Package 'tipr'

February 6, 2024

Type Package

Title Tipping Point Analyses

Version 1.0.2

Description The strength of evidence provided by epidemiological and observational studies is inherently limited by the potential for unmeasured confounding. We focus on three key quantities: the observed bound of the confidence interval closest to the null, the relationship between an unmeasured confounder and the outcome, for example a plausible residual effect size for an unmeasured continuous or binary confounder, and the relationship between an unmeasured confounder and the exposure, for example a realistic mean difference or prevalence difference for this hypothetical confounder between exposure groups. Building on the methods put forth by Cornfield et al. (1959), Bross (1966), Schlesselman (1978), Rosenbaum & Rubin (1983), Lin et al. (1998), Lash et al. (2009), Rosenbaum (1986), Cinelli & Hazlett (2020), VanderWeele & Ding (2017), and Ding & VanderWeele (2016), we can use these quantities to assess how an unmeasured confounder may tip our result to insignificance.

License MIT + file LICENSE

URL https://r-causal.github.io/tipr/, https://github.com/r-causal/tipr

BugReports https://github.com/r-causal/tipr/issues

Depends R (>= 2.10)

Imports cli (>= 3.4.1), glue, purrr, rlang (>= 1.0.6), sensemakr, tibble

Suggests broom, dplyr, MASS, testthat

Encoding UTF-8

LazyData true

RoxygenNote 7.3.1

NeedsCompilation no

Author Lucy D'Agostino McGowan [aut, cre] (<https://orcid.org/0000-0002-6983-2759>), Malcolm Barrett [aut] (<https://orcid.org/0000-0003-0299-5825>)

Maintainer Lucy D'Agostino McGowan <lucydagostino@gmail.com>

Repository CRAN

Date/Publication 2024-02-06 19:30:02 UTC

R topics documented:

adjust_coef . . . 2
adjust_coef_with_binary . . . 3
adjust_coef_with_r2 . . . 5
adjust_hr . . . 6
adjust_hr_with_binary . . . 7
adjust_or . . . 8
adjust_or_with_binary . . . 9
adjust_rr . . . 10
adjust_rr_with_binary . . . 11
exdata_continuous . . . 12
exdata_rr . . . 13
e_value . . . 13
observed_bias_order . . . 14
observed_bias_tbl . . . 14
observed_bias_tip . . . 15
observed_covariate_e_value . . . 16
r_value . . . 17
tip . . . 18
tip_coef . . . 19
tip_coef_with_r2 . . . 21
tip_hr . . . 22
tip_hr_with_binary . . . 23
tip_or . . . 24
tip_or_with_binary . . . 26
tip_rr . . . 27
tip_rr_with_binary . . . 28
tip_with_binary . . . 29

Index **32**

`adjust_coef` *Adjust an observed regression coefficient for a normally distributed confounder*

Description

Adjust an observed regression coefficient for a normally distributed confounder

Usage

```
adjust_coef(
  effect_observed,
  exposure_confounder_effect,
  confounder_outcome_effect,
  verbose = getOption("tipr.verbose", TRUE)
)

adjust_coef_with_continuous(
  effect_observed,
  exposure_confounder_effect,
  confounder_outcome_effect,
  verbose = getOption("tipr.verbose", TRUE)
)
```

Arguments

effect_observed	Numeric. Observed exposure - outcome effect from a regression model. This can be the beta coefficient, the lower confidence bound of the beta coefficient, or the upper confidence bound of the beta coefficient.
exposure_confounder_effect	Numeric. Estimated difference in scaled means between the unmeasured confounder in the exposed population and unexposed population
confounder_outcome_effect	Numeric. Estimated relationship between the unmeasured confounder and the outcome.
verbose	Logical. Indicates whether to print informative message. Default: TRUE

Value

Data frame.

Examples

```
## Update an observed coefficient of 0.5 with an unmeasured confounder
## with a difference in scaled means between exposure groups of 0.2
## and coefficient of 0.3
adjust_coef(0.5, 0.2, 0.3)
```

---

adjust_coef_with_binary

Adjust an observed coefficient from a regression model with a binary confounder

---

Description

Adjust an observed coefficient from a regression model with a binary confounder

Usage

```
adjust_coef_with_binary(
  effect_observed,
  exposed_confounder_prev,
  unexposed_confounder_prev,
  confounder_outcome_effect,
  loglinear = FALSE,
  verbose = getOption("tipr.verbose", TRUE)
)
```

Arguments

`effect_observed`	Numeric. Observed exposure - outcome effect from a loglinear model. This can be the beta coefficient, the lower confidence bound of the beta coefficient, or the upper confidence bound of the beta coefficient.
`exposed_confounder_prev`	Numeric between 0 and 1. Estimated prevalence of the unmeasured confounder in the exposed population
`unexposed_confounder_prev`	Numeric between 0 and 1. Estimated prevalence of the unmeasured confounder in the unexposed population
`confounder_outcome_effect`	Numeric. Estimated relationship between the unmeasured confounder and the outcome.
`loglinear`	Logical. Calculate the adjusted coefficient from a loglinear model instead of a linear model (the default). When `loglinear = FALSE`, `adjust_coef_with_binary()` is equivalent to `adjust_coef()` where `exposure_confounder_effect` is the difference in prevalences.
`verbose`	Logical. Indicates whether to print informative message. Default: `TRUE`

Value

Data frame.

