Package ‘MatchThem’

August 23, 2021

Title  Matching and Weighting Multiply Imputed Datasets

Description  Provides the necessary tools for the pre-processing techniques of matching and weight-
ing multiply imputed datasets to control for effects of confounders and to reduce the de-
gree of dependence on certain modeling assumptions in studying the causal associations be-
tween an exposure and an outcome. This package includes functions to perform match-
ing within and across the multiply imputed datasets using several matching methods, to esti-
mate weights of units in the imputed datasets using several weighting methods, to calcu-
late the causal effect estimate in each matched or weighted dataset using parametric or non-
parametric statistical models, and to pool the obtained estimates from these models accord-
ing to Rubin’s rules (please see <https://journal.r-project.org/archive/2021/RJ-2021-073/> for details).

Version  1.0.1

Maintainer  Farhad Pishgar <Farhad.Pishgar@Gmail.com>

Depends  R (>= 3.6.0)

Imports  MatchIt (>= 4.0.0), mice, rlang, survey, WeightIt

Suggests  Amelia, cobalt (>= 4.0.0), R.rsp

URL  https://github.com/FarhadPishgar/MatchThem

BugReports  https://github.com/FarhadPishgar/MatchThem/issues

License  GPL (>= 2)

Encoding  UTF-8

LazyData  true

VignetteBuilder  R.rsp

RoxygenNote  7.1.1

NeedsCompilation  no

Author  Farhad Pishgar [aut, cre],
     Noah Greifer [aut],
     Clémence Leyrat [ctb],
     Elizabeth Stuart [ctb]

Repository  CRAN

Date/Publication  2021-08-23 07:11:42 UTC
R topics documented:

- cbind .................................................. 2
- complete ............................................... 3
- is.mimids ............................................. 5
- is.mimipo ............................................. 6
- is.mimira ............................................. 7
- is.wimids ............................................. 9
- matchthem .......................................... 10
- mimids ................................................ 12
- mimipo ............................................... 13
- mimira ................................................. 14
- osteoarthritis ...................................... 15
- pool ..................................................... 16
- trim ..................................................... 17
- weightthem ......................................... 19
- wimids ............................................... 21
- with .................................................... 22

Index 25

---

cbind  

Combine mimids and wimids Objects by Columns

Description

This function combines a mimids or wimids object columnwise with additional datasets or variables. Typically these would be variables not included in the original imputation and therefore absent in the mimids or wimids object. with() can then be used on the output to run models with the added variables.

Usage

```r
cbind(..., deparse.level = 1)
```

## S3 method for class 'mimids'
cbind(..., deparse.level = 1)

## S3 method for class 'wimids'
cbind(..., deparse.level = 1)

Arguments

- `...` Objects to combine columnwise. The first should be a mimids or wimids object. Additional data.frames, matrices, factors, or vectors can be supplied. These can be given as named arguments.
- `deparse.level` Ignored.
complete

**Value**

An object with the same class as the first input object with the additional variables added to the components.

**Author(s)**

Farhad Pishgar and Noah Greifer

**See Also**

`mice::cbind.mids`

`cbind`

**Examples**

```r
#Loading libraries
library(MatchThem)
library(survey)

#Loading the dataset
data(osteoarthritis)

#Multiply imputing the missing values
imputed.datasets <- mice::mice(osteoarthritis, m = 5)

#Weighting the multiply imputed datasets
weighted.datasets <- weightthem(OSP ~ AGE + SEX + BMI + RAC + SMK,
                               imputed.datasets,
                               approach = 'within')

#Adding additional variables
weighted.datasets <- cbind(weighted.datasets,
                           logAGE = log(osteoarthritis$AGE))

#Using the additional variables in an analysis
models <- with(weighted.datasets,
               svyglm(KOA ~ OSP + logAGE, family = quasibinomial))

#Pooling results obtained from analyzing the datasets
results <- pool(models)
summary(results)
```

**Description**

`complete()` function extracts data from an object of the mimids or wimids class.
Usage

```r
## S3 method for class 'mimids'
complete(data, action = 1, include = FALSE, mild = FALSE, all = TRUE, ...)

## S3 method for class 'wimids'
complete(data, action = 1, include = FALSE, mild = FALSE, all = TRUE, ...)
```

Arguments

- **data**
  - A mimids or wimids object.
- **action**
  - The imputed dataset number, intended to extract its data, or an action. The input must be a positive integer or a keyword. The keywords include "all" (produces a mild object of the imputed datasets), "long" (produces a dataset with imputed datasets stacked vertically), and "broad" (produces a dataset with imputed datasets stacked horizontally). The default is 1.
- **include**
  - Whether the original data with the missing values should be included. The input must be a logical value. The default is FALSE.
- **mild**
  - Whether the return value should be an object of mild class. Please note that setting mild = TRUE overrides action keywords "long", "broad", and "repeated". The default is FALSE.
- **all**
  - Whether to include observations with a zero estimated weight. The default is TRUE.
- **...**
  - Ignored.

