Package ‘episensr’

January 4, 2017

Type Package
Title Basic Sensitivity Analysis of Epidemiological Results
Version 0.8.0
Imports triangle, trapezoid, plyr, ggplot2, grid, gridExtra, reshape, boot
Date 2017-01-03

BugReports https://github.com/dhaine/episensr/issues

Description Basic sensitivity analysis of the observed relative risks adjusting for unmeasured confounding and misclassification of the exposure/outcome, or both. It follows the bias analysis methods and examples from the book by Lash T.L, Fox M.P, and Fink A.K. "Applying Quantitative Bias Analysis to Epidemiologic Data", ('Springer', 2009).

Depends R (>= 3.3.1)
License GPL-2
Suggests testthat, knitr, rmarkdown, aplore3
RoxygenNote 5.0.1
VignetteBuilder knitr
NeedsCompilation no

Author Denis Haine [aut, cre]
Maintainer Denis Haine <denis.haine@gmail.com>
Repository CRAN
Date/Publication 2017-01-04 18:48:20

R topics documented:

  boot.bias ............................................................... 2
  confounders ........................................................... 3
  confounders.emm ...................................................... 5
  confounders.limit ................................................... 6
bootBias

Description

Generate R bootstrap replicates of either selection or misclassification bias functions. It then generates a confidence interval of the parameter, by first order normal approximation or the bootstrap percentile interval. Replicates giving negative cell(s) in the adjusted 2-by-2 table are silently ignored.

Usage

bootBias(bias_model, R = 1000, conf = 0.95, ci_type = c("norm", "perc"))

Arguments

- bias_model: An object of class "episensr.boot", i.e. either selection bias function or misclassification bias function.
- R: The number of bootstrap replicates.
- conf: Confidence level.
- ci_type: A character string giving the type of interval required. Values can be either "norm" or "perc", default to "norm".

Index

Index
Value

A list with elements:

- **model**: Model ran.
- **boot_mod**: Bootstrap resampled object, of class `boot`.
- **nrep**: Number of replicates used.
- **bias_ciRR**: Bootstrap confidence interval object for relative risk.
- **bias_ciOR**: Bootstrap confidence interval object for odds ratio.
- **ci**: Confidence intervals for the bias adjusted association measures.
- **conf**: Confidence interval.

See Also

`boot`, `selection`, `misclassification`

Examples

```r
misclass_eval <- misclassification(matrix(c(215, 1449, 668, 4296),
dimnames = list(c("Breast cancer+", "Breast cancer-"),
c("Smoker+", "Smoker-")),
nrow = 2, byrow = TRUE),
type = "exposure",
bias_parms = c(.78, .78, .99, .99))

set.seed(123)
boot.bias(misclass_eval)
```

Description

Simple sensitivity analysis to correct for unknown or unmeasured confounding without effect modification. Implementation for ratio measures (relative risk – RR, or odds ratio – OR) and difference measures (risk difference – RD).

Usage

```r
confounders(case, exposed, type = c("RR", "OR", "RD"), bias_parms = NULL,
  alpha = 0.05)
```
Arguments

- **case**: Outcome variable. If a variable, this variable is tabulated against.
- **exposed**: Exposure variable.
- **type**: Choice of implementation, with no effect measure modification for ratio measures (relative risk – RR; odds ratio – OR) or difference measures (risk difference – RD).
- **bias_parms**: Numeric vector defining the 3 necessary bias parameters. This vector has 3 elements, in the following order:
  1. the association between the confounder and the outcome among those who were not exposed,
  2. the prevalence of the confounder among the exposed, and
  3. the prevalence of the confounder among the unexposed.
- **alpha**: Significance level.

Value

A list with elements:

- **obs.data**: The analysed 2 x 2 table from the observed data.
- **cfdдер.data**: The same table for Confounder +.
- **nocfder.data**: The same table for Confounder -.
- **obs.measures**: A table of relative risk with confidence intervals; for Total, Confounder +, and Confounder -.
- **adj.measures**: A table of Standardized Morbidity Ratio and Mantel-Haenszel estimates.
- **biasparms**: Input bias parameters.

References


Examples

```r
# The data for this example come from:
# Tyndall M.W., Ronald A.R., Agoki E., Malisa W., Bwayo J.J., Ndinya-Achola J.O.
# et al.
# Increased risk of infection with human immunodeficiency virus type 1 among
# uncircumcised men presenting with genital ulcer disease in Kenya.
confounders(matrix(c(105, 85, 527, 93),
dimnames = list(c("HIV+", "HIV-"), c("Circ+", "Circ-")),
nrow = 2, byrow = TRUE),
type = "RR",
bias_parms = c(.63, .8, .05))
confounders(matrix(c(105, 85, 527, 93),
dimnames = list(c("HIV+", "HIV-"), c("Circ+", "Circ-")),
nrow = 2, byrow = TRUE),
```
Sensitivity analysis to correct for unknown or unmeasured confounding with effect modification

Description

Simple sensitivity analysis to correct for unknown or unmeasured confounding with effect measure modification. Implementation for ratio measures (relative risk – RR, or odds ratio – OR) and difference measures (risk difference – RD).

