Package: CMatching (via r-universe)

November 20, 2026

Type Package

Title Matching Algorithms for Causal Inference with Clustered Data

Version 2.4

Date 2025-02-08

Maintainer Massimo Cannas <massimo.cannas@unica.it>

Description Provides functions to perform matching algorithms for causal inference with clustered data, as described in B. Arpino and M. Cannas (2016) <doi:10.1002/sim.6880>. Pure within-cluster and preferential within-cluster matching are implemented. Both algorithms provide causal estimates with cluster-adjusted estimates of standard errors.

Depends R (>= 2.6.0), Matching

Imports stats,lmtest,multiwayvcov,lme4

License GPL-2

NeedsCompilation no

Encoding UTF-8

Author Massimo Cannas [aut, cre], Bruno Arpino [ctb], Elena Colicino [ctb]

Config/pak/sysreqs cmake make

Repository https://massimocannas.r-universe.dev

Date/Publication 2025-02-10 19:10:05 UTC

RemoteUrl https://github.com/cran/CMatching

RemoteRef HEAD

RemoteSha 92e7612e1ec008cbd750ea030dd7c4bc207de890

Contents

CMatching-package 2
CMatch 4
CMatchBalance 8

MatchPW 10
MatchW 14
schools 18
summary.CMatch 20

Index **22**

`CMatching-package` *Matching Algorithms for Causal Inference with Clustered Data*

Description

Provides functions to perform matching algorithms for causal inference with clustered data, as described in B. Arpino and M. Cannas (2016) <doi:10.1002/sim.6880>. Pure within-cluster and preferential within-cluster matching are implemented. Both algorithms provide causal estimates with cluster-adjusted estimates of standard errors.

Details

Package:	CMatching
Type:	Package
Version:	2.4
Date:	2024-02-08
License:	GPL version 3

Several strategies have been suggested for adapting propensity score matching to clustered data. Depending on researcher's belief about the strength of unobserved cluster level covariates it is possible to take into account clustering either in the estimation of the propensity score model (through the inclusion of fixed or random effects, e.g. Arpino and Mealli (2011)) and/or in the implementation of the matching algorithm (see, e.g. Rickles and Seltzer (2014); Arpino and Cannas (2016)). This package contains main function `CMatch` to adapt classic matching algorithms for causal inference to clustered data and a customized `summary` function to analyze the output. Depending on the `type` argument function `CMatch` calls either `MatchW` implementing a *pure* within-cluster matching or `MatchPW` implementing an approach which can be called *"preferential"* within-cluster matching. This approach first looks for matchable units within the same cluster and - if no match is found - continues the search in the remaining clusters. The functions also provide causal estimands with cluster-adjusted standard errors from fitting a multilevel model on matched data. `CMatch` returns an object of class "`CMatch`" which can be be summarized and used as input of the `CMatchBalance` function to examine how much the procedure resulted in improved covariate balance. Although `CMatch` has been designed for dealing with clustered data, these algorithms can be used to force a perfect balance or to improve the balance of categorical variables, respectively. In this case, the "clusters" correspond to the levels of the categorical variable(s). When used for this purpouse the user should ignore the standard error (if provided). Note that `Matchby` from package `Matching` can be used for the same purpouse.

Author(s)

Massimo Cannas [aut, cre], Bruno Arpino [ctb], Elena Colicino [ctb]. A special thanks to Thomas W. Yee for his help in updating to version 2.1.

Maintainer: Massimo Cannas <massimo.cannas@unica.it>

References

Sekhon, Jasjeet S. (2011). Multivariate and Propensity Score Matching Software with Automated Balance Optimization. *Journal of Statistical Software, 42(7): 1-52.* http://www.jstatsoft.org/v42/i07/

Arpino, B., and Cannas, M. (2016). Propensity score matching with clustered data. An application to the estimation of the impact of caesarean section on the Apgar score. *Statistics in Medicine, 35: 2074-2091*. doi: 10.1002/sim.6880.

Rickles, J. H., and Seltzer, M. (2014). A Two-Stage Propensity Score Matching Strategy for Treatment Effect Estimation in a Multisite Observational Study. *Journal of Educational and Behavioral Statistics, 39(6), 612-636*. doi: 10.3102/1076998614559748

Arpino, B. and Mealli, F. (2011). The specification of the propensity score in multilevel observational studies. *Computational Statistics & Data Analysis*, 55(4), 1770-1780. doi: 10.1016/j.csda.2010.11.008