Details

`complete()` works by running `mice::complete()` on the mids object stored within the mimids or wimids object and appending the outputs of the matching or weighting procedure. For mimids objects, the appended outputs include the matching weights, the propensity score (if included), pair membership (if included), and whether each unit was discarded. For wimids objects, the appended output is the estimated weights.

Value

This function returns the imputed dataset within the supplied mimids or wimids objects.

References


See Also

- `mimids`
- `wimids`
- `mice::complete`
Examples

```r
#Loading libraries
library(MatchThem)

#Loading the dataset
data(osteoarthritis)

#Multiply imputing the missing values
imputed.datasets <- mice::mice(osteoarthritis, m = 5)

#Matching the multiply imputed datasets
matched.datasets <- matchthem(OSP ~ AGE + SEX + BMI + RAC + SMK,
                                imputed.datasets,
                                approach = 'within',
                                method = 'nearest')

#Extracting the first imputed dataset
matched.dataset.1 <- complete(matched.datasets, n = 1)
```

---

### is.mimids

**Checks for the mimids Class**

**Description**

`is.mimids()` function checks whether class of objects is `mimids` or not.

**Usage**

```r
is.mimids(object)
```

**Arguments**

- **object**: This argument specifies the object that should be checked to see if is of the `mimids` class or not.

**Details**

The class of objects is checked to be of the `mimids`.

**Value**

This function returns a logical value indicating whether `object` is of the `mimids` class.

**Author(s)**

Farhad Pishgar
See Also

matchthem
mimids

Examples

#Loading libraries
library(MatchThem)

#Loading the dataset
data(osteoarthritis)

#Multiply imputing the missing values
imputed.datasets <- mice::mice(osteoarthritis, m = 5)

#Matching the multiply imputed datasets
matched.datasets <- matchthem(OSP ~ AGE + SEX + BMI + RAC + SMK,
imputed.datasets, 
    approach = 'within',
    method = 'nearest')

#Checking the 'matched.datasets' object
is.mimids(matched.datasets)
is(matched.datasets)

is.mimipo Checks for the mimipo Class

Description

is.mimipo() function checks whether class of objects is mimipo or not.

Usage

is.mimipo(object)

Arguments

object This argument specifies the object that should be checked to see if is of the
        mimipo class or not.

Details

The class of objects is checked to be of the mimipo.

Value

This function returns a logical value indicating whether object is of the mimipo class.
is.mimira

Author(s)
Farhad Pishgar

See Also
pool
mimipo

Examples

#Loading libraries
library(MatchThem)
library(survey)

#Loading the dataset
data(osteoarthritis)

#Multiply imputing the missing values
imputed.datasets <- mice::mice(osteoarthritis, m = 5)

#Estimating weights of observations in the multiply imputed datasets
weighted.datasets <- weightthem(OSP ~ AGE + SEX + BMI + RAC + SMK,
imputed.datasets,
   approach = 'within',
   method = 'ps',
   estimand = "ATT")

#Analyzing the weighted datasets
models <- with(data = weighted.datasets,
   exp = svyglm(KOA ~ OSP, family = binomial))

#Pooling results obtained from analysing the datasets
results <- pool(models)

#Checking the 'results' object
is.mimipo(results)
is(results)

is.mimira

Checks for the mimira Class

Description

is.mimira() function checks whether class of objects is mimira or not.

Usage

is.mimira(object)
Arguments

object    This argument specifies the object that should be checked to see if is of the mimira class or not.

Details

The class of objects is checked to be of the mimira.

Value

This function returns a logical value indicating whether object is of the mimira class.