Usage

confounders.emm(case, exposed, type = c("RR", "OR", "RD"), bias_parms = NULL, alpha = 0.05)

Arguments

case Outcome variable. If a variable, this variable is tabulated against.
exposed Exposure variable.
type Choice of implementation, with no effect measure modification for ratio measures (relative risk – RR; odds ratio – OR) or difference measures (risk difference – RD).
bias_parms Numeric vector defining the 4 necessary bias parameters. This vector has 4 elements, in the following order:
1. the association between the confounder and the outcome among those who were exposed,
2. the association between the confounder and the outcome among those who were not exposed,
3. the prevalence of the confounder among the exposed, and
4. the prevalence of the confounder among the unexposed.
alpha Significance level.

Value

A list with elements:

obs.data The analysed 2 x 2 table from the observed data.
cfder.data The same table for Confounder +.
nocfder.data  The same table for Confounder -.
obs.measures  A table of relative risk with confidence intervals; Total, for Confounder +, and for Confounder -.
adj.measures  A table of Standardized Morbidity Ratio and Mantel-Haenszel estimates.
bias.parms  Input bias parameters.

References

Examples
# The data for this example come from:
# Tyndall M.W., Ronald A.R., Agoki E., Malisa W., Bwayo J.J., Ndinya-Achola J.O.
# et al.
# Increased risk of infection with human immunodeficiency virus type 1 among
# uncircumcised men presenting with genital ulcer disease in Kenya.
confounders.emm(matrix(c(105, 85, 527, 93),
dimnames = list(c("HIV","HIV-"), c("Circ+","Circ-")),
nrow = 2, byrow = TRUE),
type = "RR",
bias_parms = c(.4, .7, .8, .05))
confounders.emm(matrix(c(105, 85, 527, 93),
dimnames = list(c("HIV","HIV-"), c("Circ+","Circ-")),
nrow = 2, byrow = TRUE),
type = "OR",
bias_parms = c(.4, .7, .8, .05))
confounders.emm(matrix(c(105, 85, 527, 93),
dimnames = list(c("HIV","HIV-"), c("Circ+","Circ-")),
nrow = 2, byrow = TRUE),
type = "RD",
bias_parms = c(-.6, -.3, .8, .05))

confounders.limit  Bounding the bias limits of unmeasured confounding.

Description
Function to elicit the limits on measures of effect corrected for an unmeasured confounder when only some of the bias parameters are known.

Usage
confounders.limit(p = NA, RR = NA, OR = NA, crude.RR = NULL, dec = 4, 
print = TRUE)
confounders.poly

Arguments

- p: Proportion with the confounder among the unexposed group.
- RR: Relative risk between the confounder and the outcome.
- OR: Odds ratio between the confounder and the outcome.
- crude.RR: Crude relative risk between the exposure and the outcome.
- dec: Number of decimals in the printout.
- print: A logical scalar. Should the results be printed?

Value

A list with elements:

- conf.limits: Limits on confounding.
- bias.parms: Input bias parameters p, RR, OR, and crude RR.

References


Examples

confounders.limit(OR = 1.65, crude.RR = 1.5)

dec = 3

print("confounders.poly", dec = dec, conf.limits = c(-0.5, 0.5), print = TRUE)

Description

Simple sensitivity analysis to correct for unknown or unmeasured polychotomous confounding without effect modification. Implementation for ratio measures (relative risk – RR, or odds ratio – OR) and difference measures (risk difference – RD).

Usage

c = confounders.poly(case, exposed, type = c("RR", "OR", "RD"),
                    bias_parms = NULL, alpha = 0.05)
Arguments

case
Outcome variable. If a variable, this variable is tabulated against.
exposed
Exposure variable.
type
Choice of implementation, with no effect measure modification for ratio measures (relative risk – RR; odds ratio – OR) or difference measures (risk difference – RD).
bias_parms
Numeric vector defining the bias parameters. This vector has 6 elements, in the following order:
1. the association between the highest level confounder and the outcome,
2. the association between the mid-level confounder and the outcome,
3. the prevalence of the highest level confounder among the exposed,
4. the prevalence of the highest level confounder among the unexposed,
5. the prevalence of the mid-level confounder among the exposed, and
6. the prevalence of the mid-level confounder among the unexposed.

alpha
Significance level.

