Package ‘thickmatch’

April 11, 2020

Type Package
Title Threshold Matching for Thick Description for Optimal Matching in Observational Studies
Version 0.3.1
Author Ruoqi Yu
Maintainer Ruoqi Yu <ruoqiyu@wharton.upenn.edu>
License MIT+file LICENSE
Encoding UTF-8
LazyData true
Imports rcbalance, stats, MASS, plyr, methods, DiPs
Suggests optmatch
Note One minimum cost flow problem may have several or many solutions that are equivalent in providing the same minimum total or mean cost. Minor differences between computers or implementations may have the minor consequence of altering which equivalent solution is produced.
NeedsCompilation no
Repository CRAN
Date/Publication 2020-04-11 13:30:02 UTC

R topics documented:

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>dmaha</td>
<td>2</td>
</tr>
<tr>
<td>feasible</td>
<td>3</td>
</tr>
<tr>
<td>netvr</td>
<td>5</td>
</tr>
<tr>
<td>nysr</td>
<td>6</td>
</tr>
<tr>
<td>threshold</td>
<td>9</td>
</tr>
<tr>
<td>threshold_match</td>
<td>11</td>
</tr>
</tbody>
</table>

Index 14
dmaha

*Create a Mahalanobis distance for matching based on a dense network.*

**Description**

Computes a Mahalanobis distance list, either the traditional version or the rank-based version, for use in dense matching, i.e. the distance for all possible pairs of treated and control.

This function and its use are discussed in Rosenbaum (2010). The rank-based Mahalanobis distance is described in Chapter 8 of Rosenbaum (2010).

**Usage**

```r
dmaha(z, X, min.control=1, exact=NULL, nearexact=NULL, penalty=1000, rank=FALSE)
```

**Arguments**

- `z` A vector whose ith coordinate is 1 for a treated unit and is 0 for a control.
- `X` A matrix with length(z) rows giving the covariates. X should be of full column rank.
- `min.control` A positive integer giving the minimum number of controls to be matched to each treated subject. If min.control is too large, the match will be infeasible.
- `exact` If not NULL, then a vector of length(z)=length(p) giving variable that need to be exactly matched.
- `nearexact` If not NULL, then a vector of length length(z) giving variable that need to be exactly matched.
- `penalty` The penalty for a mismatch on nearexact.
- `rank` If rank=TRUE, a rank-based Mahalanobis distance will be calculated. Otherwise (with default value FALSE), a traditional Mahalanobis distance will be computed.

**Details**

The usual Mahalanobis distance works well for multivariate Normal covariates, but can exhibit odd behavior with typical covariates. Long tails or an outlier in a covariate can yield a large estimated variance, so the usual Mahalanobis distance pays little attention to large differences in this covariate. Rare binary covariates have a small variance, so a mismatch on a rare binary covariate is viewed by the usual Mahalanobis distance as extremely important. If you were matching for binary covariates indicating US state of residence, the usual Mahalanobis distance would regard a mismatch for Wyoming as much worse than a mismatch for California.

The robust Mahalanobis distance uses ranks of covariates rather than the covariates themselves, but the variances of the ranks are not adjusted for ties, so ties do not make a variable more important. Binary covariates are, of course, heavily tied.
**feasible**

---

**Value**

- **d**: A distance object for each pair of treated and control.
- **start**: The treated subject for each distance.
- **end**: The control subject for each distance.

**References**


**Examples**

```r
data("nysr")
attach(nysr)
X<-cbind(family.income,family.structure,highest.education.parent.in.household,
    female,race.black,race.hispanic,age.teenager,school.dropout)
dist<-dmaha(intense,X)
head(dist$d)
detach(nysr)
```

**feasible** *Feasibility of a chosen threshold.*

**Description**

The program determines whether it is possible to find at least `select_num` matched pairs with distance not exceeding `eps`.