Examples

```
adjust_coef_with_binary(1.1, 0.5, 0.3, 1.3)
```

`adjust_coef_with_r2` *Adjust a regression coefficient using the partial R2 for an unmeasured confounder-exposure relationship and unmeasured confounder- outcome relationship*

Description

This function wraps the `sensemakr::adjusted_estimate()` and `sensemakr::adjusted_se()` functions.

Usage

```
adjust_coef_with_r2(
  effect_observed,
  se,
  df,
  confounder_exposure_r2,
  confounder_outcome_r2,
  verbose = getOption("tipr.verbose", TRUE),
  alpha = 0.05,
  ...
)
```

Arguments

`effect_observed`	Numeric. Observed exposure - outcome effect from a regression model. This is the point estimate (beta coefficient)
`se`	Numeric. Standard error of the `effect_observed` in the previous parameter.
`df`	Numeric positive value. Residual degrees of freedom for the model used to estimate the observed exposure - outcome effect. This is the total number of observations minus the number of parameters estimated in your model. Often for models estimated with an intercept this is N - k - 1 where k is the number of predictors in the model.
`confounder_exposure_r2`	Numeric value between 0 and 1. The assumed partial R2 of the unobserved confounder with the exposure given the measured covariates.
`confounder_outcome_r2`	Numeric value between 0 and 1. The assumed partial R2 of the unobserved confounder with the outcome given the exposure and the measured covariates.
`verbose`	Logical. Indicates whether to print informative message. Default: `TRUE`
`alpha`	Significance level. Default = `0.05`.
`...`	Optional arguments passed to the `sensemakr::adjusted_estimate()` function.

Value

A data frame.

References

Carlos Cinelli, Jeremy Ferwerda and Chad Hazlett (2021). sensemakr: Sensitivity Analysis Tools for Regression Models. R package version 0.1.4. https://CRAN.R-project.org/package=sensemakr

Examples

```
adjust_coef_with_r2(0.5, 0.1, 102, 0.05, 0.1)
```

`adjust_hr` *Adjust an observed hazard ratio for a normally distributed confounder*

Description

Adjust an observed hazard ratio for a normally distributed confounder

Usage

```
adjust_hr(
  effect_observed,
  exposure_confounder_effect,
  confounder_outcome_effect,
  verbose = getOption("tipr.verbose", TRUE),
  hr_correction = FALSE
)

adjust_hr_with_continuous(
  effect_observed,
  exposure_confounder_effect,
  confounder_outcome_effect,
  verbose = getOption("tipr.verbose", TRUE),
  hr_correction = FALSE
)
```

Arguments

`effect_observed`
: Numeric positive value. Observed exposure - outcome hazard ratio. This can be the point estimate, lower confidence bound, or upper confidence bound.

`exposure_confounder_effect`
: Numeric. Estimated difference in scaled means between the unmeasured confounder in the exposed population and unexposed population

confounder_outcome_effect
: Numeric. Estimated relationship between the unmeasured confounder and the outcome.

verbose
: Logical. Indicates whether to print informative message. Default: TRUE

hr_correction
: Logical. Indicates whether to use a correction factor. The methods used for this function are based on risk ratios. For rare outcomes, a hazard ratio approximates a risk ratio. For common outcomes, a correction factor is needed. If you have a common outcome (>15%), set this to TRUE. Default: FALSE.

Value

Data frame.

Examples

```
adjust_hr(0.9, -0.9, 1.3)
```

---

adjust_hr_with_binary *Adjust an observed hazard ratio with a binary confounder*

Description

Adjust an observed hazard ratio with a binary confounder

Usage

```
adjust_hr_with_binary(
  effect_observed,
  exposed_confounder_prev,
  unexposed_confounder_prev,
  confounder_outcome_effect,
  verbose = getOption("tipr.verbose", TRUE),
  hr_correction = FALSE
)
```

Arguments

effect_observed
: Numeric positive value. Observed exposure - outcome hazard ratio. This can be the point estimate, lower confidence bound, or upper confidence bound.

exposed_confounder_prev
: Numeric between 0 and 1. Estimated prevalence of the unmeasured confounder in the exposed population

unexposed_confounder_prev
: Numeric between 0 and 1. Estimated prevalence of the unmeasured confounder in the unexposed population

confounder_outcome_effect
: Numeric positive value. Estimated relationship between the unmeasured confounder and the outcome

verbose
: Logical. Indicates whether to print informative message. Default: TRUE

hr_correction
: Logical. Indicates whether to use a correction factor. The methods used for this function are based on risk ratios. For rare outcomes, a hazard ratio approximates a risk ratio. For common outcomes, a correction factor is needed. If you have a common outcome (>15%), set this to TRUE. Default: FALSE.

Value

Data frame.

