Package ‘rbounds’

July 23, 2025

Version 2.2

Title Perform Rosenbaum Bounds Sensitivity Tests for Matched and Unmatched Data

Date 2022-04-15

Author Luke J. Keele

Maintainer Luke J. Keele <`luke.keele@gmail.com`>

Description Takes matched and unmatched data and calculates Rosenbaum bounds for the treatment effect. Calculates bounds for binary outcome data, Hodges-Lehmann point estimates, Wilcoxon signed-rank test for matched data and matched IV estimators, Wilcoxon sum rank test, and for data with multiple matched controls. The sensitivity analysis methods in this package are documented in Rosenbaum (2002) Observational Studies, <doi:10.1007/978-1-4757-3692-2>, Springer-Verlag.

License GPL (>= 2)

NeedsCompilation no

Repository CRAN

Date/Publication 2022-04-29 23:20:06 UTC

Contents

`angristlavy` *Angrist and Lavy (1999) Maimonides' Rule Data Set*

Description

Angrist and Lavy (1999) data set used by Rosenbaum (2010) to demonstrate his instrumental variable sensitivity analysis.

Usage

```
data(angristlavy)
```

Format

A `data.frame` with 172 observations on the following variables).

clasz: Size of class or classes for each cohort.

avgmath: Average math test score for each class.

z: A recode of enrollment with 1 indicating a cohort with 41 or more students, i.e. two classes. This serves as the instrument which encourages smaller classes.

pair: matched pair id

Source

Angrist, Joshua and Lavy, Victor (1999). "Using Maimonides' Rule to Estimate the Effect of Class Size on Scholastic Achievement." *Quarterly Journal of Economics* 114, 533–575.

Rosenbaum, Paul R. (2010). *Design of Observational Studies*. Springer-Verlag.

References

Rosenbaum, Paul R. (2010). *Design of Observational Studies*. Springer-Verlag.

`binarysens` *Sensitivity Test for Matched Binary Outcome*

Description

Function to calculate Rosenbaum bounds for binary data.

Usage

```
binarysens(x,y, Gamma=6, GammaInc=1)
```

Arguments

`x`	Count of the first set of discrepant pairs in a table of treated and control outcomes.
`y`	Count of the second set of discrepant pairs in a table of treated and control outcomes.
`Gamma`	Upper-bound on gamma parameter.
`GammaInc`	To set user specified increments for gamma parameter.

Author(s)

Luke Keele, University of Pennsylvania, `<luke.keele@gmail.com>`

References

Rosenbaum, Paul R. (2002) *Observational Studies.* Springer-Verlag.

See Also

See also `psens`, `hlsens`, `mcontrol`

Examples

```
# Example From Rosenbaum Observational Studies Pg 112
# Success: Died From Lung Cancer
# 110 Discrepant Pairs
# 12 Discrepant Pairs

# Sensitivity Test
binarysens(12,110)
```

`FisherSens` *Rosenbaum Sensitivity Analysis for Fisher's Exact Test*

Description

Calculates sensitivity to hidden bias for Fisher's exact test for a two-by-two contingency table, following the method described in Rosenbaum (2002, sec. 4.4).

Usage

```
FisherSens(totalN, treatedN, totalSuccesses, treatedSuccesses, Gammas)
```

Arguments

`totalN`	total number of observations
`treatedN`	number of treated observations
`totalSuccesses`	total number of “successes”
`treatedSuccesses`	number of successes in treatment group
`Gammas`	vector of Gammas (bounds on the differential odds of treatment) at which to test the significance of the results

Value

Returns a matrix with three columns and number of rows equal to the length of "Gammas". Each row indicates the upper and lower bounds for the (one-sided) p-value for a given value of Gamma.

Author(s)

Devin Caughey, MIT, `<caughey@mit.edu>`

See Also

See also `binarysens`, `hlsens`, `mcontrol`

Examples

```
## Fisher's Lady Tasting Tea: milk first or tea first?
LadyTastingTea <- matrix(c(4, 0, 0, 4), nrow = 2,
                         dimnames = list(Guess = c("Milk", "Tea"),
                                         Truth = c("Milk", "Tea")))
## Define "Milk" as "treated"/"success"
FisherSens(totalN = sum(LadyTastingTea),
           treatedN = sum(LadyTastingTea["Milk", ]),
           totalSuccesses = sum(LadyTastingTea[, "Milk"]),
           treatedSuccesses = sum(LadyTastingTea["Milk", "Milk"]),
           Gammas = seq(1, 2, .2))
## Interpretation: Rejection of the null hypothesis
## (that the lady cannot discriminate between milk-first and tea-first)
## is insensitive to bias as large as Gamma = 2.
```

`hlsens` *Rosenbaum Bounds for Hodges-Lehmann Point Estimate*

Description

Function to calculate Rosenbaum bounds for continuous or ordinal outcomes based on Hodges-Lehmann point estimate.