See Also

Match, MatchBalance

Examples

```
# a paper and pencil example with a few units

id  <- c(1,2,3,4,5, 6,7,8,9,10)
 x  <- c( 1,1,1.1,1.1,1.4, 2,1,1,1.3, 1.3 )
 t  <- c( 1,1,1,1,0, 0,0,0,0, 0 )
 g  <- c(1,1,2,2,1,1,2,2,2, 2 ) # two groups of four and six units
toy <- t(data.frame(id,g, t,x))

# reorder units by ascending group
 toyord <-toy[,order(g)]
 x <-toyord["x",]
 t <-toyord["t",]
 g <- toyord["g",]

# pooled matching
pm <- Match(Y=NULL, Tr=t, X=x, caliper=2,ties=FALSE,replace=FALSE)
# quick look at matched dataset (matched pairs are vertically aligned)
pm$index.treated
pm$index.control


# within matching
wm <- CMatch(type="within",Y=NULL, Tr=t, X=x, Group=g,caliper=2,ties=FALSE,replace=FALSE)
wm$index.treated
wm$index.control
```

```
# preferential-within matching
pwm <- CMatch(type="pwithin",Y=NULL, Tr=t, X=x, Group=g, caliper=2,ties=FALSE,replace=FALSE)
pwm$index.treated
pwm$index.control
```

CMatch *Within and preferential-within cluster matching.*

Description

This function implements multivariate and propensity score matching in clusters defined by the Group variable. It returns an object of class "CMatch" which can be be summarized and used as input of the CMatchBalance function to examine how much the procedure resulted in improved covariate balance.

Usage

```
CMatch(type, Y = NULL, Tr, X, Group = NULL, estimand = "ATT", M = 1,
exact = NULL, caliper = 0.25, weights = NULL, replace = TRUE, ties = TRUE, ...)
```

Arguments

type	The type of matching desired. "within" for a pure within-cluster matching and "pwithin" for matching preferentially within. The preferential approach first searches for matchable units within the same cluster. If no match was found the algorithm searches in other clusters.
Y	A vector containing the outcome of interest.
Tr	A vector indicating the treated and control units.
X	A matrix of covariates we wish to match on. This matrix should contain all confounders or the propensity score or a combination of both.
Group	A vector describing the clustering structure (typically the cluster ID). This can be any numeric vector of the same length of Tr and X containing integer numbers in ascending order otherwise an error message will be returned. Default is NULL, however if Group is missing, NULL or it contains only one value the output of the Match function is returned with a warning.
estimand	The causal estimand desired, one of "ATE", "ATT" and "ATC", which stand for Average Treatment Effect, Average Treatment effect on the Treated and on the Controls, respectively. Default is "ATT".
M	The number of matches which are sought for each unit. Default is 1 ("one-to-one matching").
exact	An indicator for whether exact matching on the variables contained in X is desired. Default is FALSE. This option has precedence over the caliper option.

`caliper` A maximum allowed distance for matching units. Units for which no match was found within caliper distance are discarded. Default is 0.25. The caliper is interpreted in standard deviation units of the *unclustered* data for each variable. For example, if caliper=0.25 all matches at distance bigger than 0.25 times the standard deviation for any of the variables in `X` are discarded.

`weights` A vector of specific observation weights.

`replace` Matching can be with or without replacement depending on whether matches can be re-used or not. Default is TRUE.

`ties` An indicator for dealing with multiple matches. If more than M matches are found for each unit the additional matches are a) wholly retained with equal weights if ties=TRUE; b) a random one is chosen if ties=FALSE. Default is TRUE.

`...` Additional arguments to be passed to the `Match` function (not all of them can be used).

Details

This function is meant to be a natural extension of the `Match` function to clustered data. It retains the main arguments of `Match` but it has additional output showing matching results cluster by cluster. It differs from wrapper `Matchby` in package `Matching` in the way standard errors are calculated and because the caliper is in standard deviation units of the covariates on the overall dataset (so the caliper is the same for all clusters). Moreover, observation weights are available.

Value

`index.control` The index of control observations in the matched dataset.

`index.treated` The index of control observations in the matched dataset.

`index.dropped` The index of dropped observations due to the exact or caliper option. Note that these observations are treated if estimand is "ATT", controls if "ATC".

`est` The causal estimate. This is provided only if `Y` is not null. If estimand is "ATT" it is the (weighted) mean of `Y` in matched treated units minus the (weighted) mean of `Y` in matched controls. Equivalently, it is the weighted average of the within-cluster ATTs, with weights given by cluster sizes in the matched dataset.

`se` A model-based standard error for the causal estimand. This is a cluster robust estimator of the standard error for the linear model: `Y ~ constant+Tr`, run on the matched dataset (see `cluster.vcov` for details on how this estimator is obtained). Note that these standard errors differ from a weighted average of cluster specific standard errors provided by the `Matchby` function, which are generally larger. Estimating standard errors for causal parameters with clustered data is an active field of research and there is no perfect solution to date.