Examples

```
adjust_hr_with_binary(0.8, 0.1, 0.5, 1.8)
```

---

adjust_or *Adjust an observed odds ratio for a normally distributed confounder*

Description

Adjust an observed odds ratio for a normally distributed confounder

Usage

```
adjust_or(
  effect_observed,
  exposure_confounder_effect,
  confounder_outcome_effect,
  verbose = getOption("tipr.verbose", TRUE),
  or_correction = FALSE
)

adjust_or_with_continuous(
  effect_observed,
  exposure_confounder_effect,
  confounder_outcome_effect,
  verbose = getOption("tipr.verbose", TRUE),
  or_correction = FALSE
)
```

Arguments

effect_observed
: Numeric positive value. Observed exposure - outcome odds ratio. This can be the point estimate, lower confidence bound, or upper confidence bound.

`exposure_confounder_effect`
: Numeric. Estimated difference in scaled means between the unmeasured confounder in the exposed population and unexposed population

`confounder_outcome_effect`
: Numeric. Estimated relationship between the unmeasured confounder and the outcome.

`verbose`
: Logical. Indicates whether to print informative message. Default: `TRUE`

`or_correction`
: Logical. Indicates whether to use a correction factor. The methods used for this function are based on risk ratios. For rare outcomes, an odds ratio approximates a risk ratio. For common outcomes, a correction factor is needed. If you have a common outcome (>15%), set this to `TRUE`. Default: `FALSE`.

Value

Data frame.

Examples

```
adjust_or(1.2, 0.9, 1.3)
```

`adjust_or_with_binary` *Adjust an observed odds ratio with a binary confounder*

Description

Adjust an observed odds ratio with a binary confounder

Usage

```
adjust_or_with_binary(
  effect_observed,
  exposed_confounder_prev,
  unexposed_confounder_prev,
  confounder_outcome_effect,
  verbose = getOption("tipr.verbose", TRUE),
  or_correction = FALSE
)
```

Arguments

`effect_observed`
: Numeric positive value. Observed exposure - outcome odds ratio. This can be the point estimate, lower confidence bound, or upper confidence bound.

`exposed_confounder_prev`
: Numeric between 0 and 1. Estimated prevalence of the unmeasured confounder in the exposed population

unexposed_confounder_prev
: Numeric between 0 and 1. Estimated prevalence of the unmeasured confounder in the unexposed population

confounder_outcome_effect
: Numeric positive value. Estimated relationship between the unmeasured confounder and the outcome

verbose
: Logical. Indicates whether to print informative message. Default: TRUE

or_correction
: Logical. Indicates whether to use a correction factor. The methods used for this function are based on risk ratios. For rare outcomes, an odds ratio approximates a risk ratio. For common outcomes, a correction factor is needed. If you have a common outcome (>15%), set this to TRUE. Default: FALSE.

Value

Data frame.

Examples

```
adjust_or_with_binary(3, 1, 0, 3)
adjust_or_with_binary(3, 1, 0, 3, or_correction = TRUE)
```

---

adjust_rr *Adjust an observed risk ratio for a normally distributed confounder*

Description

Adjust an observed risk ratio for a normally distributed confounder

Usage

```
adjust_rr(
  effect_observed,
  exposure_confounder_effect,
  confounder_outcome_effect,
  verbose = TRUE
)

adjust_rr_with_continuous(
  effect_observed,
  exposure_confounder_effect,
  confounder_outcome_effect,
  verbose = TRUE
)
```

Arguments

`effect_observed`
: Numeric positive value. Observed exposure - outcome risk ratio. This can be the point estimate, lower confidence bound, or upper confidence bound.

`exposure_confounder_effect`
: Numeric. Estimated difference in scaled means between the unmeasured confounder in the exposed population and unexposed population

`confounder_outcome_effect`
: Numeric. Estimated relationship between the unmeasured confounder and the outcome.

`verbose`
: Logical. Indicates whether to print informative message. Default: `TRUE`

Value

Data frame.

Examples

```
adjust_rr(1.2, 0.5, 1.1)
```

`adjust_rr_with_binary` *Adjust an observed risk ratio with a binary confounder*

Description

Adjust an observed risk ratio with a binary confounder

Usage

```
adjust_rr_with_binary(
  effect_observed,
  exposed_confounder_prev,
  unexposed_confounder_prev,
  confounder_outcome_effect,
  verbose = getOption("tipr.verbose", TRUE)
)
```

Arguments

`effect_observed`
: Numeric positive value. Observed exposure - outcome risk ratio. This can be the point estimate, lower confidence bound, or upper confidence bound.

`exposed_confounder_prev`
: Numeric between 0 and 1. Estimated prevalence of the unmeasured confounder in the exposed population

unexposed_confounder_prev
: Numeric between 0 and 1. Estimated prevalence of the unmeasured confounder in the unexposed population

confounder_outcome_effect
: Numeric positive value. Estimated relationship between the unmeasured confounder and the outcome

verbose
: Logical. Indicates whether to print informative message. Default: TRUE

Value

Data frame.

Examples

```
adjust_rr_with_binary(1.1, 0.5, 0.3, 1.3)
```

---

exdata_continuous *Example Data (Continuous Outcome)*

Description

A data set simulated with two Normally distributed confounders, one "measured" and one "unmeasured", an exposure, and outcome. The "true" causal effect of the exposure on the outcome, accounting for both the measured and unmeasured confounders, should be 0.