Author(s)

Farhad Pishgar

See Also

with
mimira

Examples

#Loading libraries
library(MatchThem)
library(survey)

#Loading the dataset
data(osteoarthritis)

#Multiply imputing the missing values
imputed.datasets <- mice::mice(osteoarthritis, m = 5)

#Estimating weights of observations in the multiply imputed datasets
weighted.datasets <- weightthem(OSP ~ AGE + SEX + BMI + RAC + SMK, 
imputed.datasets, 
    approach = 'within',
    method = 'ps',
    estimand = "ATT")

#Analyzing the weighted datasets
models <- with(data = weighted.datasets, 
    exp = svyglm(KOA ~ OSP, family = binomial))

#Checking the 'models' object
is.mimira(models)
is(models)
is.wimids

Checks for the wimids Class

Description

is.wimids() function checks whether class of objects is wimids or not.

Usage

is.wimids(object)

Arguments

object

This argument specifies the object that should be checked to see if is of the wimids class or not.

Details

The class of objects is checked to be of the wimids.

Value

This function returns a logical value indicating whether object is of the wimids class.

Author(s)

Farhad Pishgar

See Also

weightthem
wimids

Examples

#Loading libraries
library(MatchThem)

#Loading the dataset
data(osteoarthritis)

#Multiply imputing the missing values
imputed.datasets <- mice::mice(osteoarthritis, m = 5)

#Estimating weights of observations in the multiply imputed datasets
weighted.datasets <- weightthem(OSP ~ AGE + SEX + BMI + RAC + SMK, 
imputed.datasets, 
  approach = 'within',
  method = 'ps',


estimand = "ATT")

# Checking the 'weighted.datasets' object
is.wimids(weighted.datasets)
is(weighted.datasets)

---

**matchthem**

*Matches Multiply Imputed Datasets*

**Description**

matchthem() performs matching in the supplied imputed datasets, given as mids or amelia objects, by running `MatchIt::matchit()` on each of the imputed datasets with the supplied arguments.

**Usage**

```r
matchthem(
  formula,
  datasets,
  approach = "within",
  method = "nearest",
  distance = "glm",
  link = "logit",
  distance.options = list(),
  discard = "none",
  reestimate = FALSE,
  ...
)
```

**Arguments**

- **formula**
  
  A formula of the form `z ~ x1 + x2`, where `z` is the exposure and `x1` and `x2` are the covariates to be balanced, which is passed directly to `MatchIt::matchit()` to specify the propensity score model or treatment and covariates to be used in matching. See `matchit()` for details.

- **datasets**
  
  This argument specifies the datasets containing the exposure indicator and the potential confounders called in the formula. This argument must be an object of the mids or amelia class, which is typically produced by a previous call to `mice()` from the mice package or to amelia() from the Amelia package (the Amelia package is designed to impute missing data in a single cross-sectional dataset or in a time-series dataset, currently, the MatchThem package only supports the former datasets).

- **approach**
  
  The approach used to combine information across imputed datasets. Currently, "within" (performing matching within each imputed dataset) and "across" (estimating propensity scores within each dataset, averaging them across datasets, and performing matching on the averaged propensity scores in each dataset) approaches are available. The default is "within", which has been shown to have superior performance in most cases.
method

This argument specifies a matching method. Currently, "nearest" (nearest neighbor matching), "exact" (exact matching), "full" (full matching), "genetic" (genetic matching), "subclass" (subclassification), "cem" (coarsened exact matching), and "optimal" (optimal matching) methods are available. Only methods that produce a propensity score ("nearest", "full", "genetic", "subclass", and "optimal") are compatible with the "across" approach. The default is "nearest" for nearest neighbor matching. See matchit() for details.

distance

The method used to estimate the distance measure (e.g., propensity scores) used in matching, if any. Only options that specify a method of estimating propensity scores (i.e., not "mahalanobis") are compatible with the "across" approach. The default is "glm" for propensity scores estimating using logistic regression. See matchit() and distance for details and allowable options.

link, distance.options, discard, reestimate

Arguments passed to matchit() to control estimation of the distance measure (e.g., propensity scores).

... Additional arguments passed to matchit().

Details

If an amelia object is supplied to datasets, it will first be transformed into a mids object for further use. matchthem() works by calling mice::complete() on the mids object to extract a complete dataset, and then calls MatchIt::matchit() on each one, storing the output of each matchit() call and the mids in the output. All arguments supplied to matchthem() except datasets and approach are passed directly to matchit(). With the "across" method, the estimated propensity scores are averaged across imputations and re-supplied to another set of calls to matchit().