**Usage**

```r
feasible(z,X,p,caliper,dat,ncontrol=1,exact=NULL,
    nearexact=NULL,fine=rep(1,length(z)),penalty=1000,nearexpenalty=100,
    rank=FALSE,select_num=0,eps=1000)
```

**Arguments**

- **z**: A vector whose ith coordinate is 1 for a treated unit and is 0 for a control.
- **X**: A matrix with `length(z)` rows giving the covariates. `X` should be of full column rank.
- **p**: A vector of with `length(z)=length(p)` giving the variable used to define the caliper. For instance, `p` might be the propensity score.
- **caliper**: If the treated-minus-control difference (in the scale of `sd(p)`) in `p` is `< -caliper` or `> caliper`, then penalty is added to the distance.
- **dat**: A data frame with `length(z)` rows. If the match is feasible, the matched portion of `dat` is returned with additional columns that define the match.
ncontrol  A positive integer giving the number of controls to be matched to each treated subject. If ncontrol is too large, the match will be infeasible.

exact     If not NULL, then a vector of length(z)=length(p) giving variable that need to be exactly matched.

nearexact If not NULL, then a vector of length length(z) giving variable that need to be exactly matched.

fine      A vector of with length(z)=length(fine) giving the nominal levels that are to be nearly-finely balanced.

penalty   A numeric penalty imposed for each violation of fine balance.

nearexpenalty The penalty for a mismatch on nearexact.

rank      If rank=TRUE, a rank-based Mahalanobis distance will be calculated. Otherwise (with default value FALSE), a traditional Mahalanobis distance will be computed.

select_num A positive number giving the required number of matched pairs with distance not exceeding eps.

eps       The threshold whose feasibility is examined.

Details

If there is a feasible matching with at least select_num matched pairs with distance not exceeding eps, then eps is said to be feasible and 1 is returned. Otherwise, eps is said to be infeasible and 0 is returned.

For details, see Rosenbaum (2017).

You MUST install and load the optmatch package to use feasible.

Value

If the match is infeasible, a warning is issued. Otherwise, a binary indicator of whether there are at least select_num matched pairs with distance not exceeding eps.

A match may be infeasible if ncontrol is too large, or if exact matching for exact is impossible.

References


Examples

# To run this example, you must load the optmatch package.
data("nysr")attach(nysr)X<-cbind(family.income,family.structure,highest.education.parent.in.household,female,race.black,race.hispanic,age.teenager,school.dropout)detach(nysr)feasible(z=nysr$intense,X=X,p=nysr$plogit,caliper=0.2,dat=nysr,select_num=10,eps=0.5)
netvr

*Optimal variable-ratio match from a distance matrix.*

**Description**

The function creates the network for optimal variable-ratio matching to be passed via callrelax to the Fortran code for Bertsekas and Tseng’s (1988) Relax IV.

Of limited interest to most users; function netvr() would typically be called by some other functions.

**Usage**

```r
netvr(z, dist, min.control=1, max.control=min.control,
      total.control=sum(z)*min.control,
      fine=rep(1,length(z)), penalty=1000)
```

**Arguments**

- `z`: A vector whose ith coordinate is 1 for a treated unit and is 0 for a control.
- `dist`: A distance list with the starting node (treated subjec), ending node (control), the extra distance between them.
- `min.control`: A positive integer giving the minimum number of controls to be matched to each treated subject. If min.control is too large, the match will be infeasible.
- `max.control`: A positive integer giving the maximum number of controls to be matched to each treated subject.
- `total.control`: A positive integer giving the total number of controls to be matched to each treated subject. If total.control is too large, the match will be infeasible.
- `fine`: A vector of with length(z)=length(fine) giving the nominal levels that are to be nearly-finely balanced.
- `penalty`: A numeric penalty imposed for each violation of fine balance.

**Details**

The network contains a bipartite graph for treated and control subjects plus additional nodes for fine balance categories, plus additional nodes accept needed deviations from fine balance yielding near-fine balance.

**Value**

A network for optimal variable-ratio matching.

**References**


Available in R via the optmatch package.
Description

NYSR data on adolescent work intensity and substance use.

Usage

data("nysr")

Format

A data frame with 2816 observations on the following 18 variables.