`mdata` A list containing the matched datasets produced by `CMatch`. Three datasets are included in this list: `Y`, `Tr` and `X`. The matched dataset for `Group` can be recovered by `rbind(Group[index.treated],Group[index.control])`.

`orig.treated.nobs.by.group`
The original number of treated observations by group in the dataset.

`orig.control.nobs.by.group`
: The original number of control observations by group in the dataset.

`orig.dropped.nobs.by.group`
: The number of dropped observations by group after within cluster matching.

`orig.nobs`
: The original number of observations in the dataset.

`orig.wnobs`
: The original number of weighted observations in the dataset.

`orig.treated.nobs`
: The original number of treated observations in the dataset.

`orig.control.nobs`
: The original number of control observations in the dataset.

`wnobs`
: the number of weighted observations in the matched dataset.

`caliper`
: The caliper used.

`intcaliper`
: The internal caliper used.

`exact`
: The value of the exact argument.

`ndrops.matches`
: The number of matches dropped either because of the caliper or exact option (or because of forcing the match within-clusters).

`estimand`
: The estimand required.

Note

The function returns an object of class `CMatch`. The `CMatchBalance` function can be used to examine the covariate balance before and after matching (see the examples below).

Author(s)

Massimo Cannas <massimo.cannas@unica.it>

References

Sekhon, Jasjeet S. 2011. Multivariate and Propensity Score Matching Software with Automated Balance Optimization. *Journal of Statistical Software 42(7): 1-52.* http://www.jstatsoft.org/v42/i07/

Arpino, B., and Cannas, M. (2016) Propensity score matching with clustered data. An application to the estimation of the impact of caesarean section on the Apgar score. *Statistics in Medicine, 35: 2074–2091*. doi: 10.1002/sim.6880.

See Also

See also `Match`, `MatchBalance`,`cluster.vcov`

Examples

```
data(schools)

# Kreft and De Leeuw, Introducing Multilevel Modeling, Sage (1988).
# The data set is the subsample of NELS-88 data consisting of 10 handpicked schools
# from the 1003 schools in the full data set.
```

```
# Let us consider the following variables:

X<-schools$ses  # (socio economic status)
Y<-schools$math #(mathematics score)
Tr<-ifelse(schools$homework > 1, 1 ,0)
Group<-schools$schid #(school ID)

# When Group is missing/NULL or there is only one group, CMatch returns
# the output of the Match function (with a warning).

# Multivariate Matching on covariates in X
# default parameters: one-to-one matching on X with replacement with a caliper of 0.25

### Matching within schools
 mw <- CMatch(type="within",Y=Y, Tr=Tr, X=X, Group=Group, caliper=0.1)

 # compare balance before and after matching
 bmw  <- CMatchBalance(Tr~X, data=schools, match.out = mw)

 # calculate proportion of matched observations
  (mw$orig.treated.nobs-mw$ndrops)/mw$orig.treated.nobs

 # check number of drops by school
 mw$orig.dropped.nobs.by.group

 # examine output
 mw           # complete output
 summary(mw)  # basic output statistics

### Match preferentially within school
# i.e. first match within schools
# then tries to match remaining units between schools
 mpw <- CMatch(type="pwithin",Y=schools$math, Tr=Tr, X=schools$ses,
 Group=schools$schid, caliper=0.1)

# examine covariate balance
  bmpw<- CMatchBalance(Tr~ses,data=schools,match.out = mpw)

# proportion of matched observations
  (mpw$orig.treated.nobs-mpw$ndrops) / mpw$orig.treated.nobs
# check drops by school
  mpw$orig.dropped.nobs.by.group.after.pref.within
# proportion of matched observations after match-within only
 (mpw$orig.treated.nobs-sum(mpw$orig.dropped.nobs.by.group.after.within)) / mpw$orig.treated.nobs

# see complete output
   mpw
# or use summary method for main results
   summary(mpw)

#### Propensity score matching

# estimate the ps model
```

```
mod <- glm(Tr~ses+parented+public+sex+race+urban,
family=binomial(link="logit"),data=schools)
eps <- fitted(mod)

# eg 1: within school propensity score matching
psmw <- CMatch(type="within",Y=schools$math, Tr=Tr, X=eps,
Group=schools$schid, caliper=0.1)

# eg 2: preferential within school propensity score matching
psmw <- CMatch(type="pwithin",Y=schools$math, Tr=Tr, X=eps, Group=schools$schid, caliper=0.1)

# eg 3: propensity score matching using ps estimated from a logit model with dummies for hospitals

mod <- glm(Tr ~ ses + parented + public + sex + race + urban
+schid - 1,family=binomial(link="logit"),data=schools)
eps <- fitted(mod)

dpsm <- CMatch(type="within",Y=schools$math, Tr=Tr, X=eps, Group=NULL, caliper=0.1)
# this is equivalent to run Match with X=eps

# eg4: propensity score matching using ps estimated from multilevel logit model
# (random intercept at the hospital level); see Arpino and Mealli

require(lme4)
mod <- glmer(Tr ~ ses + parented + public + sex + race + urban + (1 | schid),
family=binomial(link="logit"), data=schools)
eps <- fitted(mod)

mpsm <- CMatch(type="within",Y=schools$math, Tr=Tr, X=eps, Group=NULL, caliper=0.1)
# note: equivalent to run Match with X=eps
```