Usage

```
# Default Method
hlsens(x, y, pr = 0.1, Gamma = 6, GammaInc = 1)
```

Arguments

`x`	Treatment group outcomes in same order as treatment group.
`y`	Control group outcomes in same order as treatment group.
`pr`	Search precision parameter.
`Gamma`	Upper-bound on gamma parameter.
`GammaInc`	To set user specified increments for gamma parameter.

Details

For large data sets this function can be quite slow if pr is set to low. If the data set is larger, it is best to set pr to .5 before trying values such as .01. Generally, the results from the function are insensitive to the value for pr.

Author(s)

Luke Keele, University of Pennsylvania, <`luke.keele@gmail.com`>

References

Rosenbaum, Paul R. (2002) *Observational Studies*. Springer-Verlag.

See Also

See also `binarysens`, `psens`, `mcontrol`

Examples

```
# Replication of Rosenbaum Sensitivity Tests From Chapter 4 of
# Observational Studies

# Data: Matched Data of Lead Blood Levels in Children
trt <- c(38, 23, 41, 18, 37, 36, 23, 62, 31, 34, 24, 14, 21, 17, 16, 20,
15, 10, 45, 39, 22, 35, 49, 48, 44, 35, 43, 39, 34, 13, 73, 25, 27)

ctrl <- c(16, 18, 18, 24, 19, 11, 10, 15, 16, 18, 18, 13, 19, 10, 16,
16, 24, 13, 9, 14, 21, 19, 7, 18, 19, 12, 11, 22, 25, 16, 13, 11, 13)

hlsens(trt, ctrl)
```

`iv_sens` *Function to calculate Rosenbaum bounds for IV Estimator based on Wilcoxon sign rank test.*

Description

`iv_sens` performs a non-parametric, instrumental variable sensitivity analysis on matched pairs following the logic of the Neyman-Rubin framework for causal inference. The function supports a variable-valued instrument.

Usage

```
iv_sens(Rt, Rc, Dt, Dc, Gamma = 6, GammaInc = 1)
```

Arguments

`Rt, Rc`	Vectors of observed response outcomes for matched treatment and control observations, respectively.
`Dt, Dc`	Vectors of observed doses for matched observations, respectively. This is level of dose encouraged by the instrument.
`Gamma`	Upper-bound on gamma parameter.
`GammaInc`	To set user specified increments for gamma parameter.

Details

Given matched pairs of observations on an instrument `Z`, which encourages dose `D`, this function performs a Rosenbaum's bounds sensitivity analysis. Note that matching is done on levels of the instrument.

Value

Returns an object of class `rbounds`.

Author(s)

Luke Keele, University of Pennsylvania, `<luke.keele@gmail.com>`

References

Angrist, Joshua D., Imbens, Guido W., and Rubin, Donald B. (1996). "Identification of Causal Effects Using Instrumental Variables." *Journal of the American Statistical Association* 91/434, pp. 444–455.

Rosenbaum, Paul R. (1996). "Comment." *Journal of the American Statistical Association* 91/434, pp. 465–468.

Rosenbaum, Paul R. (2002). *Observational Studies*. Springer-Verlag.

Rosenbaum, Paul R. (2010). *Design of Observational Studies*. Springer-Verlag.

See Also

See also `binarysens`, `hlsens`, `mcontrol`

Examples

```
data(angristlavy)

# Example from Ch 5 of Design of Observational Studies

#Extract Matched Outome Data
Rt <- angristlavy$avgmath[angristlavy$z==1]
Rc <- angristlavy$avgmath[angristlavy$z==0]

#Extract Matched Doses
#Doses Encouraged By Instrument - Here Class Size
Dt <- angristlavy$clasz[angristlavy$z==1]
Dc <- angristlavy$clasz[angristlavy$z==0]


#Run Sensitivity Analsyis - Rank Sum Test
iv_sens(Rc, Rt, Dc, Dt, Gamma=1.5, GammaInc=.01)
```

`mcontrol` *Sensitivity Analysis For Multiple Matched Controls*

Description

Function to calculate Rosenbaum bounds for continuous or ordinal outcomes based on Wilcoxon sign rank test *p*-value when there are multiple matched control units.