IDS  NYSR identification number
intense  Based on question “During the school year, about how many hours per week did you normally work at a paid job, or did you not have a job”. “Never”: student did not have a job; “Moderate”: 1-19 hours; “Intense”: >=20 hours.
family.income  Household income with 5000 = (between 0-10,000), 15000 = (between 10,000 and 20,000), ..., 95000 = (between 90,000 and 100,000) and 105,000 (above 100,000).
family.income.impute  Household income with 5000 = (between 0-10,000), 15000 = (between 10,000 and 20,000), ..., 95000 = (between 90,000 and 100,000) and 105,000 (above 100,000). For subjects with missing family income, the mean is imputed.
family.income.mis  dummy variable for whether household income is missing and the mean is imputed.
family.structure  “Two Parent Biological”: both biological father and mother living with child; “Two Parent Nonbiological”: someone assuming a mother role (biological, adoptive, stepparent) living with a husband who assumes a father role (biological, adoptive, step parent) where both parents are biological; “Single Parent/Other”: any other living situation for child.
highest.education.parent.in.household  Maximum education level of household resident who assumes a mother role (biological, adoptive, stepparent) and household resident who assumes a father role (biological, adoptive, stepparent). If the child is living with a single parent, then this is just the education level of that single parent.
highest.education.parent.in.household.impute  Maximum education level of household resident who assumes a mother role (biological, adoptive, stepparent) and household resident who assumes a father role (biological, adoptive, stepparent). If the child is living with a single parent, then this is just the education level of that single parent. For subjects with missing highest education of parent in household, the mean is imputed.
highest.education.parent.in.household.mis  Dummy variable for whether household income is missing and the mean is imputed.
female  1 = female, 0 = male
race.black  1=black race, 0=other
race.hispanic  1=hispanic race, 0=other
age. teenager  age of teenager. Age is imputed with the mean if it is missing

school.dropout Dummy variable of whether student has dropped out of school

alcohol.use Based on question “How often, if at all, do you drink alcohol, such as beer, wine or mixed drinks, not including at religious services”. “Never”: answered “Never”; “Moderate”: answered “A few times year” or “About once a month”; “Heavy”: answered “A few times a month”, “About once a week”, “A few times a week” or “Almost every day”.

marijuana.use Based on question “How often, if ever, have you used marijuana?”. “Never”: answered “Never”; “Experimenter” answered “You tried it once or twice”; “Continual User”: answered “You use it occasionally” or “You use it regularly”.

p Propensity score.

plogit Logit of propensity score.

Details

The following code constructed the data as used here. 

```R
# Construct family income
wave1data$family.income = rep(NA, nrow(wave1data))
wave1data$family.income[wave1data$PINCOME1 == 1 & wave1data$PINCOME2 == 4] = 5000
wave1data$family.income[wave1data$PINCOME1 == 1 & wave1data$PINCOME2 == 3] = 15000
wave1data$family.income[wave1data$PINCOME1 == 1 & wave1data$PINCOME2 == 2] = 25000
wave1data$family.income[wave1data$PINCOME1 == 1 & wave1data$PINCOME2 == 1] = 45000
wave1data$family.income[wave1data$PINCOME1 == 2 & wave1data$PINCOME3 == 3] = 55000
wave1data$family.income[wave1data$PINCOME1 == 2 & wave1data$PINCOME3 == 2] = 65000
wave1data$family.income[wave1data$PINCOME1 == 2 & wave1data$PINCOME3 == 1] = 75000
wave1data$family.income[wave1data$PINCOME1 == 2 & wave1data$PINCOME3 == 4] = 85000
wave1data$family.income[wave1data$PINCOME1 == 2 & wave1data$PINCOME3 == 5] = 95000
wave1data$family.income[wave1data$PINCOME1 == 2 & wave1data$PINCOME3 == 6] = 105000

# For subjects with missing family income data, fill in mean and create a missing data indicator
wave1data$family.income.mis = is.na(wave1data$family.income)

# Find family structure variable
wave1data$family.structure = rep(NA, nrow(wave1data))

# Two Parent Biological
wave1data$family.structure[wave1data$PMOTHER == 1 & wave1data$PLIVE == 1 & wave1data$PSPRELAT == 1] = "Two Parent Biological"

# Two Parent Nonbiological
wave1data$family.structure[(wave1data$PMOTHER == 2 | wave1data$PMOTHER == 3) & (wave1data$PSPRELAT == 1 | wave1data$PSPRELAT == 2 | wave1data$PSPRELAT == 3)] = "Two Parent Nonbiological"
```