`CMatchBalance` *Analyze covariate balance before and after matching.*

Description

Generic function for analyzing covariate balance. If `match.out` is `NULL` only balance statistics for the unmatched data are returned otherwise both before and after matching balance are given. The function is a wrapper calling `MatchBalance`, possibly after coercing the class of `match.out`. See `MatchBalance` for more detailed description.

Usage

```
CMatchBalance(match.out, formula, data = NULL, ks = TRUE,
nboots = 500, weights = NULL, digits = 5, paired = TRUE, print.level = 1)
```

Arguments

match.out	A matched data set, i.e., the result of a call to Match or CMatch.
formula	This formula does not estimate a model. It is a compact way to describe which variables should be compared between the treated and control group. See MatchBalance.
data	An optional data set for the variables indicated in the formula argument.
ks	A flag for whether Kolmogorov-Smirnov tests should be calculated.
weights	A vector of observation-specific weights.
nboots	The number of bootstrap replication to be used.
digits	The number of digits to be displayed in the output
paired	A flag for whether a paired t.test should be used for the matched data. An unpaired t.test is always used for unmatched data.
print.level	The amount of printing, taking values 0 (no printing), 1(summary) and 2 (dtailed results). Default to 1.

Details

The function is a wrapper of the MatchBalance function. If match.out is of class Match (or NULL) then it calls MatchBalance. If match.out is of classCMatch then it coerces the class to Match before calling MatchBalance. This function is meant to exploit MatchBalance for CMatch objects for which MatchBalance would not work.

Value

Balance statistics for the covariates specified in the *right* side of formula argument. Statistics are compared between the two groups specified by the binary variable in the *left* side of formula.

Author(s)

Massimo Cannas <massimo.cannas@unica.it> and a special thanks to Thomas W. Yee for his help.

References

Sekhon, Jasjeet S. 2011. Multivariate and Propensity Score Matching Software with Automated Balance Optimization. *Journal of Statistical Software 42(7): 1-52.* http://www.jstatsoft.org/v42/i07/

See Also

MatchBalance

Examples

```
data(schools)

# Kreft and De Leeuw, Introducing Multilevel Modeling, Sage (1988).
# The data set is the subsample of NELS-88 data consisting of 10 handpicked schools
# from the 1003 schools in the full data set.
```

```
# Let us consider the following variables:

X<-schools$ses  # (socio economic status)
Y<-schools$math #(mathematics score)
Tr<-ifelse(schools$homework > 1, 1 ,0)
Group<-schools$schid #(school ID)

# Multivariate Matching on covariates X

### Matching within schools
 mw <- CMatch(type="within",Y=Y, Tr=Tr, X=X, Group=Group, caliper=0.1)

 # Balance statistics for X variables(s) before and after matching within schools.
 CMatchBalance(Tr~X,data=schools,match.out = mw)


### Match preferentially within school
# i.e. first match within schools
# then tries to match remaining units between schools

 mpw <- CMatch(type="pwithin",Y=schools$math, Tr=Tr, X=schools$ses,
 Group=schools$schid, caliper=0.1)

# examine covariate balance of variable(s) X before and after preferential matching within schools
  CMatchBalance(Tr~X, data=schools, match.out = mpw)
```

MatchPW *Preferential Within-cluster Matching*

Description

This function implements preferential within-cluster matching. In other words, units that do not match within clusters (as defined by the Group variable) can match between cluster in the second step.

Usage

```
MatchPW(Y = NULL, Tr, X, Group = NULL, estimand = "ATT", M = 1,
 exact = NULL, caliper = 0.25, replace = TRUE, ties = TRUE, weights = NULL, ...)
```

Arguments

Y	A vector containing the outcome of interest.
Tr	A vector indicating the treated and control units.
X	A matrix of covariates we wish to match on. This matrix should contain all confounders or the propensity score or a combination of both.