Usage

```
exdata_continuous
```

Format

A data frame with 2,000 rows and 4 columns:

- `.unmeasured_confounder`: A simulated unmeasured confounder
- `measured_confounder`: A simulated measured confounder
- `exposure`
- `outcome`

exdata_rr *Example Data (Risk Ratio)*

Description

A data set simulated with two Normally distributed confounders, one "measured" and one "unmeasured", an exposure, and outcome. The "true" causal effect of the exposure on the outcome, accounting for both the measured and unmeasured confounders, should be 0.

Usage

```
exdata_rr
```

Format

A data frame with 2,000 rows and 4 columns:

- `.unmeasured_confounder`: A simulated unmeasured confounder
- `measured_confounder`: A simulated measured confounder
- `exposure`
- `outcome`

e_value *Calculate an E-value*

Description

Calculate an E-value

Usage

```
e_value(effect_observed)
```

Arguments

`effect_observed`
: Numeric positive value. Observed exposure - outcome effect (assumed to be the exponentiated coefficient, so a risk ratio, odds ratio, or hazard ratio). This can be the point estimate, lower confidence bound, or upper confidence bound.

Value

Numeric value

Examples

```
e_value(0.9)
e_value(1.3)
```

observed_bias_order *Order observed bias data frame for plotting*

Description

Order observed bias data frame for plotting

Usage

```
observed_bias_order(d, by)
```

Arguments

d	Observed bias data frame. Must have columns dropped and type
by	Character. Variable in d to order by.

Value

Data frame in the correct order

observed_bias_tbl *Create a data frame to assist with creating an observed bias plot*

Description

Create a data frame to assist with creating an observed bias plot

Usage

```
observed_bias_tbl(ps_mod, outcome_mod, drop_list = NULL)
```

Arguments

ps_mod	Model object for the propensity score model
outcome_mod	Model object for the outcome model
drop_list	Named list of covariates or groups of covariates to drop if NULL, will default to dropping each covariate one at a time.

Value

Data frame with the following columns:

- dropped: The covariate or group of covariates that were dropped
- type: Explanation of dropped, whether it refers to a single covariate (covariate) or a group of covariates (group)
- ps_formula: The new formula for the updated propensity score model
- outcome_formula: The new formula for the updated outcome model
- ps_model: The new model object for the updated propensity score model
- p: The updated propensity score

Examples

```
ps_mod <- glm(am ~ mpg + cyl + I(hp^2), data = mtcars)
outcome_mod <- lm(qsec ~ am + hp + disp + wt, data = mtcars)
observed_bias_tbl(
 ps_mod,
 outcome_mod,
 drop_list = list(
   group_one = c("mpg", "hp"),
   group_two = c("cyl", "wt")
   )
 )
```

---

observed_bias_tip *Create a data frame to combine with an observed bias data frame demonstrating a hypothetical unmeasured confounder*

Description

Create a data frame to combine with an observed bias data frame demonstrating a hypothetical unmeasured confounder

Usage

```
observed_bias_tip(
  tip,
  point_estimate,
  lb,
  ub,
  tip_desc = "Hypothetical unmeasured confounder"
)
```

Arguments

tip	Numeric. Value you would like to tip to.
point_estimate	Numeric. Result estimate from the full model.
lb	Numeric. Result lower bound from the full model.
ub	Numeric. Result upper bound from the full model.
tip_desc	Character. A description of the tipping point.

Value

A data frame with five columns:

- dropped: the input from tip_desc
- type: Explanation of dropped, here tip to clarify that this was calculated as a tipping point.
- point_estimate: the shifted point estimate
- lb: the shifted lower bound
- ub: the shifted upper bound

---

observed_covariate_e_value

Calculate the Observed Covariate E-value

Description

Calculate the Observed Covariate E-value

Usage

```
observed_covariate_e_value(lb, ub, lb_adj, ub_adj, transform = NULL)
```

Arguments

lb	Numeric. The lower bound of the full model
ub	Numeric. The upper bound of the full model
lb_adj	Numeric. The lower bound of the adjusted model
ub_adj	Numeric. The upper bound of the adjusted model
transform	Character. If your effect is an odds ratio or hazard ratio, this will perform the transformation suggested by VanderWeele and Ding. Allowed values are: • "OR" • "HR"

Value

The Observed Covariate E-value

r_value *Robustness value*

Description

This function wraps the sensemakr::robustness_value() function

Usage

```
r_value(effect_observed, se, df, ...)
```

Arguments

effect_observed	Numeric. Observed exposure - outcome effect from a regression model. This is the point estimate (beta coefficient)
se	Numeric. Standard error of the effect_observed in the previous parameter.
df	Numeric positive value. Residual degrees of freedom for the model used to estimate the observed exposure - outcome effect. This is the total number of observations minus the number of parameters estimated in your model. Often for models estimated with an intercept this is N - k - 1 where k is the number of predictors in the model.
...	Optional arguments passed to the sensemakr::robustness_value() function.