# For single parent
wave1data$family.structure[is.na(wave1data$family.structure)] = "Single Parent/Other"

# Single Parent
wave1data$family.structure[wave1data$PMOTHER == 1 & wave1data$PLIVE == 1 & wave1data$PSPRELAT == 1] = 1
wave1data$family.structure[wave1data$PMOTHER == 1 & wave1data$PLIVE == 2 & wave1data$PSPPARTPAR == 1] = 1
wave1data$family.structure[wave1data$PMOTHER == 1 & wave1data$PLIVE == 2 & wave1data$PSPPARTPAR == 2] = 1
wave1data$family.structure[wave1data$PMOTHER == 1 & wave1data$PLIVE == 2 & wave1data$PSPPARTPAR == 3] = 1
wave1data$family.structure[wave1data$PMOTHER == 1 & wave1data$PLIVE == 2 & wave1data$PSPPARTPAR == 4] = 1
wave1data$family.structure[wave1data$PMOTHER == 1 & (wave1data$PSPRELAT == 2 | wave1data$PSPRELAT == 3)] = 1
wave1data$family.structure[wave1data$PMOTHER == 1 & (wave1data$PSPRELAT == 2 | wave1data$PSPRELAT == 3 | wave1data$PSPRELAT == 4)] = 1
wave1data$family.structure[wave1data$PMOTHER == 1 & (wave1data$PSPRELAT == 2 | wave1data$PSPRELAT == 3 | wave1data$PSPRELAT == 4 | wave1data$PSPRELAT == 5)] = 1
wave1data$family.structure[wave1data$PMOTHER == 1 & (wave1data$PSPRELAT == 2 | wave1data$PSPRELAT == 3 | wave1data$PSPRELAT == 4 | wave1data$PSPRELAT == 5 | wave1data$PSPRELAT == 6)] = 1
wave1data$family.structure[wave1data$PMOTHER == 1 & (wave1data$PSPRELAT == 2 | wave1data$PSPRELAT == 3 | wave1data$PSPRELAT == 4 | wave1data$PSPRELAT == 5 | wave1data$PSPRELAT == 6 | wave1data$PSPRELAT == 7)] = 1
```
wave1data$PFATHER==3 & (wave1data$SPRELAT==1 | wave1data$SPRELAT==3)]=1

# Highest parent education in household
dadeductemp=rep(NA,nrow(wave1data))
dadeductemp[wave1data$PDADEDUC==0 | wave1data$PDADEDUC==1 | wave1data$PDADEDUC==2]=0
dadeductemp[wave1data$PDADEDUC==3 | wave1data$PDADEDUC==4 | wave1data$PDADEDUC==5 | wave1data$PDADEDUC==6 | wave1data$PDADEDUC==8]=2
dadeductemp[wave1data$PDADEDUC==9 | wave1data$PDADEDUC==10]=3
dadeductemp[wave1data$PDADEDUC==11 & wave1data$PDADEDUC<=14]=4
momeductemp=rep(NA,nrow(wave1data))
momeductemp[wave1data$PMOMEDUC==0 | wave1data$PMOMEDUC==1 | wave1data$PMOMEDUC==2]=0
momeductemp[wave1data$PMOMEDUC==3 | wave1data$PMOMEDUC==4 | wave1data$PMOMEDUC==5 | wave1data$PMOMEDUC==6 | wave1data$PMOMEDUC==8]=2
momeductemp[wave1data$PMOMEDUC==9 | wave1data$PMOMEDUC==10]=3
momeductemp[wave1data$PMOMEDUC==11 & wave1data$PMOMEDUC<=14]=4
parents.highest.educ=pmax(dadeductemp,momeductemp,na.rm=TRUE)

# Gender of teenager
wave1data$gender=rep(NA,nrow(wave1data))
wave1data$gender[wave1data$TEENSEX==0]="MALE"
wave1data$gender[wave1data$TEENSEX==1]="FEMALE"

# Race/ethnicity of teenager
wave1data$race.ethnicity=rep(NA,nrow(wave1data))
wave1data$race.ethnicity[wave1data$TEENRACE==1]="White/Other"
wave1data$race.ethnicity[wave1data$TEENRACE==2]="African American"
wave1data$race.ethnicity[wave1data$TEENRACE==3]="Hispanic"
wave1data$race.ethnicity[wave1data$TEENRACE>=4]="White/Other"

# Age of teenager
wave1data$age.teenager=wave1data$AGE
wave1data$age.missing=(wave1data$AGE==888)
wave1data$age.missing.mean=mean(wave1data$age.teenager,na.rm=TRUE)

## Drop from consideration for matching fifth and sixth graders; students missing work intensity, alcohol use and marijuana use; students with moderate working intensity
wave1data$not.included.in.sample=(wave1data$PSCHGRA2==5)
| wave1data$PSCHGRA2 == 6 | wave1data$age.missing == TRUE | is.na(wave1data$work.intensity) |
| is.na(wave1data$alcohol.use) | is.na(wave1data$marijuana.use) | wave1data$work.intensity == "Moderate"

# Create variable which identifies whether wave 1 interview exists for subject interviewerdata = read.csv("C:/Users/ruoqi/Desktop/Penn/research/Dylan-ThickDescription/ivlink.csv") wave1interviews = interviewerdata$ids ![interviewerdata$iver == "W3"
| interviewerdata$iver == "W4")! wave1data$wave1.interview = wave1data$IDS

wave1data$wave1.interview = wave1data$wave1.interview & (!wave1data$family.income.mis) & (!wave1data$highest.education.parent.in.household.mis)
data = wave1data dsub = data ![which(data$not.included.in.sample == FALSE),!, dim(dsub) # 2816 932
dsub = dsub ![which(dsub$work.intensity != "Moderate"),!, dim(dsub) # 2816 932 dsub$intense = rep(0, dim(dsub)[1])
dsub$intense ![which(dsub$work.intensity == "Intense"),!]=1