Group	A vector describing the clustering structure (typically the cluster ID). This can be any numeric vector of the same length of Tr and X containing integer numbers in ascending order otherwise an error message will be returned. Default is NULL, however if Group is missing, NULL or contains only one value the output of the *Match* function is returned with a warning.
estimand	The causal estimand desired, one of "ATE", "ATT" and "ATC", which stand for Average Treatment Effect, Average Treatment effect on the Treated and on the Controls, respectively. Default is "ATT".
M	The number of matches which are sought for each unit. Default is 1 ("one-to-one matching").
exact	An indicator for whether exact matching on the variables contained in X is desired. Default is FALSE. This option has precedence over the caliper option.
caliper	A maximum allowed distance for matching units. Units for which no match was found within caliper distance are discarded. Default is 0.25. The caliper is interpreted in standard deviation units of the *unclustered* data for each variable. For example, if caliper=0.25 all matches at distance bigger than 0.25 times the standard deviation for any of the variables in X are discarded. The caliper is used for both within and between clusters matching.
replace	Default is TRUE. From version 2.3 this parameter can be set to FALSE. Assuming ATT this means that controls matched within cannot be matched between (i.e. in the second step). However note that, even when replace is set to FALSE, controls can be re-used during match between.
ties	An indicator for dealing with multiple matches. If more than M matches are found for each unit the additional matches are a) wholly retained with equal weights if ties=TRUE; b) a random one is chosen if ties=FALSE. Default is TRUE.
weights	A vector of observation specific weights.
...	Please note that all additional arguments of the Match function are not used.

Details

The function performs preferential within-cluster matching in the clusters defined by the variable Group. In the first phase matching within clusters is performed (see MatchW) and in the second the unmatched treated (or controls if estimand="ATC") are matched with all controls (treated) units. This can be helpful to avoid dropping many units in small clusters.

Value

index.control	The index of control observations in the matched dataset.
index.treated	The index of control observations in the matched dataset.
index.dropped	The index of dropped observations due to the exact or caliper option. Note that these observations are treated if estimand is "ATT", controls if "ATC".
est	The causal estimate. This is provided only if Y is not null. If estimand is "ATT" it is the (weighted) mean of Y in matched treated minus the (weighted) mean of Y in matched controls. Equivalently it is the weighted average of the within-cluster ATTs, with weights given by cluster sizes in the matched dataset.

se A model-based standard error for the causal estimand. This is a cluster robust estimator of the standard error for the linear model: y ~ constant+Tr, run on the matched dataset (see cluster.vcov for details on how this estimator is obtained).

mdata A list containing the matched datasets produced by MatchPW. Three datasets are included in this list: Y, Tr and X. The matched dataset for Group can be recovered by rbind(Group[index.treated],Group[index.control]).

orig.treated.nobs.by.group
The original number of treated observations by group in the dataset.

orig.control.nobs.by.group
The original number of control observations by group in the dataset.

orig.dropped.nobs.by.group
The number of dropped observations by group after within cluster matching.

orig.dropped.nobs.by.group.after.pref.within
The number of dropped observations by group after preferential within group matching.

orig.nobs The original number of observations in the dataset.

orig.wnobs The original number of weighted observations in the dataset.

orig.treated.nobs
The original number of treated observations in the dataset.

orig.control.nobs
The original number of control observations in the dataset.

wnobs the number of weighted observations in the matched dataset.

caliper The caliper used.

intcaliper The internal caliper used.

exact The value of the exact argument.

ndrops.matches The number of matches dropped either because of the caliper or exact option.

estimand The estimand required.

Note

The function returns an object of class CMatch. The CMatchBalance function can be used to examine the covariate balance before and after matching. See the examples below.

Author(s)

Massimo Cannas <massimo.cannas@unica.it>

References

Sekhon, Jasjeet S. 2011. Multivariate and Propensity Score Matching Software with Automated Balance Optimization. *Journal of Statistical Software, 42(7): 1-52.* http://www.jstatsoft.org/v42/i07/

Arpino, B., and Cannas, M. (2016) Propensity score matching with clustered data. An application to the estimation of the impact of caesarean section on the Apgar score. *Statistics in Medicine, 35: 2074-2091* doi: 10.1002/sim.6880.