Value

A list with elements:

obs.data
The analysed 2 x 2 table from the observed data.
cfder1.data
The same table for Mid-level Confounder +.
cfder2.data
The same table for Highest-level Confounder +.
ncfder.data
The same table for Confounder -.
obs.measures
A table of relative risk with confidence intervals; Total and by confounders.
adj.measures
A table of Standardized Morbidity Ratio and Mantel-Haenszel estimates.
bias.parms
Input bias parameters.

References


Examples

# The data for this example come from:
# Tyndall M.W., Ronald A.R., Agoki E., Malisa W., Bwayo J.J., Ndinya-Achola J.O.
# et al.
# Increased risk of infection with human immunodeficiency virus type 1 among
# uncircumcised men presenting with genital ulcer disease in Kenya.
confounders.poly(matrix(c(105, 85, 527, 93),
    dimnames = list(c("HIV+", "HIV-"), c("Circ+", "Circ-")),
    nrow = 2, byrow = TRUE),
    type = "RR",
    bias_parms = c(.4, .8, .6, .05, .2, .2))
confounders.poly(matrix(c(105, 85, 527, 93),
dimnames = list(c("HIV+", "HIV-"), c("Circ+", "Circ-")),
nrow = 2, byrow = TRUE),
type = "OR",
bias_parms = c(.4, .8, .6, .05, .2, .2))
confounders.poly(matrix(c(105, 85, 527, 93),
dimnames = list(c("HIV+", "HIV-"), c("Circ+", "Circ-")),
nrow = 2, byrow = TRUE),
type = "RD",
bias_parms = c(-.4, -.2, .6, .05, .2, .2))

episensr

Basic sensitivity analysis of epidemiological results

Description
episensr provides basic sensitivity analysis of the observed relative risks adjusting for unmeasured confounding and misclassification of the exposure/outcome, or both.

Details

Package: episensr
Type: Package
Version: 0.6
Date: 2015-03-24
License: GPL-2

Author(s)
Denis Haine <denis.haine@gmail.com>

References

mbias

Sensitivity analysis to correct for selection bias caused by M bias.

Description
Simple sensitivity analysis to correct for selection bias caused by M bias using estimates of the odds ratios relating the variables.
Usage

mbias(or, var)

Arguments

or Vector defining the input bias parameters, in the following order:
1. Odds ratio between A and the exposure E,
2. Odds ratio between A and the collider C,
3. Odds ratio between B and the collider C,
4. Odds ratio between B and the outcome D,
5. Odds ratio observed between the exposure E and the outcome D.

var Vector defining variable names, in the following order:
1. Outcome,
2. Exposure,
3. A,
4. B,
5. Collider.

Value

A list with elements:

mbias.parms Maximum bias parameters.
adj.measures Selection bias corrected measures.
bias.parms Input bias parameters.

References


Examples

mbias(or = c(2, 5.4, 2.5, 1.5, 1),
var = c("HIV", "Circumcision", "Muslim", "Low CD4", "Participation"))
Description

Simple sensitivity analysis for misclassification.

Usage

misclassification(case, exposed, type = c("exposure", "outcome", "confounder"), bias = NULL, bias_parms = NULL, alpha = 0.05)

Arguments

case Outcome variable. If a variable, this variable is tabulated against.
exposed Exposure variable.
type Choice of misclassification:

1. exposure: bias analysis for exposure misclassification; corrections using sensitivity and specificity: nondifferential and independent errors,
2. outcome: bias analysis for outcome misclassification.
3. confounder: bias analysis for confounder misclassification.
bias Deprecated, please use bias_parms instead.
bias_parms Vector defining the bias parameters. This vector has 4 elements between 0 and 1, in the following order:
1. Sensitivity of exposure (or outcome, or confounder) classification among those with the outcome,
2. Sensitivity of exposure (or outcome, or confounder) classification among those without the outcome,
3. Specificity of exposure (or outcome, or confounder) classification among those with the outcome, and
4. Specificity of exposure (or outcome, or confounder) classification among those without the outcome.
alpha Significance level.

