Package ‘bpbounds’

October 12, 2022

Title Nonparametric Bounds for the Average Causal Effect Due to Balke and Pearl and Extensions

Version 0.1.4

Description Implementation of the nonparametric bounds for the average causal effect under an instrumental variable model by Balke and Pearl (Bounds on Treatment Effects from Studies with Imperfect Compliance, JASA, 1997, 92, 439, 1171-1176). The package can calculate bounds for a binary outcome, a binary treatment/phenotype, and an instrument with either 2 or 3 categories. The package implements bounds for situations where these 3 variables are measured in the same dataset (trivariate data) or where the outcome and instrument are measured in one study and the treatment/phenotype and instrument are measured in another study (bivariate data).

Depends R (>= 3.5.0)

Imports methods

License GPL-3

Encoding UTF-8

LazyData true

Suggests dplyr, testthat, knitr, rmarkdown, tidyr, shiny

VignetteBuilder knitr

RoxygenNote 7.0.2

URL `https://github.com/remlapmot/bpbounds`

BugReports `https://github.com/remlapmot/bpbounds/issues`

NeedsCompilation no

Author Tom Palmer [aut, cre] (<`https://orcid.org/0000-0003-4655-4511`>),
Roland Ramsahai [aut] (<`https://orcid.org/0000-0002-7349-1977`>),
Vanessa Didelez [aut] (<`https://orcid.org/0000-0001-8587-7706`>),
Nuala Sheehan [aut] (<`https://orcid.org/0000-0001-9094-2605`>)

Maintainer Tom Palmer <`remlapmot@hotmail.com`>

Repository CRAN

Date/Publication 2020-01-21 11:10:03 UTC

R topics documented:

`bpbounds-package` *bpbounds: Nonparametric bounds for the Average Causal Effect due to Balke and Pearl and extensions*

Description

This package implements the nonparametric bounds for the average causal effect defined by Balke and Pearl, Bounds on Treatment Effects from Studies with Imperfect Compliance, JASA, 1997; and some extensions.

Details

The functions implement bounds for the situation where each of the outcome, treatment/phenotype, and instrumental variable are binary; and additionally for when the instrument has 3 categories (e.g. a single genotype under an additive model in a Mendelian randomization study).

The package implements bounds for when the three variables are measured in the same study (trivariate data) and when the outcome and instrument are measured in one study and the treatment/phenotype and instrument in another sample (bivariate/two sample data).

`bpbounds` *Nonparametric Bounds for the Average Causal Effect due to Balke and Pearl.*

Description

Nonparametric Bounds for the Average Causal Effect due to Balke and Pearl.

Usage

```
bpbounds(p, t = NULL, fmt = "trivariate")
```

Arguments

`p`

Object of class "table" containing either cell counts or conditional probabilities. For trivariate data these are for the phenotype/treatment-outcome association given Z, i.e. P(X, Y | Z).

Cell counts could be generated from `xtabs(~ x + y + z, data = data)`. And then conditional probabilities obatained by calling `prop.table(..., margins = 3)` on your object from `xtabs()`.

If you only know the conditional probabilities you can enter these, e.g. for the Balke and Pearl Vitamin A example:

```
cp   = c(.0064, 0, .9936, 0, .0028, .001, .1972, .799)
tabp = as.table(array(
  cp,
  dim = c(2, 2, 2),
  dimnames = list(
    x = c(0, 1),
    y = c(0, 1),
    z = c(0, 1)
  )
))
```

And then call `bpbounds()` using this object.

For bivariate data this object contains cell conditional probabilities for the outcome-instrument (Y|Z) association.

`t`

Specified for bivariate data. Object with treatment/phenotype-instrument cell counts or conditional probabilities, i.e. (X|Z).

`fmt`

A character string which sould be either "bivariate" (i.e. X, Z in one dataset and Y, Z in another dataset) or "trivariate" (X, Y, Z in the same dataset).

Value

List with the following elements:

fmt whether the data is bivariate or trivariate

nzcats 2 or 3, the no. instrument categories

inequality Logical, indicating whether the IV inquality is satisfied

bplb Lower bound of ACE

bpub Upper bound of ACE

bplower Vector of lower bound probabilities

bpupper Vector of upper bound probabilities

p11low Lower bound of P(Y=1|do(X=1))

p11upp Upper bound of P(Y=1|do(X=1))

p10low Lower bound of P(Y=1|do(X=0))

p10upp Upper bound of P(Y=1|do(X=0))

p11lower Vector of probabilities for lower bound of P(Y=1|do(X=1))

p11upper Vector of probabilities for upper bound of P(Y=1|do(X=1))

p10lower Vector of probabilities for lower bound of P(Y=1|do(X=0))

p10upper Vector of probabilities for upper bound of P(Y=1|do(X=0))

crrlb Lower bound of CRR

crrub Upper bound of CRR

monoinequality Logical, indicating whether the monoticity inequality is satisfied

monobplb Lower bound of ACE assuming monotonicity

monobpub Upper bound of ACE assuming monotonicity

monobplower Vector of probabilities for lower bound of ACE assuming monotonicity

monobpupper Vector of probabilities for upper bound of ACE assuming monotonicity

monop11low Lower bound of P(Y=1|do(X=1)) assuming monotonicity

monop11upp Upper bound of P(Y=1|do(X=1)) assuming monotonicity

monop10low Lower bound of P(Y=1|do(X=0)) assuming monotonicity

monop10upp Upper bound of P(Y=1|do(X=0)) assuming monotonicity

monop11lower Vector for corresponding bound above

monop11upper Vector for corresponding bound above

monop10lower Vector for corresponding bound above

monop10upper Vector for corresponding bound above

monocrrlb Lower bound of CRR assuming monotonicity

monocrrub Upper bound of CRR assuming monotonicity

Examples

```
# Vitamin A example, using cell counts

require(tidyr)
require(bpbounds)

tab1dat <- data.frame(
  z = c(0, 0, 1, 1, 1, 1, 0, 0),
  x = c(0, 0, 0, 0, 1, 1, 1, 1),
  y = c(0, 1, 0, 1, 0, 1, 0, 1),
  freq = c(74, 11514, 34, 2385, 12, 9663, 0, 0)
)

tab1inddat = uncount(tab1dat, freq)
xt = xtabs(~ x + y + z, data = tab1inddat)
p = prop.table(xt, margin = 3)
bpres = bpbounds(p)
sbpres = summary(bpres)
print(sbpres)
```

```
# Vitamin A example, using conditional probabilities

require(bpbounds)
cp = c(.0064, 0, .9936, 0, .0028, .001, .1972, .799)
tabp = as.table(array(
  cp,
  dim = c(2, 2, 2),
  dimnames = list(
    x = c(0, 1),
    y = c(0, 1),
    z = c(0, 1)
  )
))
bpbounds(tabp)
```

`runExample` *Run Shiny App demonstrating the package*

Description

Run Shiny App demonstrating the package

Usage

```
runExample(...)
```

Arguments

`...`	passed to `shiny::runApp()`, e.g. port, launch.browser

Examples

```
if (interactive()) {
  bpbounds::runExample()
}
```

Index