Value

An object of the mimids (matched multiply imputed datasets) class, which includes the supplied mids object (or an amelia object transformed into a mids object if supplied) and the output of the calls to matchit() on each imputed dataset.

Author(s)

Farhad Pishgar and Noah Greifer

References


See Also

mimids
with pool weightthem MatchIt:::matchit

Examples

#1

#Loading libraries
library(MatchThem)

#Loading the dataset
data(osteoarthritis)

#Multiply imputing the missing values
imputed.datasets <- mice::mice(osteoarthritis, m = 5)

#Matching the multiply imputed datasets
matched.datasets <- matchthem(OSP ~ AGE + SEX + BMI + RAC + SMK,
                              imputed.datasets,
                              approach = 'within',
                              method = 'nearest')

#2

#Loading libraries
library(MatchThem)

#Loading the dataset
data(osteoarthritis)

#Multiply imputing the missing values
imputed.datasets <- Amelia::amelia(osteoarthritis, m = 5,
                                   noms = c("SEX", "RAC", "SMK", "OSP", "KOA"))

#Matching the multiply imputed datasets
matched.datasets <- matchthem(OSP ~ AGE + SEX + BMI + RAC + SMK,
                               imputed.datasets,
                               approach = 'across',
                               method = 'nearest')
mimids object contains data of matched multiply imputed datasets. mimids objects are generated by calls to matchthem().

Details
mimids objects have methods for print(), summary(), plot(), and cbind().

Note
The MatchThem package does not use the S4 class definitions and instead relies on the S3 list equivalents.

Author(s)
Farhad Pishgar

References

See Also
matchthem

mimipo

Multiply Imputed Pooled Outcome

mimipo object contains data of multiply imputed pooled outcome. mimipo objects are generated by calls to pool().

Details
mimipo objects has methods for the print() and summary() functions (please see mice package reference manual for details).

Note
The MatchThem package does not use the S4 class definitions and instead relies on the S3 list equivalents.

Author(s)
Farhad Pishgar
References


See Also

pool

---

**mimira**  
* Multiply Imputed Repeated Analyses

Description

*mimira* object contains data of multiply imputed repeated analyses. *mimira* objects are generated by calls to *with()*.

Details

*mimira* objects has methods for the *print()* and *summary()* functions (please see *mice* package reference manual for details).

Note

The *MatchThem* package does not use the S4 class definitions and instead relies on the S3 list equivalents.

Author(s)

Farhad Pishgar

References


See Also

with
Description

osteoarthritis includes demographic data of 2,585 units (individuals) with or at risk of knee osteoarthritis. The recorded data has missing values in body mass index (BMI, a quantitative variable), race (RAC, a categorical qualitative variable), smoking status (SMK, a binary qualitative variable), and knee osteoarthritis status at follow-up (KOA, a binary qualitative variable).

Usage

osteoarthritis

Format

This dataset contains 2,585 rows and 7 columns. Each row presents data of an unit (individual) and each column presents data of a characteristic of that individual. The columns are:

AGE  Age of each unit (individual);
SEX  Gender of each unit (individual), coded as 0 (female) and 1 (male);
BMI  Estimated body mass index of each unit (individual);
RAC  Race of each unit (individual), coded as 0 (other), 1 (Caucasian), 2 (African American), and 3 (Asian);
SMK  The smoking status of each unit (individual), coded as 0 (non-smoker) and 1 (smoker);
OSP  Osteoporosis status of each unit (individual) at baseline, coded as 0 (negative) and 1 (positive);
KOA  Knee osteoarthritis status of each unit (individual) in the follow-up, coded as 0 (at risk) and 1 (diagnosed).

Source

The information presented in the osteoarthritis dataset is based on the publicly available data of the Osteoarthritis Initiative (OAI) project (see https://nda.nih.gov/oai/ for details), with changes.
pool

Description

pool() pools estimates from the analyses done within each imputed dataset. The typical sequence of steps to do a matching procedure on the imputed datasets are:

1. Impute the missing values using the mice() function (from the mice package) or the amelia() function (from the Amelia package), resulting in a multiple imputed dataset (an object of the mids or amelia class);
2. Match or weight each imputed dataset using matchthem() or weightthem(), resulting in an object of the mimids or wimids class;
3. Check the extent of balance of covariates across the matched datasets (using functions in cobalt);
4. Fit the statistical model of interest on each matched dataset by the with() function, resulting in an object of the mimira class; and
5. Pool the estimates from each model into a single set of estimates and standard errors, resulting in an object of the mimipo class.