Value

Numeric. Robustness value

References

Carlos Cinelli, Jeremy Ferwerda and Chad Hazlett (2021). sensemakr: Sensitivity Analysis Tools for Regression Models. R package version 0.1.4. https://CRAN.R-project.org/package=sensemakr

Examples

```
r_value(0.5, 0.1, 102)
```

`tip` *Tip a result with a normally distributed confounder.*

Description

choose one of the following, and the other will be estimated:

- `exposure_confounder_effect`
- `confounder_outcome_effect`

Usage

```
tip(
  effect_observed,
  exposure_confounder_effect = NULL,
  confounder_outcome_effect = NULL,
  verbose = getOption("tipr.verbose", TRUE),
  correction_factor = "none"
)

tip_with_continuous(
  effect_observed,
  exposure_confounder_effect = NULL,
  confounder_outcome_effect = NULL,
  verbose = getOption("tipr.verbose", TRUE),
  correction_factor = "none"
)

tip_c(
  effect_observed,
  exposure_confounder_effect = NULL,
  confounder_outcome_effect = NULL,
  verbose = getOption("tipr.verbose", TRUE),
  correction_factor = "none"
)
```

Arguments

`effect_observed`
: Numeric positive value. Observed exposure - outcome effect (assumed to be the exponentiated coefficient, so a risk ratio, odds ratio, or hazard ratio). This can be the point estimate, lower confidence bound, or upper confidence bound.

`exposure_confounder_effect`
: Numeric. Estimated difference in scaled means between the unmeasured confounder in the exposed population and unexposed population

confounder_outcome_effect
: Numeric positive value. Estimated relationship between the unmeasured confounder and the outcome

verbose
: Logical. Indicates whether to print informative message. Default: TRUE

correction_factor
: Character string. Options are "none", "hr", "or". For common outcomes (>15%), the odds ratio or hazard ratio is not a good estimate for the risk ratio. In these cases, we can apply a correction factor. If you are supplying a hazard ratio for a common outcome, set this to "hr"; if you are supplying an odds ratio for a common outcome, set this to "or"; if you are supplying a risk ratio or your outcome is rare, set this to "none" (default).

Value

Data frame.

Examples

```
## to estimate the relationship between an unmeasured confounder and outcome
## needed to tip analysis
tip(1.2, exposure_confounder_effect = -2)

## to estimate the number of unmeasured confounders specified needed to tip
## the analysis
tip(1.2, exposure_confounder_effect = -2, confounder_outcome_effect = .99)

## Example with broom
if (requireNamespace("broom", quietly = TRUE) &&
    requireNamespace("dplyr", quietly = TRUE)) {
  glm(am ~ mpg, data = mtcars, family = "binomial") %>%
   broom::tidy(conf.int = TRUE, exponentiate = TRUE) %>%
   dplyr::filter(term == "mpg") %>%
   dplyr::pull(conf.low) %>%
   tip(confounder_outcome_effect = 2.5)
}
```

---

tip_coef *Tip a linear model coefficient with a continuous confounder.*

Description

choose one of the following, and the other will be estimated:

- exposure_confounder_effect
- confounder_outcome_effect

Usage

```
tip_coef(
  effect_observed,
  exposure_confounder_effect = NULL,
  confounder_outcome_effect = NULL,
  verbose = getOption("tipr.verbose", TRUE)
)

tip_coef_with_continuous(
  effect_observed,
  exposure_confounder_effect = NULL,
  confounder_outcome_effect = NULL,
  verbose = getOption("tipr.verbose", TRUE)
)
```

Arguments

effect_observed	Numeric. Observed exposure - outcome effect from a regression model. This can be the beta coefficient, the lower confidence bound of the beta coefficient, or the upper confidence bound of the beta coefficient.
exposure_confounder_effect	Numeric. Estimated scaled mean difference between the unmeasured confounder in the exposed population and unexposed population
confounder_outcome_effect	Numeric positive value. Estimated relationship between the unmeasured confounder and the outcome
verbose	Logical. Indicates whether to print informative message. Default: TRUE

Value

Data frame.

Examples

```
## to estimate the relationship between an unmeasured confounder and outcome
## needed to tip analysis
tip_coef(1.2, exposure_confounder_effect = -2)

## to estimate the number of unmeasured confounders specified needed to tip
## the analysis
tip_coef(1.2, exposure_confounder_effect = -2, confounder_outcome_effect = -0.05)

## Example with broom
if (requireNamespace("broom", quietly = TRUE) &&
    requireNamespace("dplyr", quietly = TRUE)) {
  lm(wt ~ mpg, data = mtcars) %>%
   broom::tidy(conf.int = TRUE) %>%
   dplyr::filter(term == "mpg") %>%
```

```
   dplyr::pull(conf.low) %>%
   tip_coef(confounder_outcome_effect = 2.5)
}
```

`tip_coef_with_r2` *Tip a regression coefficient using the partial R2 for an unmeasured confounder-exposure relationship and unmeasured confounder- outcome relationship*

Description

Choose one of the following, and the other will be estimated:

- `confounder_exposure_r2`
- `confounder_outcome_r2`

Usage

```
tip_coef_with_r2(
  effect_observed,
  se,
  df,
  confounder_exposure_r2 = NULL,
  confounder_outcome_r2 = NULL,
  verbose = getOption("tipr.verbose", TRUE),
  alpha = 0.05,
  tip_bound = FALSE,
  ...
)
```

Arguments

`effect_observed`
: Numeric. Observed exposure - outcome effect from a regression model. This is the point estimate (beta coefficient)

`se`
: Numeric. Standard error of the `effect_observed` in the previous parameter.