See Also

See also Match, MatchBalance,cluster.vcov

Examples

```
data(schools)

# Kreft and De Leeuw, Introducing Multilevel Modeling, Sage (1988).
# The data set is the subsample of NELS-88 data consisting of 10 handpicked schools
# from the 1003 schools in the full data set.

X<-schools$ses  # (socio economic status)
Y<-schools$math #(mathematics score)
Tr<-ifelse(schools$homework > 1, 1 ,0)
Group<-schools$schid #(school ID)
# Note that when Group is missing, NULL or there is only one Group,
# MatchPW returns the same output of the Match function (with a warning).


# Matching math scores between group of students. X are confounders.


### Match preferentially within-school
# first match students within schools
# then tries to match remaining students between schools
 mpw <- MatchPW(Y=schools$math, Tr=Tr, X=schools$ses, Group=schools$schid, caliper=0.1)


# examine covariate balance
  bmpw<- CMatchBalance(Tr~ses,data=schools,match.out=mpw)

# proportion of matched observations
  (mpw$orig.treated.nobs-mpw$ndrops) / mpw$orig.treated.nobs
# check drops by school
  mpw$orig.ndrops.by.group

# estimate the math score difference (default is ATT)
  mpw$estimand

# complete results
   mpw
# or use summary method for main results
   summary(mpw)


#### Propensity score matching

# estimate the propensity score (eps)

mod <- glm(Tr~ses+parented+public+sex+race+urban,
family=binomial(link="logit"),data=schools)
eps <- fitted(mod)
```

```
# eg 1: preferential within-school propensity score matching
MatchPW(Y=schools$math, Tr=Tr, X=eps, Group=schools$schid, caliper=0.1)

# eg 2: standard propensity score matching using eps
# from a logit model with dummies for schools

mod <- glm(Tr ~ ses + parented + public + sex + race + urban
+schid - 1,family=binomial(link="logit"),data=schools)
eps <- fitted(mod)

MatchPW(Y=schools$math, Tr=Tr, X=eps, caliper=0.1)

# eg3: standard propensity score matching using ps estimated from
# multilevel logit model (random intercept at the school level)

require(lme4)
mod<-glmer(Tr ~ ses + parented + public + sex + race + urban + (1|schid),
family=binomial(link="logit"), data=schools)
eps <- fitted(mod)

MatchPW(Y=schools$math, Tr=Tr, X=eps, Group=NULL, caliper=0.1)
```

`MatchW` *Within-cluster Matching*

Description

This function implements multivariate and propensity score matching within clusters defined by the `Group` variable.

Usage

```
MatchW(Y = NULL, Tr, X, Group = NULL, estimand = "ATT", M = 1,
exact = NULL, caliper = 0.25, weights = NULL, replace = TRUE, ties = TRUE, ...)
```

Arguments

`Y`	A vector containing the outcome of interest.
`Tr`	A vector indicating the treated and control units.
`X`	A matrix of covariates we wish to match on. This matrix should contain all confounders or the propensity score or a combination of both.
`Group`	A vector describing the clustering structure (typically the cluster ID). This can be any numeric vector of the same length of `Tr` and `X` containing integer numbers in ascending order otherwise an error message will be returned. Default is NULL, however if `Group` is missing, NULL or it contains only one value the output of the `Match` function is returned with a warning.

estimand	The causal estimand desired, one of "ATE", "ATT" and "ATC", which stand for Average Treatment Effect, Average Treatment effect on the Treated and on the Controls, respectively. Default is "ATT".
M	The number of matches which are sought for each unit. Default is 1 ("one-to-one matching").
exact	An indicator for whether exact matching on the variables contained in X is desired. Default is FALSE. This option has precedence over the caliper option.
caliper	A maximum allowed distance for matching units. Units for which no match was found within caliper distance are discarded. Default is 0.25. The caliper is interpreted in standard deviation units of the *unclustered* data for each variable. For example, if caliper=0.25 all matches at distance bigger than 0.25 times the standard deviation for any of the variables in X are discarded.
weights	A vector of specific observation weights.
replace	Matching can be with or without replacement depending on whether matches can be re-used or not. Default is TRUE.
ties	An indicator for dealing with multiple matches. If more than M matches are found for each unit the additional matches are a) wholly retained with equal weights if ties=TRUE; b) a random one is chosen if ties=FALSE. Default is TRUE.
...	Note that additional arguments of the Match function are not used.

Details

This function is meant to be a natural extension of the Match function to clustered data. It retains the main arguments of Match but it has additional output showing matching results cluster by cluster. It differs from wrapper Matchby in package Matching in the way standard errors are calculated and because the caliper is in standard deviation units of the covariates on the overall dataset (so the caliper is the same for all clusters). Moreover, observation weights are available.