Value

A list with elements:

obs.data The analysed 2 x 2 table from the observed data.
corr.data The expected observed data given the true data assuming misclassification.
obs.measures A table of observed relative risk and odds ratio with confidence intervals.
adj.measures A table of adjusted relative risk and odds ratio.
bias.parms Input bias parameters.
References


Examples

# The data for this example come from:
# Fink, A.K., Lash, T.L. A null association between smoking during pregnancy
# and breast cancer using Massachusetts registry data (United States).
# Cancer Causes Control 2003;14:497–503.
misclassification(matrix(c(215, 1449, 668, 4296),
dimnames = list(c("Breast cancer+", "Breast cancer-"),
c("Smoker+", "Smoker-")),
nrow = 2, byrow = TRUE),
type = "exposure",
bias_parms = c(.78, .78, .99, .99))
misclassification(matrix(c(4558, 3428, 46305, 46085),
dimnames = list(c("AMI death+", "AMI death-"),
c("Male+", "Male-")),
nrow = 2, byrow = TRUE),
type = "outcome",
bias_parms = c(.53, .53, .99, .99))

multidimBias

Multidimensional sensitivity analysis for different sources of bias

Description

Multidimensional sensitivity analysis for different sources of bias

Usage

```r
multidimBias(case, exposed, type = c("exposure", "outcome", "confounder", "selection"),
              se = NULL, sp = NULL, bias = NULL, bias_parms = NULL,
              OR.sel = NULL, alpha = 0.05, dec = 4, print = TRUE)
```

Arguments

- **case**: Outcome variable. If a variable, this variable is tabulated against.
- **exposed**: Exposure variable.
- **type**: Implement analysis for exposure misclassification, outcome misclassification, unmeasured confounder, or selection bias.
- **se**: Numeric vector of sensitivities.
- **sp**: Numerical vector of specificities.
- **bias**: Deprecated, please use bias_parms instead.
- **bias_parms**: List of bias parameters. The list is made of 3 vectors of the same length:
multidimBias

1. Prevalence of Confounder in Exposure+ population,
2. Prevalence of Confounder in Exposure- population, and
3. Relative risk between Confounder and Outcome.

OR.sel Selection odds ratios, for selection bias implementation.
alpha Significance level.
dec Number of decimals in the printout.
print A logical scalar. Should the results be printed?

Value

A list with elements:

obs.data The analysed 2 x 2 table from the observed data.
obs.measures A table of odds ratios and relative risk with confidence intervals.
adj.measures Multidimensional corrected relative risk and/or odds ratio data.
bias.parms Bias parameters.

References


Examples

multidimBias(matrix(c(45, 94, 257, 945),
dimnames = list(c("HIV+", "HIV-"), c("Circ+", "Circ-")),
nrow = 2, byrow = TRUE),
type = "exposure",
se = c(1, 1, .9, .9, .9, .8, .8, .8),
sp = c(1, .9, .8, 1, .9, .8, 1, .9, .8))
multidimBias(matrix(c(45, 94, 257, 945),
dimnames = list(c("HIV+", "HIV-"), c("Circ+", "Circ-")),
nrow = 2, byrow = TRUE),
type = "outcome",
se = c(1, 1, 1, .9, .9, .8, .8, .8),
sp = c(1, .9, .8, 1, .9, .8, 1, .9, .8))
multidimBias(matrix(c(105, 85, 527, 93),
dimnames = list(c("HIV+", "HIV-"), c("Circ+", "Circ-")),
nrow = 2, byrow = TRUE),
type = "confounder",
bias_parms = list(seq(.72, .92, by = .02),
seq(.81, .11, by = .01), seq(.13, 1.13, by = .1)))
multidimBias(matrix(c(136, 107, 297, 165),
dimnames = list(c("Uveal Melanoma+", "Uveal Melanoma-"),
c("Mobile Use+", "Mobile Use -")),
nrow = 2, byrow = TRUE),
type = "selection",
OR.sel = seq(1.5, 6.5, by = .5))
plot.episensr.booted

Plot of bootstrap simulation output for selection and misclassification bias

Description

This takes an episensr bootstrap object and produces the plot of bootstrap replicates for selection or misclassification bias of the variable of interest, either relative risk or odds ratio.

Usage

## S3 method for class 'episensr.booted'
plot(x, association = c("rr", "or"), ...)

Arguments

x An object of class "episensr.booted" returned from the episensr bootstrap generation function.

association Choice between bias adjusted relative risk and odds ratio.

... Other unused arguments.

See Also

boot.bias, boot, selection, misclassification

Examples

misclass_eval <- misclassification(matrix(c(215, 1449, 668, 4296),
dimnames = list(c("Breast cancer+", "Breast cancer-"),
c("Smoker+", "Smoker-")),
nrow = 2, byrow = TRUE),
type = "exposure",
bias_parms = c(.78, .78, .99, .99))

set.seed(123)
misclass_boot <- boot.bias(misclass_eval)
plot(misclass_boot, association = "rr")
plot.mbias

**Description**

Create two DAGs, before and after conditioning on the collider C, for selection bias caused by M bias, using ggplot2.