`df`
: Numeric positive value. Residual degrees of freedom for the model used to estimate the observed exposure - outcome effect. This is the total number of observations minus the number of parameters estimated in your model. Often for models estimated with an intercept this is N - k - 1 where k is the number of predictors in the model.

`confounder_exposure_r2`
: Numeric value between 0 and 1. The assumed partial R2 of the unobserved confounder with the exposure given the measured covariates.

`confounder_outcome_r2`
: Numeric value between 0 and 1. The assumed partial R2 of the unobserved confounder with the outcome given the exposure and the measured covariates.

verbose	Logical. Indicates whether to print informative message. Default: TRUE
alpha	Significance level. Default = 0.05.
tip_bound	Do you want to tip at the bound? Default = FALSE, will tip at the point estimate
...	Optional arguments passed to the sensemakr::adjusted_estimate() function.

Value

A data frame.

Examples

```
tip_coef_with_r2(0.5, 0.1, 102, 0.5)
```

---

tip_hr *Tip an observed hazard ratio with a normally distributed confounder.*

Description

choose one of the following, and the other will be estimated:

- exposure_confounder_effect
- confounder_outcome_effect

Usage

```
tip_hr(
  effect_observed,
  exposure_confounder_effect = NULL,
  confounder_outcome_effect = NULL,
  verbose = getOption("tipr.verbose", TRUE),
  hr_correction = FALSE
)

tip_hr_with_continuous(
  effect_observed,
  exposure_confounder_effect = NULL,
  confounder_outcome_effect = NULL,
  verbose = getOption("tipr.verbose", TRUE),
  hr_correction = FALSE
)
```

Arguments

effect_observed	Numeric positive value. Observed exposure - outcome hazard ratio. This can be the point estimate, lower confidence bound, or upper confidence bound.
exposure_confounder_effect	Numeric. Estimated difference in scaled means between the unmeasured confounder in the exposed population and unexposed population
confounder_outcome_effect	Numeric positive value. Estimated relationship between the unmeasured confounder and the outcome
verbose	Logical. Indicates whether to print informative message. Default: TRUE
hr_correction	Logical. Indicates whether to use a correction factor. The methods used for this function are based on risk ratios. For rare outcomes, a hazard ratio approximates a risk ratio. For common outcomes, a correction factor is needed. If you have a common outcome (>15%), set this to TRUE. Default: FALSE.

Value

Data frame.

Examples

```
## to estimate the relationship between an unmeasured confounder and outcome
## needed to tip analysis
tip_hr(1.2, exposure_confounder_effect = -2)

## to estimate the number of unmeasured confounders specified needed to tip
## the analysis
tip_hr(1.2, exposure_confounder_effect = -2, confounder_outcome_effect = .99)
```

---

tip_hr_with_binary *Tip an observed hazard ratio with a binary confounder.*

Description

Choose two of the following three to specify, and the third will be estimated:

- exposed_confounder_prev
- unexposed_confounder_prev
- confounder_outcome_effect

Alternatively, specify all three and the function will return the number of unmeasured confounders specified needed to tip the analysis.

Usage

```
tip_hr_with_binary(
  effect_observed,
  exposed_confounder_prev = NULL,
  unexposed_confounder_prev = NULL,
  confounder_outcome_effect = NULL,
  verbose = getOption("tipr.verbose", TRUE),
  hr_correction = FALSE
)
```

Arguments

effect_observed	Numeric positive value. Observed exposure - outcome hazard ratio. This can be the point estimate, lower confidence bound, or upper confidence bound.
exposed_confounder_prev	Numeric between 0 and 1. Estimated prevalence of the unmeasured confounder in the exposed population
unexposed_confounder_prev	Numeric between 0 and 1. Estimated prevalence of the unmeasured confounder in the unexposed population
confounder_outcome_effect	Numeric positive value. Estimated relationship between the unmeasured confounder and the outcome
verbose	Logical. Indicates whether to print informative message. Default: TRUE
hr_correction	Logical. Indicates whether to use a correction factor. The methods used for this function are based on risk ratios. For rare outcomes, a hazard ratio approximates a risk ratio. For common outcomes, a correction factor is needed. If you have a common outcome (>15%), set this to TRUE. Default: FALSE.

Value

Data frame.

Examples

```
tip_hr_with_binary(0.9, 0.9, 0.1)
```

---

tip_or *Tip an observed odds ratio with a normally distributed confounder.*

Description

choose one of the following, and the other will be estimated:

- exposure_confounder_effect
- confounder_outcome_effect

Usage

```
tip_or(
  effect_observed,
  exposure_confounder_effect = NULL,
  confounder_outcome_effect = NULL,
  verbose = getOption("tipr.verbose", TRUE),
  or_correction = FALSE
)

tip_or_with_continuous(
  effect_observed,
  exposure_confounder_effect = NULL,
  confounder_outcome_effect = NULL,
  verbose = getOption("tipr.verbose", TRUE),
  or_correction = FALSE
)
```

Arguments

`effect_observed`	Numeric positive value. Observed exposure - outcome odds ratio. This can be the point estimate, lower confidence bound, or upper confidence bound.
`exposure_confounder_effect`	Numeric. Estimated difference in scaled means between the unmeasured confounder in the exposed population and unexposed population
`confounder_outcome_effect`	Numeric positive value. Estimated relationship between the unmeasured confounder and the outcome
`verbose`	Logical. Indicates whether to print informative message. Default: `TRUE`
`or_correction`	Logical. Indicates whether to use a correction factor. The methods used for this function are based on risk ratios. For rare outcomes, an odds ratio approximates a risk ratio. For common outcomes, a correction factor is needed. If you have a common outcome (>15%), set this to `TRUE`. Default: `FALSE`.