Usage

```
# Default Method
mcontrol(y, grp.id, treat.id, group.size = 3, Gamma = 4, GammaInc = 1)
```

Arguments

`y`	Vector of grouped matched outcomes.
`treat.id`	A vector indicating the treated unit in each matched group.
`grp.id`	A vector indicating matched groups.
`group.size`	The size of the matched groups. Three for one treated unit and two control units.
`Gamma`	Upper-bound on gamma parameter.
`GammaInc`	To set increments for gamma parameter.

Details

The matched data needs to be in a very particular form for this function to work. The data must be sorted by matched groups with indicators for each matched group and for treated and control units. The simplest way to prepare the data is to use the Match() function and use the data.prep() function to format the data.

Currently this function only takes matched data with 2 or 3 controls units matched to each treated unit.

This function does cannot handle data where the number of control units is not the same for every treated unit.

Author(s)

Luke Keele, University of Pennsylvania, `<luke.keele@gmail.com>`

References

Rosenbaum, Paul R. (2002) *Observational Studies*. Springer-Verlag.

See Also

See also `binarysens`, `psens`, `hlsens`

Examples

```
grp <- c(1, 1, 1, 2, 2, 2, 3, 3, 3, 4, 4, 4, 5, 5, 5, 6,
         6, 6, 7, 7, 7, 8, 8, 8, 9, 9, 9, 10,10,10,11,
         11,11,12,12,12)
trt <- rep(c(1,0,0), 12)
score <- c(-11.39,-8.45, -19.57,-8.33, 3.06, -19.93,-18.73,-11.99,-7.55,
            11.94, 9.4, -25.16,-0.77, -10.46,-7.27, 24.03, -8.23, 2.67,
            -4.04, -6.67, -1.12, -14.4, -26.21,5, -1.7, -15.3, -7.73,
            -0.87, -19.71,-12.69, -3.36, -11.21,-35.83,5.89, -10.79,2)


mcontrol(score, grp, trt, group.size=3)
```

`print.rbounds` *Summarize output from rbounds object*

Description

Summary method for `rbounds` object.

Usage

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

Arguments

`x`	An object of class `rbounds` as produced by `binarysens`, `hlsens`, or `psens`.
`...`	Any additional arguments.

Author(s)

Jason W. Morgan, Ohio State University, `<morgan.746@osu.edu>`

See Also

Also see `binarysens`, `psens`, and `hlsens`.

`psens` *Rosenbaum Bounds for Sign Rank*

Description

Function to calculate Rosenbaum bounds for continuous or ordinal outcomes based on Wilcoxon sign rank test.

Usage

```
# Default Method
psens(x, y, Gamma = 6, GammaInc = 1)
```

Arguments

`x`	Treatment group outcomes in same order as treatment group.
`y`	Control group outcomes in same order as treatment group.
`Gamma`	Upper-bound on gamma parameter.
`GammaInc`	To set user-specified increments for gamma parameter.

Author(s)

Luke Keele, University of Pennsylvania, `<luke.keele@gmail.com>`

References

Rosenbaum, Paul R. (2002) *Observational Studies*. Springer-Verlag.

See Also

See also `binarysens`, `hlsens`, `mcontrol`

Examples

```
# Replication of Rosenbaum Sensitivity Tests From Chapter 4 of
# Observational Studies

# Data:  Matched Data of Lead Blood Levels in Children
trt <- c(38, 23, 41, 18, 37, 36, 23, 62, 31, 34, 24, 14, 21, 17, 16, 20,
         15, 10, 45, 39, 22, 35, 49, 48, 44, 35, 43, 39, 34, 13, 73, 25,
         27)
ctrl <- c(16, 18, 18, 24, 19, 11, 10, 15, 16, 18, 18, 13, 19, 10, 16,
          16, 24, 13, 9, 14, 21, 19, 7, 18, 19, 12, 11, 22, 25, 16, 13,
          11, 13)

psens(trt, ctrl)
```

`SumTestSens` *Rosenbaum Sensitivity Analysis for Unmatched Groups*

Description

Calculates sensitivity to hidden bias for tests based on sum statistics (e.g., Wilcoxon's rank sum test), following the method described by Rosenbaum (2002, sec. 4.6). It is meant for unmatched/unstratified data with ordinal or continuous responses.