**Usage**

```
## S3 method for class 'mbias'
plot(x, title1 = "DAG before conditioning on C",
     title2 = "DAG after conditioning on C", title.size = 6, size = 6,
     dec = 2, layout = c("landscape", "portrait"), ...)
```

**Arguments**

- `x` 'mbias' object to plot.
- `title1` Title of DAG graph before conditioning on C.
- `title2` Title of DAG graph after conditioning on C.
- `title.size` Title size.
- `size` Text size.
- `dec` Number of digits displayed.
- `layout` Side-by-side graphs in landscape or portrait layout.
- `...` Other unused arguments.

**Value**

Two DAGs for selection bias caused by M bias.

**See Also**

`mbias`

**Examples**

```r
plot(mbias(or = c(2, 5.4, 2.5, 1.5, 1),
       var = c("HIV", "Circumcision", "Muslim", "Low CD4", "Participation")))
```
**print.episensr**  
*Print associations for episensr class*

**Description**

Print associations for episensr objects.

**Usage**

```r
## S3 method for class 'episensr'
print(x, digits = getOption("digits"), ...)
```

**Arguments**

- `x` An object of class 'episensr'.
- `digits` Minimal number of _significant_ digits, see 'print.default'.
- `...` Other unused arguments.

**Value**

Print the observed and adjusted measures of association.

---

**print.episensr.booted**  
*Print bootstraped confidence intervals*

**Description**

Print bootstraped confidence intervals for selection and misclassification bias functions.

**Usage**

```r
## S3 method for class 'episensr.booted'
print(x, digits = getOption("digits"), ...)
```

**Arguments**

- `x` An object of class 'episensr.booted'.
- `digits` Minimal number of _significant_ digits, see 'print.default'.
- `...` Other unused arguments.

**Value**

Print the confidence interval of the adjusted measures of association.
**print.mbias**  
*Print association corrected for M bias*

**Description**

Print association corrected for M bias.

**Usage**

```r
## S3 method for class 'mbias'
print(x, ...)
```

**Arguments**

- `x`: An object of class `mbias`.
- `...`: Other unused arguments.

**Value**

Print the observed and adjusted measures of association.

---

**probsens**  
*Probabilistic sensitivity analysis.*

**Description**

Probabilistic sensitivity analysis to correct for exposure misclassification or outcome misclassification and random error.

**Usage**

```r
probsens(case, exposed, type = c("exposure", "outcome"), reps = 1000, 
seca.parms = list(dist = c("constant", "uniform", "triangular",  
"trapezoidal", "logit-logistic", "logit-normal"), parms = NULL), 
seexp.parms = NULL, spca.parms = list(dist = c("constant", "uniform",  
"triangular", "trapezoidal", "logit-logistic", "logit-normal"), parms = NULL), 
spexp.parms = NULL, corr.se = NULL, corr.sp = NULL, discard = TRUE,  
alpha = 0.05)
```
Arguments

- **case**: Outcome variable. If a variable, this variable is tabulated against.
- **exposed**: Exposure variable.
- **type**: Choice of correction for exposure or outcome misclassification.
- **reps**: Number of replications to run.
- **seca.parms**: List defining the sensitivity of exposure classification among those with the outcome. The first argument provides the probability distribution function (constant, uniform, triangular, trapezoidal, logit-logistic, or logit-normal) and the second its parameters as a vector:
  1. Constant: constant value,
  2. Uniform: min, max,
  3. Triangular: lower limit, upper limit, mode,
  4. Trapezoidal: min, lower mode, upper mode, max,
  5. Logit-logistic: location, scale, lower bound shift, upper bound shift,
- **seexp.parms**: List defining the sensitivity of exposure classification among those without the outcome.
- **spca.parms**: List defining the specificity of exposure classification among those with the outcome.
- **spexp.parms**: List defining the specificity of exposure classification among those without the outcome.
- **corr.se**: Correlation between case and non-case sensitivities.
- **corr.sp**: Correlation between case and non-case specificities.
- **discard**: A logical scalar. In case of negative adjusted count, should the draws be discarded? If set to FALSE, negative counts are set to zero.
- **alpha**: Significance level.

Value

A list with elements:

- **obs.data**: The analysed $2 \times 2$ table from the observed data.
- **obs.measures**: A table of observed relative risk and odds ratio with confidence intervals.
- **adj.measures**: A table of corrected relative risks and odds ratios.
- **sim.df**: Data frame of random parameters and computed values.