Value

Data frame.

Examples

```
## to estimate the relationship between an unmeasured confounder and outcome
## needed to tip analysis
tip_or(1.2, exposure_confounder_effect = -2)

## to estimate the number of unmeasured confounders specified needed to tip
## the analysis
tip_or(1.2, exposure_confounder_effect = -2, confounder_outcome_effect = .99)
```

```
## Example with broom
if (requireNamespace("broom", quietly = TRUE) &&
    requireNamespace("dplyr", quietly = TRUE)) {
  glm(am ~ mpg, data = mtcars, family = "binomial") %>%
   broom::tidy(conf.int = TRUE, exponentiate = TRUE) %>%
   dplyr::filter(term == "mpg") %>%
   dplyr::pull(conf.low) %>%
   tip_or(confounder_outcome_effect = 2.5, or_correction = TRUE)
}
```

`tip_or_with_binary` *Tip an observed odds ratio with a binary confounder.*

Description

Choose two of the following three to specify, and the third will be estimated:

- `exposed_confounder_prev`
- `unexposed_confounder_prev`
- `confounder_outcome_effect`

Alternatively, specify all three and the function will return the number of unmeasured confounders specified needed to tip the analysis.

Usage

```
tip_or_with_binary(
  effect_observed,
  exposed_confounder_prev = NULL,
  unexposed_confounder_prev = NULL,
  confounder_outcome_effect = NULL,
  verbose = getOption("tipr.verbose", TRUE),
  or_correction = FALSE
)
```

Arguments

`effect_observed`
: Numeric positive value. Observed exposure - outcome odds ratio. This can be the point estimate, lower confidence bound, or upper confidence bound.

`exposed_confounder_prev`
: Numeric between 0 and 1. Estimated prevalence of the unmeasured confounder in the exposed population

`unexposed_confounder_prev`
: Numeric between 0 and 1. Estimated prevalence of the unmeasured confounder in the unexposed population

confounder_outcome_effect
: Numeric positive value. Estimated relationship between the unmeasured confounder and the outcome

verbose
: Logical. Indicates whether to print informative message. Default: TRUE

or_correction
: Logical. Indicates whether to use a correction factor. The methods used for this function are based on risk ratios. For rare outcomes, an odds ratio approximates a risk ratio. For common outcomes, a correction factor is needed. If you have a common outcome (>15%), set this to TRUE. Default: FALSE.

Value

Data frame.

Examples

```
tip_or_with_binary(0.9, 0.9, 0.1)
```

---

tip_rr *Tip an observed risk ratio with a normally distributed confounder.*

Description

choose one of the following, and the other will be estimated:

- exposure_confounder_effect
- confounder_outcome_effect

Usage

```
tip_rr(
  effect_observed,
  exposure_confounder_effect = NULL,
  confounder_outcome_effect = NULL,
  verbose = getOption("tipr.verbose", TRUE)
)

tip_rr_with_continuous(
  effect_observed,
  exposure_confounder_effect = NULL,
  confounder_outcome_effect = NULL,
  verbose = getOption("tipr.verbose", TRUE)
)
```

Arguments

effect_observed	Numeric positive value. Observed exposure - outcome risk ratio. This can be the point estimate, lower confidence bound, or upper confidence bound.
exposure_confounder_effect	Numeric. Estimated difference in scaled means between the unmeasured confounder in the exposed population and unexposed population
confounder_outcome_effect	Numeric positive value. Estimated relationship between the unmeasured confounder and the outcome
verbose	Logical. Indicates whether to print informative message. Default: TRUE

Value

Data frame.

Examples

```
## to estimate the relationship between an unmeasured confounder and outcome
## needed to tip analysis
tip_rr(1.2, exposure_confounder_effect = -2)

## to estimate the number of unmeasured confounders specified needed to tip
## the analysis
tip_rr(1.2, exposure_confounder_effect = -2, confounder_outcome_effect = .99)
```

---

tip_rr_with_binary *Tip an observed risk ratio with a binary confounder.*

Description

Choose two of the following three to specify, and the third will be estimated:

- exposed_confounder_prev
- unexposed_confounder_prev
- confounder_outcome_effect

Alternatively, specify all three and the function will return the number of unmeasured confounders specified needed to tip the analysis.