# propensity score dsub$family.income.impute = dsub$family.income dsub$family.income.impute ![dsub$family.income.mis == TRUE,!
| dsub$highest.education.parent.in.household.impute = dsub$highest.education.parent.in.household impute + highest.education.parent.in.household.impute + highest.education.parent.in.household.impute + race.black + race.hispanic + age.teenager + school.dropout, family = binomial(link = "logit"), data = dsub, x = TRUE)
x = subset(dsub[c("family.income.impute", "family.income.mis", "family.structure", "highest.education.parent.in.household.impute", "race.black", "race.hispanic", "age.teenager", "school.dropout")]) pred <- predict(model, newdata = x, type = "response") dsub$p = pred dsub$plogit = car::logit(pred) # boxplot(prop ~ intense, data = dsub)
nysr = dsub save(nysr, file = "nysr.rda")

Source

The National Survey of Youth and Religion.

References


Examples

data("nysr")
summary(nysr)

---

| threshold | Smallest threshold for thick description. |

Description

Finds the smallest threshold on such that a treated-control matching with that at least select_num matched pairs having distance not exceeding the threshold exists.

Usage

threshold(z, X, p, caliper, dat, ncontrol = 1, exact = NULL, nearexact = NULL, fine = NULL, penalty = 1000, nearexpenalty = 100, rank = FALSE, select_num = 0, tol = 0.1)
Arguments

- **z**: A vector whose ith coordinate is 1 for a treated unit and is 0 for a control.
- **X**: A matrix with length(z) rows giving the covariates. X should be of full column rank.
- **p**: A vector of with length(z)=length(p) giving the variable used to define the caliper. For instance, p might be the propensity score.
- **caliper**: If the treated-minus-control difference (in the scale of sd(p)) in p is < -caliper or > caliper, then penalty is added to the distance.
- **dat**: A data frame with length(z) rows. If the match is feasible, the matched portion of dat is returned with additional columns that define the match.
- **ncontrol**: A positive integer giving the number of controls to be matched to each treated subject. If ncontrol is too large, the match will be infeasible.
- **exact**: If not NULL, then a vector of length(z)=length(p) giving variable that need to be exactly matched.
- **nearexact**: If not NULL, then a vector of length length(z) giving variable that need to be exactly matched.
- **fine**: A vector of with length(z)=length(fine) giving the nominal levels that are to be nearly-finely balanced.
- **penalty**: A numeric penalty imposed for each violation of fine balance.
- **nearexpenalty**: The penalty for a mismatch on nearexact.
- **rank**: If rank=TRUE, a rank-based Mahalanobis distance will be calculated. Otherwise (with default value FALSE), a traditional Mahalanobis distance will be computed.
- **select_num**: A positive number giving the required number of matched pairs with distance not exceeding eps.
- **tol**: The tolerance. The smallest threshold is determined with an error of at most tol.

Details

The method uses binary search to find the best threshold. It applies threshold algorithm with function feasible; details see Rosenbaum (2017).

Often, we need a small and feasible threshold, and we prefer to estimate the threshold very precisely. Making tol smaller makes the number of closest pairs close to select_num.

You MUST install and load the optmatch package to use threshold.

Value

If the match is infeasible, a warning is issued. Otherwise, a list of results is returned.

A match may be infeasible if the caliper on p is too small, or ncontrol is too large, or if exact matching for exact is impossible.

- **epsilon**: The smallest threshold, with an error of at most tol. This threshold is a little too large, at most tol too large, but because its error is on the high side, a match with this threshold ensures at least select_num matched pairs with distance not exceeding epsilon.
threshold_match

interval An interval that contains the best threshold. The upper bound of the interval was returned as epsilon above.

interval.length The length of interval. By definition, length.interval<=tol.

References

Examples
# To run this example, you must load the optmatch package.

data("nysr")
attach(nysr)
X<-cbind(family.income,family.structure,highest.education.parent.in.household,