References

Examples

# The data for this example come from:
# Greenland S., Salvan A., Wegman D.H., Hallock M.F., Smith T.J.
# A case-control study of cancer mortality at a transformer-assembly facility.
set.seed(123)
# Exposure misclassification, non-differential
probsens(matrix(c(45, 94, 257, 945),
dimnames = list(c("BC+", "BC-"), c("Smoke+", "Smoke-")), nrow = 2, byrow = TRUE),
type = "exposure",
reps = 20000,
seca.parms = list("trapezoidal", c(.75, .85, .95, 1)),
spca.parms = list("trapezoidal", c(.75, .85, .95, 1)))
# Exposure misclassification, differential
probsens(matrix(c(45, 94, 257, 945),
dimnames = list(c("BC+", "BC-"), c("Smoke+", "Smoke-")), nrow = 2, byrow = TRUE),
type = "exposure",
reps = 20000,
seca.parms = list("trapezoidal", c(.75, .85, .95, 1)),
seexp.parms = list("trapezoidal", c(.7, .8, .9, .95)),
spca.parms = list("trapezoidal", c(.75, .85, .95, 1)),
spec.parms = list("trapezoidal", c(.7, .8, .9, .95)),
corr.se = .8,
corr.sp = .8)
# Disease misclassification
probsens(matrix(c(173, 662, 134, 663),
dimnames = list(c("BC+", "BC-"), c("Smoke+", "Smoke-")), nrow = 2, byrow = TRUE),
type = "outcome",
reps = 20000,
seca.parms = list("uniform", c(.8, 1)),
spca.parms = list("uniform", c(.8, 1))

---

Description

Probabilistic sensitivity analysis to correct for unknown or unmeasured confounding and random error simultaneously.

Usage

probsens.conf(case, exposed, reps = 1000, prev.exp = list(dist =
c("constant", "uniform", "triangular", "trapezoidal", "logit-logistic",
"logit-normal"), params = NULL), prev.nexp = list(dist = c("constant",
"uniform", "triangular", "trapezoidal", "logit-logistic", "logit-normal"),
params = NULL), risk = list(dist = c("constant", "uniform", "triangular",
"trapezoidal", "log-logistic", "log-normal"), params = NULL), corr.p = NULL,
discard = TRUE, alpha = .05)
Arguments

- **case**: Outcome variable. If a variable, this variable is tabulated against.
- **exposed**: Exposure variable.
- **reps**: Number of replications to run.
- **prev.exp**: List defining the prevalence of exposure among the exposed. The first argument provides the probability distribution function (constant, uniform, triangular, trapezoidal, logit-logistic, or logit-normal) and the second its parameters as a vector:
  1. Constant: constant value,
  2. Uniform: min, max,
  3. Triangular: lower limit, upper limit, mode,
  4. Trapezoidal: min, lower mode, upper mode, max.
  5. Logit-logistic: location, scale, lower bound shift, upper bound shift,
- **prev.nexp**: List defining the prevalence of exposure among the unexposed.
- **risk**: List defining the confounder-disease relative risk or the confounder-exposure odds ratio. The first argument provides the probability distribution function (constant, uniform, triangular, trapezoidal, log-logistic, or log-normal) and the second its parameters as a vector:
  1. Constant: constant value,
  2. Uniform: min, max,
  3. Triangular: lower limit, upper limit, mode,
  4. Trapezoidal: min, lower mode, upper mode, max.
  5. Log-logistic: location, scale,
- **corr.p**: Correlation between the exposure-specific confounder prevalences.
- **discard**: A logical scalar. In case of negative adjusted count, should the draws be discarded? If set to FALSE, negative counts are set to zero.
- **alpha**: Significance level.

Value

A list with elements:

- **obs.data**: The analysed 2 x 2 table from the observed data.
- **obs.measures**: A table of observed relative risk and odds ratio with confidence intervals.
- **adj.measures**: A table of corrected relative risks and odds ratios.
- **sim.df**: Data frame of random parameters and computed values.

References

Examples

# The data for this example come from:
# Increased risk of infection with human immunodeficiency virus type 1 among
# uncircumcised men presenting with genital ulcer disease in Kenya.
# set.seed(123)
probsens.conf(matrix(c(105, 85, 527, 93),
dimnames = list(c("HIV+", "HIV-"), c("Circ+", "Circ-")), nrow = 2, byrow = TRUE),
reps = 20000,
prev.exp = list("triangular", c(.7, .9, .8)),
prev.nexp = list("trapezoidal", c(.03, .04, .05, .06)),
risk = list("triangular", c(.6, .7, .63)),
corr.p = .8)

probsens. irr  
Probabilistic sensitivity analysis for exposure misclassification of
person-time data and random error.

Description

Probabilistic sensitivity analysis to correct for exposure misclassification when person-time data
has been collected.