Usage

pool(object, dfcom = NULL)

Arguments

object An object of the mimira class (produced by a previous call to with()).
dfcom A positive number representing the degrees of freedom in the data analysis. The default is NULL, which means to extract this information from the fitted model with the lowest number of observations or the first fitted model (when that fails the parameter is set to 999999).

Details

pool() function averages the estimates of the model and computes the total variance over the repeated analyses by Rubin’s rules. It calls mice::pool() after computing the model degrees of freedom.

Value

This function returns an object of the mimipo class. Methods for mimipo objects (e.g., print(), summary(), etc.) are available in mice, which does not need to be attached to use them.

References

See Also

with()
mice::pool()

Examples

#Loading libraries
library(MatchThem)
library(survey)

#Loading the dataset
data(osteoarthritis)

#Multiply imputing the missing values
imputed.datasets <- mice::mice(osteoarthritis, m = 5)

#Weighting the multiply imputed datasets
weighted.datasets <- weightthem(OSP ~ AGE + SEX + BMI + RAC + SMK,
                                 imputed.datasets,
                                 approach = 'within',
                                 method = 'ps')

#Analyzing the weighted datasets
models <- with(weighted.datasets,
               svyglm(KOA ~ OSP, family = quasibinomial))

#Pooling results obtained from analyzing the datasets
results <- pool(models)
summary(results)

trim

Trim Weights

Description

Trims (i.e., truncates) large weights by setting all weights higher than that at a given quantile to
the weight at the quantile. This can be useful in controlling extreme weights, which can reduce
effective sample size by enlarging the variability of the weights.

Usage

## S3 method for class 'wimids'
trim(w, at = 0, lower = FALSE, ...)

Arguments

w A wimids object; the output of a call to weightthem().
at  Either the quantile of the weights above which weights are to be trimmed (given as a single number between 0.5 and 1) or the number of weights to be trimmed (e.g., at = 3 for the top 3 weights to be set to the 4th largest weight). The input must be a numeric value. The default is 0.

lower  Whether also to trim at the lower quantile (e.g., for at = 0.9, trimming at both the 0.1 and 0.9 quantiles, or for at = 3, trimming the top and bottom 3 weights). The input must be a logical value. The default is FALSE.

Details

trim.wimids() works by calling WeightIt::trim() on each weightit object stored in the models component of the wimids object. Because trim() itself is not exported from MatchThem, it must be called using WeightIt::trim() or by attaching WeightIt (i.e., running library(WeightIt)) before use.

Value

An object of class wimids, identical to the original object except with trim() applied to each of the weightit objects in the models component.

Author(s)

Noah Greifer

See Also

WeightIt::trim()

Examples

#Loading libraries
library(MatchThem)

#Loading the dataset
data(osteoarthritis)

#Multiply imputing the missing values
imputed.datasets <- mice::mice(osteoarthritis, m = 5)

#Estimating weights of observations in the multiply imputed datasets
weighted.datasets <- weightthem(OSP ~ AGE + SEX + BMI + RAC + SMK,
                                 imputed.datasets,
                                 approach = 'within',
                                 method = 'ps',
                                 estimand = "ATE")

#Trimming the top 10% of weights in each dataset
to the 90th percentile
trimmed.datasets <- trim(weighted.datasets, at = 0.9)
weightthem

Weights Multiply Imputed Datasets

Description

weightthem() performs weighting in the supplied imputed datasets, given as mids or amelia objects, by running WeightIt::weightit() on each of the imputed datasets with the supplied arguments.

Usage

weightthem(formula, datasets, approach = "within", method = "ps", ...)

Arguments

formula

A formula of the form \( z \sim x_1 + x_2 \), where \( z \) is the exposure and \( x_1 \) and \( x_2 \) are the covariates to be balanced, which is passed directly to WeightIt::weightit() to specify the propensity score model or treatment and covariates to be used to estimate the weights. See weightit() for details.

datasets

The datasets containing the exposure and covariates mentioned in the formula. This argument must be an object of the mids or amelia class, which is typically produced by a previous call to mice() from the mice package or to amelia() from the Amelia package (the Amelia package is designed to impute missing data in a single cross-sectional dataset or in a time-series dataset, currently, the MatchThem package only supports the former datasets).

approach

The approach used to combine information across imputed datasets. Currently, "within" (estimating weights within each imputed dataset) and "across" (estimating propensity scores within each dataset, averaging them across datasets, and computing a single set of weights to be applied to all datasets) approaches are available. The default is "within", which has been shown to have superior performance in most cases.

method

The method used to estimate weights. See weightit() for allowable options. Only methods that produce a propensity score ("ps", "gbm", "cbps", "super", and "bart") are compatible with the "across" approach. The default is "ps" propensity score weighting using logistic regression propensity scores.

