types include risk ratios, odds ratio with common or rare outcomes, hazard ratios with common or rare outcomes, and standardized differences in outcomes. Optionally, you can use the bias_plot() function to plot the bias factor as a function of two sensitivity parameters. (See VanderWeele & Ding, 2017 [<http://annals.org/aim/article/2643434>] for details.) For a meta-analysis, use the function confounded_meta to compute point estimates and inference for: (1) the proportion of studies with true causal effect sizes more extreme than a specified threshold of scientific importance; and (2) the minimum bias factor and confounding strength required to reduce to less than a specified threshold the proportion of studies with true effect sizes of scientifically significant size. The functions sens_plot() and sens_table() create plots and tables for visualizing these meta-analysis metrics across a range of bias values. (See Mathur & VanderWeele, 2019 [<https://amstat.tandfonline.com/doi/full/10.1080/01621459.2018.1529598#.XKIJtOtKjdc>] for details.) Most of the analyses available in this package can also be conducted using web-based graphical interfaces (for a single observational study: [<https://evalue.hmdc.harvard.edu>]; for a meta-analysis: [<https://mmathur.shinyapps.io/meta_gui_2/>]).

**LazyData** true

**License** GPL-2

**Imports** stats, graphics, ggplot2 (>= 2.2.1), metafor, msm, devtools, methods

**RoxygenNote** 7.1.0

**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>

**Repository** CRAN

**Date/Publication** 2020-06-16 14:10:23 UTC

# R topics documented:

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 *Estimates and inference for sensitivity analyses*

### Description

Computes point estimates, standard errors, and confidence interval bounds for (1) prop, the proportion of studies with true effect sizes above q (or below q for an apparently preventive yr) 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 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(
  q,
  r = NA,
  muB = NA,
  sigB = 0,
  yr,
  vyr = NA,
  t2,
  vt2 = NA,
  CI.level = 0.95,
  tail = NA
)
```

### Arguments

q — True effect size that is the threshold for "scientific significance"

r — For Tmin and Gmin, value to which the proportion of large effect sizes is to be reduced

muB — Mean bias factor on the log scale across studies

sigB — Standard deviation of log bias factor across studies

yr — Pooled point estimate (on log scale) from confounded meta-analysis

vyr — Estimated variance of pooled point estimate from confounded meta-analysis

t2 — Estimated heterogeneity (tau^2) from confounded meta-analysis

vt2 — Estimated variance of tau^2 from confounded meta-analysis

CI.level — Confidence level as a proportion

tail — above for the proportion of effects above q; below for the proportion of effects below q. By default, is set to above for relative risks above 1 and to below for relative risks below 1.

### Details

To compute all three point estimates (prop,Tmin,and Gmin) and inference, all arguments must be non-NA. To compute only a point estimate for prop, arguments r,vyr, and vt2 can be left NA. To compute only point estimates for Tmin and Gmin, arguments muB,vyr, and vt2 can be left NA. To compute inference for all point estimates, vyr and vt2 must be supplied.

### Examples

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

m = metafor::rma.uni(yi= d$yi, vi=d$vi, knha=FALSE,
                     measure="RR", method="DL" )
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
confounded_meta( q=log(0.90), r=0.20, 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( q=log(0.90), muB=log(1.5),
                 yr=yr, t2=t2, CI.level=0.95 )

# passing only arguments needed for Tmin, Gmin point estimates
confounded_meta( q=log(0.90), r=0.20,
                 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 an unmeasured confounder 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).

## 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.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 correspond 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 bias factor on the relative risk (RR) scale vs. the proportion of studies with true RRs above `q` (or below it for an apparently preventive relative risk). The plot secondarily includes a X-axis scaled based on the minimum strength of confounding to produce the given bias factor. The shaded region represents a 95% 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(
  type,
  q,
  muB = NA,
  Bmin = log(1),
  Bmax = log(5),
  sigB = 0,
  yr,
  vyr = NA,
  t2,
  vt2 = NA,
  breaks.x1 = NA,
  breaks.x2 = NA,
  CI.level = 0.95
)
```

## Arguments

`type` `dist` for distribution plot; `line` for line plot (see Details)

`q` True effect size that is the threshold for "scientific significance"

`muB` Single mean bias factor on log scale (only needed for distribution plot)

`Bmin` Lower limit of lower X-axis on the log scale (only needed for line plot)

`Bmax` Upper limit of lower X-axis on the log scale (only needed for line plot)

`sigB` Standard deviation of log bias factor across studies (length 1)

`yr` Pooled point estimate (on log scale) from confounded meta-analysis

`vyr` Estimated variance of pooled point estimate from confounded meta-analysis

`t2` Estimated heterogeneity (tau^2) from confounded meta-analysis

`vt2` Estimated variance of tau^2 from confounded meta-analysis

`breaks.x1` Breaks for lower X-axis (bias factor) on RR scale (optional for line plot; not used for distribution plot)

`breaks.x2` Breaks for upper X-axis (confounding strength) on RR scale (optional for line plot; not used for distribution plot)

`CI.level` Pointwise confidence level as a proportion

## Details

Arguments `vyr` and `vt2` can be left `NA`, in which case no confidence band will appear on the line plot.