Usage

probsens. irr(counts, pt = NULL, reps = 1000, seca.parms = list(dist =
c("constant", "uniform", "triangular", "trapezoidal", "logit-logistic",
"logit-normal"), parms = NULL), seexp.parms = NULL, spca.parms = list(dist =
c("constant", "uniform", "triangular", "trapezoidal", "logit-logistic",
"logit-normal"), parms = NULL), spexp.parms = NULL, corr.se = NULL, corr.sp = NULL, discard = TRUE, alpha = 0.05)

Arguments

counts  
A table or matrix where first row contains disease counts and second row con-
tains person-time at risk, and first and second columns are exposed and unex-
posed observations, as:

<table>
<thead>
<tr>
<th></th>
<th>Exposed</th>
<th>Unexposed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cases</td>
<td>a</td>
<td>b</td>
</tr>
<tr>
<td>Person-time</td>
<td>N1</td>
<td>N0</td>
</tr>
</tbody>
</table>

pt  
A numeric vector of person-time at risk. If provided, counts must be a numeric vector of disease counts.

reps  
Number of replications to run.

seca.parms  
List defining the sensitivity of exposure classification among those with the out-
The first argument provides the probability distribution function (uniform, triangular, trapezoidal, logit-logistic, or logit-normal) and the second its parameters as a vector:

1. Constant: constant value,
2. Uniform: min, max,
3. Triangular: lower limit, upper limit, mode,
4. Trapezoidal: min, lower mode, upper mode, max,
5. Logit-logistic: location, scale, lower bound shift, upper bound shift,

- `seexp.parms` List defining the sensitivity of exposure classification among those without the outcome.
- `spca.parms` List defining the specificity of exposure classification among those with the outcome.
- `spexp.parms` List defining the specificity of exposure classification among those without the outcome.
- `corr.se` Correlation between case and non-case sensitivities.
- `corr.sp` Correlation between case and non-case specificities.
- `discard` A logical scalar. In case of negative adjusted count, should the draws be discarded? If set to FALSE, negative counts are set to zero.
- `alpha` Significance level.

**Value**

A list with elements:

- `obs.data` The analysed 2 x 2 table from the observed data.
- `obs.measures` A table of observed incidence rate ratio with exact confidence interval.
- `adj.measures` A table of corrected incidence rate ratios.
- `sim.df` Data frame of random parameters and computed values.

**References**


**Examples**

```
set.seed(123)
# Exposure misclassification, non-differential
probsens.rr(matrix(c(2, 67232, 58, 105390000),
dimnames = list(c("GBS+", "Person-time"), c("HPV+", "HPV-")), ncol = 2),
reps = 20000,
seca.parms = list("trapezoidal", c(.4, .45, .55, .6)),
spca.parms = list("constant", 1))
```
Probabilistic sensitivity analysis for unmeasured confounding of person-time data and random error.

Description

Probabilistic sensitivity analysis to correct for unmeasured confounding when person-time data has been collected.

Usage

```r
probsens irr.conf(counts, pt = NULL, reps = 1000, prev.exp = list(dist =
c("constant", "uniform", "triangular", "trapezoidal", "logit-logistic",
"logit-normal"), parms = NULL), prev.nexp = list(dist = c("constant",
"uniform", "triangular", "trapezoidal", "logit-logistic", "logit-normal"),
parms = NULL), risk = list(dist = c("constant", "uniform", "triangular",
"trapezoidal", "log-logistic", "log-normal"), parms = NULL), corr.p = NULL,
alpha = 0.05)
```

Arguments

- **counts**: A table or matrix where first row contains disease counts and second row contains person-time at risk, and first and second columns are exposed and unexposed observations, as:

<table>
<thead>
<tr>
<th></th>
<th>Exposed</th>
<th>Unexposed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cases</td>
<td>a</td>
<td>b</td>
</tr>
<tr>
<td>Person-time</td>
<td>N1</td>
<td>N0</td>
</tr>
</tbody>
</table>

- **pt**: A numeric vector of person-time at risk. If provided, counts must be a numeric vector of disease counts.

- **reps**: Number of replications to run.

- **prev.exp**: List defining the prevalence of exposure among the exposed. The first argument provides the probability distribution function (constant, uniform, triangular, trapezoidal, logit-logistic, or logit-normal) and the second its parameters as a vector:

  1. Constant; value,
  2. Uniform: min, max,
  3. Triangular: lower limit, upper limit, mode,
  4. Trapezoidal: min, lower mode, upper mode, max.
  5. Logit-logistic: location, scale, lower bound shift, upper bound shift,

- **prev.nexp**: List defining the prevalence of exposure among the unexposed.

- **risk**: List defining the confounder-disease relative risk or the confounder-exposure odds ratio. The first argument provides the probability distribution function...
(constant, uniform, triangular, trapezoidal, log-logistic, or log-normal) and the second its parameters as a vector:

1. Constant: value,
2. Uniform: min, max,
3. Triangular: lower limit, upper limit, mode,
4. Trapezoidal: min, lower mode, upper mode, max.
5. Log-logistic: location, scale,

corr.p  Correlation between the exposure-specific confounder prevalences.
alpha  Significance level.