Value

index.control	The index of control observations in the matched dataset.
index.treated	The index of control observations in the matched dataset.
index.dropped	The index of dropped observations due to the exact or caliper option. Note that these observations are treated if estimand is "ATT", controls if "ATC".
est	The causal estimate. This is provided only if Y is not null. If estimand is "ATT" it is the (weighted) mean of Y in matched treated units minus the (weighted) mean of Y in matched controls. Equivalently, it is the weighted average of the within-cluster ATTs, with weights given by cluster sizes in the matched dataset.
se	A model-based standard error for the causal estimand. This is a cluster robust estimator of the standard error for the linear model: Y ~ constant+Tr, run on the matched dataset (see cluster.vcov for details on how this estimator is obtained). Note that these standard errors differ from a weighted average of cluster specific standard errors provided by the Matchby function, which are generally larger. Estimating standard errors for causal parameters with clustered data is an active field of research and there is no perfect solution to date.

mdata	A list containing the matched datasets produced by MatchPW. Three datasets are included in this list: Y, Tr and X. The matched dataset for Group can be recovered by rbind(Group[index.treated],Group[index.control]).
orig.treated.nobs.by.group	The original number of treated observations by group in the dataset.
orig.control.nobs.by.group	The original number of control observations by group in the dataset.
orig.dropped.nobs.by.group	The number of dropped observations by group after within cluster matching.
orig.nobs	The original number of observations in the dataset.
orig.wnobs	The original number of weighted observations in the dataset.
orig.treated.nobs	The original number of treated observations in the dataset.
orig.control.nobs	The original number of control observations in the dataset.
wnobs	the number of weighted observations in the matched dataset.
caliper	The caliper used.
intcaliper	The internal caliper used.
exact	The value of the exact argument.
ndrops.matches	The number of matches dropped either because of the caliper or exact option (or because of forcing the match within-clusters).
estimand	The estimand required.

Note

The function returns an object of class CMatch. The CMatchBalance function can be used to examine the covariate balance before and after matching (see the examples below).

Author(s)

Massimo Cannas <massimo.cannas@unica.it>

References

Sekhon, Jasjeet S. 2011. Multivariate and Propensity Score Matching Software with Automated Balance Optimization. *Journal of Statistical Software 42(7): 1-52.* http://www.jstatsoft.org/v42/i07/

Arpino, B., and Cannas, M. (2016) Propensity score matching with clustered data. An application to the estimation of the impact of caesarean section on the Apgar score. *Statistics in Medicine, 35: 2074–2091*. doi: 10.1002/sim.6880.

See Also

See also See also Match, MatchBalance,cluster.vcov

Examples

```
data(schools)

# Kreft and De Leeuw, Introducing Multilevel Modeling, Sage (1988).
# The data set is the subsample of NELS-88 data consisting of 10 handpicked schools
# from the 1003 schools in the full data set.

# Let us consider the following variables:

X<-schools$ses
Y<-schools$math
Tr<-ifelse(schools$homework>1,1,0)
Group<-schools$schid

# Note that when Group is missing / NULL or there is only one group the function MatchW returns
# the output of the Match function with a warning.

# Matching math scores between gropus of students. X are covariate(s) we wish to match on.

### Matching within schools
 mw <- MatchW(Y=Y, Tr=Tr, X=X, Group=Group, caliper=0.1)

 # compare balance before and after matching
   CMatchBalance(Tr~X,data=schools,match.out=mw)

 # find proportion of matched observations
  (mw$orig.treated.nobs-mw$ndrops)/mw$orig.treated.nobs

 # check number of drops by school
 mw$orig.ndrops.by.group

 # estimate the math score difference (default is ATT)
  mw$estimand

 # examine output
 mw                   # complete results
 summary(mw)          # main results


#### Propensity score matching

# estimate the propensity score (ps) model

mod <- glm(Tr~ses+parented+public+sex+race+urban,
family=binomial(link="logit"),data=schools)
eps <- fitted(mod)

# eg 1: within-school propensity score matching
psmw <- MatchW(Y=schools$math, Tr=Tr, X=eps, Group=schools$schid, caliper=0.1)

# We can use other strategies for controlling unobserved cluster covariates
# by using different specifications of ps:
```

```
# eg 2: standard propensity score matching using ps estimated
# from a logit model with dummies for schools

mod <- glm(Tr ~ ses + parented + public + sex + race + urban
+schid - 1,family=binomial(link="logit"),data=schools)
eps <- fitted(mod)



dpsm <- MatchW(Y=schools$math, Tr=Tr, X=eps, caliper=0.1)
# this is equivalent to run Match with X=eps

# eg3: standard propensity score matching using ps estimated from
# multilevel logit model (random intercept at the school level)

require(lme4)
mod<-glmer(Tr ~ ses + parented + public + sex + race + urban + (1|schid),
family=binomial(link="logit"), data=schools)
eps <- fitted(mod)

mpsm<-MatchW(Y=schools$math, Tr=Tr, X=eps, Group=NULL, caliper=0.1)
# this is equivalent to run Match with X=eps
```

`schools` *Schools data set (NELS-88)*

Description

Data set used by Kreft and De Leeuw in their book *Introducing Multilevel Modeling, Sage (1988)* to analyse the relationship between math score and time spent by students to do math homework. The data set is a subsample of NELS-88 data consisting of 10 handpicked schools from the 1003 schools in the full data set. Students are nested within schools and information is available both at the school and student level.