Usage

```
SumTestSens(T, q, n, m, Gamma)
```

Arguments

`T`	observed value of the test statistic (e.g., the sum of the ranks of the responses of the treated units; note that a higher rank corresponds to a higher response)
`q`	vector of functions of the responses (e.g., their ranks), sorted in decreasing order
`n`	total number of observations
`m`	number treated units
`Gamma`	scalar indicating upper limit on the ratio of the a priori odds of treatment assignment between the treated and control groups

Value

This function prints the upper bound of the normal approximation one-sided p-value for the test at the given value of Gamma. It also invisibly returns a list of intermediate statistics.

Warning

Since 'SumTestSens' calculates through enumeration the exact expectation and variance of the test under the null, it is very computationally intensive and may be unworkable for even medium-sized datasets.

Author(s)

Devin Caughey, MIT, <caughey@mit.edu>

References

Paul R. Rosenbaum. Observational Studies. Springer, New York, 2nd edition, 2002, sec. 4.6

See Also

See also `binarysens`, `hlsens`, `mcontrol`

Examples

```
## Example from Rosenbaum (2002, p.~146)
  mercury <- data.frame(matrix(c(1, 0, 2.7,    5.3,
                                2, 0, 0.5,   15.0,
                                3, 0, 0.0,   11.0,
                                4, 0, 0.0,    5.8,
                                5, 0, 5.0,   17.0,
                                6, 0, 0.0,    7.0,
                                7, 0, 0.0,    8.5,
                                8, 0, 1.3,    9.4,
                                9, 0, 0.0,    7.8,
                               10, 0, 1.8,   12.0,
                               11, 0, 0.0,    8.7,
                               12, 0, 0.0,    4.0,
                               13, 0, 1.0,    3.0,
                               14, 0, 1.8,   12.2,
                               15, 0, 0.0,    6.1,
                               16, 0, 3.1,   10.2,
                               17, 1, 0.7,  100.0,
                               18, 1, 4.6,   70.0,
                               19, 1, 0.0,  196.0,
                               20, 1, 1.7,   69.0,
                               21, 1, 5.2,  370.0,
                               22, 1, 0.0,  270.0,
                               23, 1, 5.0,  150.0,
                               24, 1, 9.5,   60.0,
                               25, 1, 2.0,  330.0,
                               26, 1, 3.0, 1100.0,
                               27, 1, 1.0,   40.0,
                               28, 1, 3.5,  100.0,
                               29, 1, 2.0,   70.0,
                               30, 1, 5.0,  150.0,
                               31, 1, 5.5,  200.0,
                               32, 1, 2.0,  304.0,
                               33, 1, 3.0,  236.0,
                               34, 1, 4.0,  178.0,
                               35, 1, 0.0,   41.0,
                               36, 1, 2.0,  120.0,
                               37, 1, 2.2,  330.0,
                               38, 1, 0.0,   62.0,
```

```
                              39, 1, 2.0,   12.8),
                             nrow = 39, ncol = 4, byrow = TRUE))
colnames(mercury) <- c("ID", "Tr", "Pct.cu.cells", "Hg.in.blood")

(T_test <- rank(mercury$Hg.in.blood) %*% mercury$Tr)
(q_test <- sort(rank(mercury$Hg.in.blood), decreasing = TRUE))
(n_test <- nrow(mercury))
(m_test <- sum(mercury$Tr))

## Note: since this function uses exact rather than approximate
## formulas for the mean and variance of T, the p-values it
## calculates do not precisely match those in Rosenbaum (2002).

#A single Gamma value - example not run
#testOut2 <- SumTestSens(T = T_test,
#                          q = q_test,
#                          n = n_test,
#                          m = m_test,
#                          Gamma = 35)

## Apply to vector of Gamma values
sapply(c(1, 5, 35), SumTestSens,
       T = T_test, q = q_test, n = n_test, m = m_test)
```

Index