Value

A list with elements:

- obs.data  The analysed 2 x 2 table from the observed data.
- obs.measures  A table of observed incidence rate ratio with exact confidence interval.
- adj.measures  A table of corrected incidence rate ratios.
- sim.df  Data frame of random parameters and computed values.

References


Examples

```r
set.seed(123)
# Unmeasured confounding
probsens irr.conf(matrix(c(77, 10000, 87, 10000),
dimnames = list(c("D+", "Person-time"), c("E+", "E-")), ncol = 2),
reps = 20000,
prev.exp = list("trapezoidal", c(.01, .2, .3, .51)),
prev.nexp = list("trapezoidal", c(.09, .27, .35, .59)),
risk = list("trapezoidal", c(2, 2.5, 3.5, 4.5)),
corr.p = .8)
```

---

**probsens.sel**  Probabilistic sensitivity analysis for selection bias.

**Description**

Probabilistic sensitivity analysis to correct for selection bias.
Usage

probsens.sel(case, exposed, reps = 1000, or.parms = list(dist =
c("constant", "uniform", "triangular", "trapezoidal", "logit-logistic",
"logit-normal"), parms = NULL), alpha = 0.05)

Arguments

case Outcome variable. If a variable, this variable is tabulated against.
exposed Exposure variable.
reps Number of replications to run.
or.parms List defining the selection bias odds. The first argument provides the probability distribution function (constant, uniform, triangular, or trapezoidal) and the second its parameters as a vector:
1. Constant: constant value,
2. Uniform: min, max,
3. Triangular: lower limit, upper limit, mode,
4. Trapezoidal: min, lower mode, upper mode, max.
5. Logit-logistic: location, scale, lower bound shift, upper bound shift,

alpha Significance level.

Value

A list with elements:

obs.data The analysed 2 x 2 table from the observed data.
obs.measures A table of observed odds ratio with confidence intervals.
adj.measures A table of corrected odds ratios.
sim.df Data frame of random parameters and computed values.

References


Examples

# The data for this example come from:
# Population-based incidence estimates of uveal melanoma in Germany.
# Supplementing cancer registry data by case-control data.
set.seed(123)
probsens.sel(matrix(c(136, 107, 297, 165),
dimnames = list(c("Melanoma+", "Melanoma-"), c("Mobile+", "Mobile-")), nrow = 2, byrow = TRUE),
reps = 20000,
or.parms = list("triangular", c(.35, 1.1, .43)))
**Description**

Simple sensitivity analysis to correct for selection bias using estimates of the selection proportions.

**Usage**

```r
selection(case, exposed, selprob = NULL, bias_parms = NULL, alpha = 0.05)
```

**Arguments**

- `case`: Outcome variable. If a variable, this variable is tabulated against.
- `exposed`: Exposure variable.
- `selprob`: Deprecated, please use `bias_parms` instead.
- `bias_parms`: Numeric vector defining the selection probabilities. This vector has 4 elements between 0 and 1, in the following order:
  1. Selection probability among cases exposed,
  2. Selection probability among cases unexposed,
  3. Selection probability among noncases exposed, and
  4. Selection probability among noncases unexposed.
- `alpha`: Significance level.

**Value**

A list with elements:

- `obs.data`: The analysed 2 x 2 table from the observed data.
- `corr.data`: The same table corrected for selection proportions.
- `obs.measures`: A table of odds ratios and relative risk with confidence intervals.
- `adj.measures`: Selection bias corrected measures of outcome-exposure relationship.
- `selbias.or`: Selection bias odds ratio based on the bias parameters chosen.

**Examples**

```r
# The data for this example come from:
# Population-based incidence estimates of uveal melanoma in Germany. Supplementing
# cancer registry data by case-control data.
selection(matrix(c(136, 107, 297, 165),
dimnames = list(c("UM+", "UM-"), c("Mobile+", "Mobile-")),
nrow = 2, byrow = TRUE),
bias_parms = c(.94, .85, .64, .25))
```
Index

boot, 3, 14
boot.bias, 2, 14

confounders, 3
confounders.emm, 5
confounders.limit, 6
confounders.poly, 7

episensr, 9
episensr-package (episensr), 9

mbias, 9, 15
misclassification, 3, 11, 14
multidimBias, 12

plot.episensr.booted, 14
plot.mbias, 15
print.episensr, 16
print.episensr.booted, 16
print.mbias, 17
probsens, 17
probsens.conf, 19
probsens.irr, 21
probsens.irr.conf, 23
probsens.sel, 24

selection, 3, 14, 26