Usage

```
data("schools")
```

Format

A data frame with 260 observations on the following 19 variables.

`schid` School ID: a numeric vector identyfing each school.

`stuid` The student ID.

`ses` Socioeconomic status.

`meanses` Mean ses for the school.

`homework` The number of hours spent weekly doing homeworks.

`white` A dummy for white race (=1) versus non-white (=0).

`parented` Parents highest education level.

`public` Public school: 1=public, 0=non public.

`ratio` Student-teacher ratio.

`percmin` Percent minority in school.

`math` Math score

`sex` Sex: 1=male, 2=female.

`race` Race of student, 1=asian, 2=Hispanic, 3=Black, 4=White, 5=Native American.

`sctype` Type of school: 1=public, 2=catholic, 3= Private other religion, 4=Private non-r.

`cstr` Classroom environment structure: ordinal from 1=not accurate to 5=very much accurate.

`scsize` School size: ordinal from 1=[1,199) to 7=[1200+).

`urban` Urbanicity: 1=Urban, 2=Suburban, 3=Rural.

`region` Geographic region of the school: NE=1,NC=2,South=3,West=4.

`schnum` Standardized school ID.

Details

The data set is used in the example section to illustrate the use of functions `MatchW` and `MatchPW`.

Source

Ita G G Kreft, Jan De Leeuw 1988. Introducing Multilevel Modeling, Sage National Education Longitudinal Study of 1988 (NELS:88): https://nces.ed.gov/surveys/nels88/

See Also

See also `MatchW`, `MatchPW`

Examples

```
data(schools)

# Kreft and De Leeuw, Introducing Multilevel Modeling, Sage (1988).
# The data set is the subsample of NELS-88 data consisting of 10 handpicked schools
# from the 1003 schools in the full data set.

# To study the effect of the homeworks on the outcome math score, conditional on
# confounder(s) X and unobserved school features, we can define the following variables:

X<-schools$ses
# or define a vector for more than one confounder
X<-as.matrix(schools[,c("ses","white","public")])
Y<-schools$math
Tr<-ifelse(schools$homework>1,1,0)
Group<-schools$schid
```

summary.CMatch *Summarizing output from MatchW and MatchPW functions*

Description

Summary method for MatchW and MatchPW

Usage

```
## S3 method for class 'CMatch'
summary(object, ..., full = FALSE, digits = 5)
```

Arguments

object	An object of class "CMatch".
...	Other options for the generic summary function.
full	A flag for whether the unadjusted estimates and naive standard errors should also be summarized.
digits	The number of significant digits that should be displayed.

Details

If Group contains only one value the output is the same of the summary method of package Matching. Otherwise the output shows also the distribution of treated, control and possibly drop units, *by group*.

Value

A list giving a summary of the output from a "CMatch" object. The list includes the size of the original and the matched dataset, the number of treated and control observations in each group and the estimate (if Y is not NULL).

Note

Naive standard errors are not available when there is more than one group so the full parameter is ineffective in that case.

Author(s)

Massimo Cannas <massimo.cannas@unica.it>

References

Sekhon, Jasjeet S. 2011. Multivariate and Propensity Score Matching Software with Automated Balance Optimization. *Journal of Statistical Software 42(7): 1-52.* http://www.jstatsoft.org/v42/i07/

Arpino, B., and Cannas, M. (2016) Propensity score matching with clustered data. An application to the estimation of the impact of caesarean section on the Apgar score. *Statistics in Medicine, 35: 2074–2091*. doi: 10.1002/sim.6880.

See Also

See also CMatch, CMatchBalance

Index

∗ **causal inference**
CMatching-package, 2
∗ **clustered data**
CMatch, 4
MatchPW, 10
MatchW, 14
∗ **cluster**
CMatching-package, 2
∗ **covariate balance**
CMatchBalance, 8
∗ **matching**
CMatch, 4
CMatchBalance, 8
MatchPW, 10
MatchW, 14
∗ **school dataset (NELS-88)**
schools, 18

cluster.vcov, *5, 6*, *12, 13*, *15, 16*
CMatch, 4, *21*
CMatchBalance, 8, *21*
CMatching *(*CMatching-package*)*, 2
CMatching-package, 2

Match, *3*, *6*, *13*, *16*
MatchBalance, *3*, *6*, *9*, *13*, *16*
MatchPW, 10, *19, 20*
MatchW, 14, *19, 20*

print.summary.CMatch *(*summary.CMatch*)*, 20

schools, 18
summary.CMatch, 20