

Package ‘rbounds’

October 23, 2009

Type Package

Title Perform Rosenbaum bounds sensitivity tests for matched data.

Version 0.4

Date 2009-10-21

Author Luke J. Keele

Maintainer Luke J. Keele <keele.4@osu.edu>

Depends R (>= 2.7.1), Matching (>= 4.6-2)

Description Takes matched data and calculates Rosenbaum bounds for the treatment effect. Calculates bounds for binary data, Hodges-Lehmann point estimates, Wilcoxon signed-rank test, and for data with multiple matched controls. Is designed to work with the Matching package and operate on Match() objects.

License GPL (>= 2)

Repository CRAN

Date/Publication 2009-10-23 06:41:12

R topics documented:

binarysens	2
data.prep	3
hlsens	5
mcontrol	7
psens	8

Index	11
--------------	-----------

binarysens

Binary Sensitivity Test

Description

Function to calculate Rosenbaum bounds for binary data.

Usage

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

Arguments

x	Number of successes in control group.
y	Number of successes in treatment group.
Gamma	Upper-bound on gamma parameter.
GammaInc	To set user specified increments for gamma parameter.

Author(s)

Luke Keele, Ohio State University, <keele.4@osu.edu>

References

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

See Also

See also [data.prep](#), [psens](#), [hlsens](#), [Match](#), [mcontrol](#)

Examples

```
#
# Example From Rosenbaum Observational Studies Pg 112
# Success: Died From Lung Cancer
# 110 Treated Successes
# 12 Control Successes
#
# Sensitivity Test
#
binarysens(12,110)

#
# Example Using Match()
#

#
#Load Matching Software and Data
```

```

#
library(Matching)
data(GerberGreenImai)

#
# Estimate Propensity Score
#
pscore.glm <- glm(PHN.C1 ~ PERSONS + VOTE96.1 + NEW +
  MAJORPTY + AGE + WARD + PERSONS:VOTE96.1 + PERSONS:NEW +
  AGE2, family = binomial(logit), data = GerberGreenImai)

#
#save data objects
#
D <- GerberGreenImai$PHN.C1
Y <- GerberGreenImai$VOTED98
X <- fitted(pscore.glm)

#
# Match - without replacement
#
m.obj <- Match(Y = Y, Tr = D, X = X, M = 1, replace=FALSE)
summary(m.obj)

#
# One should check balance, but let's skip that step for now.
#

#
# Sensitivity Test
#
binarysens(m.obj, Gamma=2, GammaInc=.1)

```

data.prep

Format Match() object for sensitivity test.

Description

This function reshapes the output from Match() to create the necessary objects for mcontrol()

Usage

```

#Default Method
data.prep(obj, Y=NULL, group.size=3)

```

Arguments

<code>obj</code>	An object from the <code>Match()</code> function.
<code>Y</code>	Internal argument to the <code>Match</code> object.
<code>group.size</code>	The size of the matched groups. Three for one treated unit and two control units.

Details

This functions takes a `Match()` object and formats it for use with the `mcontrol()` function. The output is a list with the three objects needed for the arguments of the `mcontrol()` function.

Value

<code>Y</code>	The matched outcomes
<code>id</code>	A vector which identifies the matched groups: 1, 1, 1 for matched group one; 2, 2, 2 for match group 2, etc.
<code>treat</code>	A vector with 1's for treated units and 0's for control units

Author(s)

Luke Keele, Ohio State University, <keele.4@osu.edu>

References

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

See Also

See also [binarysens](#), [psens](#), [hlsens](#), [Match](#), [mcontrol](#)

Examples

```
#
#Load Matching Software and Data
#
library(Matching)
data(lalonde)

#
# Estimate Propensity Score
#
DWglm <- glm(treat~age + I(age^2) + educ + I(educ^2) + black +
             hisp + married + nodegr + re74 + I(re74^2) + re75 + I(re75^2) +
             u74 + u75, family=binomial, data=lalonde)

#
#save data objects
#
Y <- lalonde$re78 #the outcome of interest
Tr <- lalonde$treat #the treatment of interest
```

```

#
# Match
#
mDW <- Match(Y=Y, Tr=Tr, X=DWglm$fitted, M=2)

#
# One should check balance, but let's skip that step for now.
#

#Create Data Object
tmp <- data.prep(mDW, group.size=3)

#
# Sensitivity Test
#
mcontrol(tmp$Y, tmp$id, tmp$treat, group.size=3)