Usage

```
tip_rr_with_binary(
  effect_observed,
  exposed_confounder_prev = NULL,
  unexposed_confounder_prev = NULL,
  confounder_outcome_effect = NULL,
  verbose = getOption("tipr.verbose", TRUE)
)
```

Arguments

effect_observed	Numeric positive value. Observed exposure - outcome risk ratio. This can be the point estimate, lower confidence bound, or upper confidence bound.
exposed_confounder_prev	Numeric between 0 and 1. Estimated prevalence of the unmeasured confounder in the exposed population
unexposed_confounder_prev	Numeric between 0 and 1. Estimated prevalence of the unmeasured confounder in the unexposed population
confounder_outcome_effect	Numeric positive value. Estimated relationship between the unmeasured confounder and the outcome
verbose	Logical. Indicates whether to print informative message. Default: TRUE

---

tip_with_binary *Tip a result with a binary confounder.*

Description

Choose two of the following three to specify, and the third will be estimated:

- exposed_confounder_prev
- unexposed_confounder_prev
- confounder_outcome_effect

Alternatively, specify all three and the function will return the number of unmeasured confounders specified needed to tip the analysis.

Usage

```
tip_with_binary(
  effect_observed,
  exposed_confounder_prev = NULL,
  unexposed_confounder_prev = NULL,
  confounder_outcome_effect = NULL,
```

```
  verbose = getOption("tipr.verbose", TRUE),
  correction_factor = "none"
)

tip_b(
  effect_observed,
  exposed_confounder_prev = NULL,
  unexposed_confounder_prev = NULL,
  confounder_outcome_effect = NULL,
  verbose = getOption("tipr.verbose", TRUE),
  correction_factor = "none"
)
```

Arguments

`effect_observed`
: Numeric positive value. Observed exposure - outcome effect (assumed to be the exponentiated coefficient, so a risk ratio, odds ratio, or hazard ratio). This can be the point estimate, lower confidence bound, or upper confidence bound.

`exposed_confounder_prev`
: Numeric between 0 and 1. Estimated prevalence of the unmeasured confounder in the exposed population

`unexposed_confounder_prev`
: Numeric between 0 and 1. Estimated prevalence of the unmeasured confounder in the unexposed population

`confounder_outcome_effect`
: Numeric positive value. Estimated relationship between the unmeasured confounder and the outcome

`verbose`
: Logical. Indicates whether to print informative message. Default: `TRUE`

`correction_factor`
: Character string. Options are "none", "hr", "or". For common outcomes (>15%), the odds ratio or hazard ratio is not a good estimate for the risk ratio. In these cases, we can apply a correction factor. If you are supplying a hazard ratio for a common outcome, set this to "hr"; if you are supplying an odds ratio for a common outcome, set this to "or"; if you are supplying a risk ratio or your outcome is rare, set this to "none" (default).

Details

`tip_b()` is an alias for `tip_with_binary()`.

Examples

```
## to estimate the relationship between an unmeasured confounder and outcome
## needed to tip analysis
tip_with_binary(1.2, exposed_confounder_prev = 0.5, unexposed_confounder_prev = 0)

## to estimate the number of unmeasured confounders specified needed to tip
## the analysis
```

```
tip_with_binary(1.2,
  exposed_confounder_prev = 0.5,
  unexposed_confounder_prev = 0,
  confounder_outcome_effect = 1.1)

## Example with broom
if (requireNamespace("broom", quietly = TRUE) &&
    requireNamespace("dplyr", quietly = TRUE)) {
  glm(am ~ mpg, data = mtcars, family = "binomial") %>%
   broom::tidy(conf.int = TRUE, exponentiate = TRUE) %>%
   dplyr::filter(term == "mpg") %>%
   dplyr::pull(conf.low) %>%
   tip_with_binary(exposed_confounder_prev = 1, confounder_outcome_effect = 1.15)
}
```

Index

∗ **datasets**
exdata_continuous, 12
exdata_rr, 13

adjust_coef, 2
adjust_coef_with_binary, 3
adjust_coef_with_continuous *(*adjust_coef*)*, 2
adjust_coef_with_r2, 5
adjust_hr, 6
adjust_hr_with_binary, 7
adjust_hr_with_continuous *(*adjust_hr*)*, 6
adjust_or, 8
adjust_or_with_binary, 9
adjust_or_with_continuous *(*adjust_or*)*, 8
adjust_rr, 10
adjust_rr_with_binary, 11
adjust_rr_with_continuous *(*adjust_rr*)*, 10

e_value, 13
exdata_continuous, 12
exdata_rr, 13

observed_bias_order, 14
observed_bias_tbl, 14
observed_bias_tip, 15
observed_covariate_e_value, 16

r_value, 17

sensemakr::adjusted_estimate(), *5*, *22*
sensemakr::adjusted_se(), *5*
sensemakr::robustness_value(), *17*

tip, 18
tip_b *(*tip_with_binary*)*, 29
tip_b(), *30*
tip_c *(*tip*)*, 18
tip_coef, 19
tip_coef_with_continuous *(*tip_coef*)*, 19
tip_coef_with_r2, 21
tip_hr, 22
tip_hr_with_binary, 23
tip_hr_with_continuous *(*tip_hr*)*, 22
tip_or, 24
tip_or_with_binary, 26
tip_or_with_continuous *(*tip_or*)*, 24
tip_rr, 27
tip_rr_with_binary, 28
tip_rr_with_continuous *(*tip_rr*)*, 27
tip_with_binary, 29
tip_with_binary(), *30*
tip_with_continuous *(*tip*)*, 18