## Examples

```
# with variable bias and with confidence band
sens_plot( type="line", q=log(1.1), Bmin=log(1), Bmax=log(4), sigB=0.1,
           yr=log(1.3), vyr=0.005, t2=0.4, vt2=0.03 )

# with fixed bias and without confidence band
sens_plot( type="line", q=log(1.1), Bmin=log(1), Bmax=log(4),
           yr=log(1.3), t2=0.4 )

# apparently preventive
sens_plot( type="line", q=log(0.90), Bmin=log(1), Bmax=log(4),
           yr=log(0.6), vyr=0.005, t2=0.4, vt2=0.04 )

# distribution plot: apparently causative
# commented out because takes 5-10 seconds to run
# sens_plot( type="dist", q=log(1.1), muB=log(2),
#            yr=log(1.3), t2=0.4 )
# distribution plot: apparently preventive
# commented out because takes 5-10 seconds to run
# sens_plot( type="dist", q=log(0.90), muB=log(1.5),
#           yr=log(0.7), t2=0.2 )
```

## sens_table *Tables for sensitivity analyses*

### Description

Produces table showing the proportion of true effect sizes more extreme than `q` across a grid of bias parameters `muB` and `sigB` (for `meas == "prop"`). Alternatively, produces a table showing the minimum bias factor (for `meas == "Tmin"`) or confounding strength (for `meas == "Gmin"`) required to reduce to less than `r` the proportion of true effects more extreme than `q`.

### Usage

```
sens_table(meas, q, r = seq(0.1, 0.9, 0.1), muB = NA, sigB = NA, yr, t2)
```

### Arguments

`meas` prop, Tmin, or Gmin

`q` True effect size that is the threshold for "scientific significance"

`r` For `Tmin` and `Gmin`, vector of values to which the proportion of large effect sizes is to be reduced

`muB` Mean bias factor on the log scale across studies

`sigB` Standard deviation of log bias factor across studies

`yr` Pooled point estimate (on log scale) from confounded meta-analysis

`t2` Estimated heterogeneity (tau^2) from confounded meta-analysis

### Details

For `meas=="Tmin"` or `meas=="Gmin"`, arguments `muB` and `sigB` can be left `NA`; `r` can also be `NA` as it will default to a reasonable range of proportions. Returns a `data.frame` whose rows are values of `muB` (for `meas=="prop"`) or of `r` (for `meas=="Tmin"` or `meas=="Gmin"`). Its columns are values of `sigB` (for `meas=="prop"`) or of `q` (for `meas=="Tmin"` or `meas=="Gmin"`). Tables for `Gmin` will display `NaN` for cells corresponding to `Tmin`<1, i.e., for which no bias is required to reduce the effects as specified.

### Examples

```
sens_table( meas="prop", q=log(1.1), muB=c( log(1.1),
log(1.5), log(2.0) ), sigB=c(0, 0.1, 0.2),
yr=log(2.5), t2=0.1 )

sens_table( meas="Tmin", q=c( log(1.1), log(1.5) ),
yr=log(1.3), t2=0.1 )

# Tmin is 1 here because we already have <80% of effects
#  below log(1.1) even without any confounding
sens_table( meas="Gmin", r=0.8, q=c( log(1.1) ),
yr=log(1.3), t2=0.1 )
```

## stronger_than *Estimate proportion of population effect sizes above or below a threshold*

### Description

Estimates the proportion of true effect sizes in a meta-analysis above or below a specified threshold of scientific importance. Effect sizes may be of any type (they need not be relative risks). This is a wrapper for `confounded_meta`; it is the special case in which there is no unmeasured confounding.

### Usage

```
stronger_than(q, yr, vyr = NA, t2, vt2 = NA, CI.level = 0.95, tail)
```

### Arguments

| | |
|---|---|
| `q` | True effect size that is the threshold for "scientific importance" |
| `yr` | Pooled point estimate from meta-analysis |
| `vyr` | Estimated variance of pooled point estimate from meta-analysis |
| `t2` | Estimated heterogeneity (tau^2) from meta-analysis |
| `vt2` | Estimated variance of tau^2 from meta-analysis |
| `CI.level` | Confidence level as a proportion |
| `tail` | `above` for the proportion of effects above `q`; `below` for the proportion of effects below `q`. |

## svalues.HR *Compute selection bias E-value for an estimate 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 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)
```

## 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