```

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=NULL, pr=.1, Gamma=6, GammaInc=1)

```

Arguments

x	Treatment group outcomes in same order as treatment group outcomes or an objects from Match().
y	Control group outcomes in same order as treatment group outcomes unnecessary when using Match() object.
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, Ohio State University, <keele.4@osu.edu>

References

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

See Also

See also [data.prep](#), [binarysens](#), [psens](#), [Match](#), [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)

ctrl <- c(16,18,18,24,19,11,10,15,16,18,18,13,19,10,16,16,24,13,9,14,21,19,7,18,19,12,11,22,20)

hlsens(trt, ctrl)

#
#Load Matching Software and Data
#
library(Matching)
data(lalonde)

#
# Estimate Propensity Score
#
DWglm <- glm(treat~age + I(age^2) + educ + I(educ^2) + black +
             hisp + married + nodegr + re74 + I(re74^2) + re75 + I(re75^2) +
             u74 + u75, family=binomial, data=lalonde)

#
#save data objects
#
Y <- lalonde$re78 #the outcome of interest
Tr <- lalonde$treat #the treatment of interest

#
# Match - without replacement
#
mDW <- Match(Y=Y, Tr=Tr, X=DWglm$fitted, replace=FALSE)

#
# One should check balance, but let's skip that step for now.
#

#
# Sensitivity Test
```

```
#
hlsens(mDW, pr=.1, Gamma=2, GammaInc=.1)
```

`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

<code>y</code>	Vector of grouped matched outcomes.
<code>treat.id</code>	A vector indicating the treated unit in each matched group.
<code>grp.id</code>	A vector indicating matched groups.
<code>group.size</code>	The size of the matched groups. Three for one treated unit and two control units.
<code>Gamma</code>	Upper-bound on gamma parameter.
<code>GammaInc</code>	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 not handle data where the number of control units is not the same for every treated unit.

Author(s)

Luke Keele, Ohio State University, <keele.4@osu.edu>

References

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

See Also

See also [data.prep](#), [binarysens](#), [psens](#), [Match](#), [hlsens](#)

Examples

```
#
#Load Matching Software and Data
#
library(Matching)
data(lalonde)

#
# Estimate Propensity Score
#
DWglm <- glm(treat~age + I(age^2) + educ + I(educ^2) + black +
             hisp + married + nodegr + re74 + I(re74^2) + re75 + I(re75^2) +
             u74 + u75, family=binomial, data=lalonde)

#
#save data objects
#
Y <- lalonde$re78 #the outcome of interest
Tr <- lalonde$treat #the treatment of interest

#
# Match - without replacement
#
mDW <- Match(Y=Y, Tr=Tr, X=DWglm$fitted, M=2, replace=FALSE)

#
# One should check balance, but let's skip that step for now.
#

#Create Data Object
tmp <- data.prep(mDW, group.size=3)

#
# Sensitivity Test
#
mcontrol(tmp$Y, tmp$id, tmp$treat, group.size=3)
```

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=NULL, Gamma=6, GammaInc=1)
```

Arguments

x	Treatment group outcomes in same order as treatment group outcomes or an objects from Match().
y	Control group outcomes in same order as treatment group outcomes unnecessary when using Match() object.
Gamma	Upper-bound on gamma parameter.
GammaInc	To set user specified increments for gamma parameter.

Author(s)

Luke Keele, Ohio State University, <keele.4@osu.edu>

References

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

See Also

See also [data.prep](#), [binarysens](#), [hlsens](#), [Match](#), [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)

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

psens(trt, ctrl)

#Example With Match()

#
#Load Matching Software and Data
#
library(Matching)
data(lalonde)

#
# Estimate Propensity Score
#
DWglm <- glm(treat~age + I(age^2) + educ + I(educ^2) + black +
```

```
      hisp + married + nodegr + re74 + I(re74^2) + re75 + I(re75^2) +
      u74 + u75, family=binomial, data=lalonde)

#
#save data objects
#
Y <- lalonde$re78 #the outcome of interest
Tr <- lalonde$treat #the treatment of interest

#
# Match - without replacement
#
mDW <- Match(Y=Y, Tr=Tr, X=DWglm$fitted, replace=FALSE)

#
# One should check balance, but let's skip that step for now.
#

#
# Sensitivity Test
#
psens(mDW, Gamma=2, GammaInc=.1)
```

Index

binarysens, 1, 4, 6, 7, 9

data.prep, 2, 3, 6, 7, 9

hlsens, 2, 4, 5, 7, 9

Match, 2, 4, 6, 7, 9

mcontrol, 2, 4, 6, 6, 9

psens, 2, 4, 6, 7, 8