female,race.black,race.hispanic,age.teenager,school.dropout)
detach(nysr)
threshold(z=nysr$intense,X=X,p=nysr$plogit,caliper=0.2,dat=nysr,select_num=10,tol=0.00001)

threshold_match Minimum-distance threshold matching.

Description
The program finds an optimal threshold match with a given threshold on distance, plus near-fine balance, exact match and near-exact match constraints. That is, it finds a match that minimizes the penalized Mahalanobis distance.

Usage
threshold_match(z,p,caliper,X,dat,min.control=1, max.control=min.control,total.control=sum(z)*min.control, exact=NULL,fine=rep(1,length(z)),finepenalty=1000,nearexact=NULL, nearexpenalty=100,eps=NULL,penalty=10000,rank=FALSE)

Arguments
z A vector whose ith coordinate is 1 for a treated unit and is 0 for a control.
p A vector of with length(z)=length(p) giving the variable used to define the caliper. For instance, p might be the propensity score.
caliper If the treated-minus-control difference (in the scale of sd(p)) in p is < -caliper or > caliper, then penalty is added to the distance.
X A matrix with length(z) rows giving the covariates. X should be of full column rank.
threshold_match

dat A data frame with length(z) rows. If the match is feasible, the matched portion of dat is returned with additional columns that define the match.

min.control A positive integer giving the minimum number of controls to be matched to each treated subject. If min.control is too large, the match will be infeasible.

max.control A positive integer giving the maximum number of controls to be matched to each treated subject.

total.control A positive integer giving the total number of controls to be matched to each treated subject. If total.control is too large, the match will be infeasible. Fine balance constraint can be determined based on total.control.

exact If not NULL, then a vector of length(z)=length(p) giving variable that need to be exactly matched.

fine A vector of with length(z)=length(fine) giving the nominal levels that are to be nearly-finely balanced.

finepenalty A numeric penalty imposed for each violation of fine balance.

nearexact If not NULL, then a vector of length length(z) giving variable that need to be exactly matched.

nearexpenalty The penalty for a mismatch on nearexact.

eps The threshold whose feasibility is examined. If eps is NULL, the conventional optimal match with the propensity score caliper, fine balance, exact and near-exact match constraints is returned.

penalty A numeric penalty imposed for each distance greater than eps.

rank If rank=TRUE, a rank-based Mahalanobis distance will be calculated. Otherwise (with default value FALSE), a traditional Mahalanobis distance will be computed.

Details

The match minimizes the total distance between treated subjects and their matched controls subject to a threshold which imposes a penalty on distances above the threshold.

For discussion of the choice of threshold, see Rosenbaum (2017).

You MUST install and load the optmatch package to use threshold_match.

Value

If the match is infeasible, a warning is issued. Otherwise, a list of results is returned.

A match may be infeasible if min.control or total.control is too large, or if exact matching for exact is impossible.

data The matched sample, selected rows of dat.

data The matched closest pairs, selected rows of dat.

balance Balance table of the matched sample, including 5 columns: treated mean, matched control mean, all control mean, matched SMD and all SMD.

sbalance Balance table of the matched closest pairs, including 5 columns: treated mean, matched control mean, all control mean, matched SMD and all SMD.
threshold_match

References


Examples

# To run this example, you must load the optmatch package.

data("nysr")
attach(nysr)
X<-cbind(family.income,family.structure,highest.education.parent.in.household,female,race.black,race.hispanic,age.teenager,school.dropout)
detach(nysr)

eps=threshold(z=nysr$intense,X=X,p=nysr$plogit,caliper=0.2,dat=nysr,select_num=10,tol=0.00001)$epsilon
m<-threshold_match(z=nysr$intense,p=nysr$plogit,caliper=0.2,X=X,dat=nysr,min.control=2,eps=eps)
dim(m$sdata)
m$sbalance
m$balance
Index

*Topic datasets
   nysr, 6

dmaha, 2
feasible, 3
netvr, 5
nysr, 6
threshold, 9
threshold_match, 11