...

Additional arguments to be passed to weightit(). see weightit() for more details.

Details

If an amelia object is supplied to datasets, it will first be transformed into a mids object for further use. weightthem() works by calling mice::complete() on the mids object to extract a complete dataset, and then calls WeightIt::weightit() on each one, storing the output of each weightit() call and the mids in the output. All arguments supplied to weightthem() except datasets and approach are passed directly to weightit(). With the across method, the estimated propensity scores are averaged across imputations and re-supplied to another set of calls to weightit().
Value

An object of the \texttt{wimids} (weighted multiply imputed datasets) class, which includes the supplied \texttt{mids} object (or an \texttt{amelia} object transformed into a \texttt{mids} object if supplied) and the output of the calls to \texttt{weightit()} on each imputed dataset.

Author(s)

Farhad Pishgar and Noah Greifer

References


See Also

\texttt{wimids}
\texttt{with}
\texttt{pool}
\texttt{matchthem}
\texttt{WeightIt::weightit}

Examples

#1

#Loading libraries
library(MatchThem)

#Loading the dataset
data(osteoarthritis)

#Multiply imputing the missing values
imputed.datasets <- mice::mice(osteoarthritis, m = 5)

#Estimating weights of observations in the multiply imputed datasets
weighted.datasets <- weightthem(OSP ~ AGE + SEX + BMI + RAC + SMK,
                                 imputed.datasets,
                                 approach = 'within',
                                 method = 'ps',
                                 estimand = "ATT")

#2

#Loading libraries
library(MatchThem)

#Loading the dataset
data(osteoarthritis)
```r
# Multiply imputing the missing values
imputed.datasets <- Amelia::amelia(osteoarthritis, m = 5,
   noms = c("SEX", "RAC", "SMK", "OSP", "KOA"))

# Estimating weights of observations in the multiply imputed datasets
weighted.datasets <- weightthem(OSP ~ AGE + SEX + BMI + RAC + SMK,
   imputed.datasets,
   approach = 'within',
   method = 'ps',
   estimand = "ATT")
```

---

### wimids

**Weighted Multiply Imputed Datasets**

---

#### Description

wimids object contains data of weighted multiply imputed datasets. The wimids object is generated by calls to the weightthem().

#### Details

wimids objects have methods for `print()`, `summary()`, and `cbind()`.

#### Note

The **MatchThem** package does not use the S4 class definitions and instead relies on the S3 list equivalents.

#### Author(s)

Farhad Pishgar

#### References


#### See Also

weightthem
with \( \text{Evaluates an Expression in Matched or Weighted Imputed Datasets} \)

Description

\( \text{with}() \) runs a model on the \( n \) imputed datasets of the supplied \text{mimids} or \text{wimids} object. The typical sequence of steps to do a matching procedure on the imputed datasets are:

1. Impute the missing values using the \text{mice()} function (from the \text{mice} package) or the \text{amelia()} function (from the \text{Amelia} package), resulting in a multiple imputed dataset (an object of the \text{mids} or \text{amelia} class);
2. Match or weight each imputed dataset using \text{matchthem()} or \text{weightthem()}, resulting in an object of the \text{mimids} or \text{wimids} class;
3. Check the extent of balance of covariates across the matched datasets (using functions in \text{cobalt});
4. Fit the statistical model of interest on each matched dataset by the \text{with()} function, resulting in an object of the \text{mimira} class; and
5. Pool the estimates from each model into a single set of estimates and standard errors, resulting in an object of the \text{mimipo} class.

Usage

\[
\begin{align*}
\text{## S3 method for class 'mimids'} \\
\text{with(data, expr, cluster, \ldots)}
\end{align*}
\]

\[
\begin{align*}
\text{## S3 method for class 'wimids'} \\
\text{with(data, expr, \ldots)}
\end{align*}
\]

Arguments

- \text{data} \quad A \text{mimids} or \text{wimids} object, typically produced by a previous call to the \text{matchthem()} or \text{weightthem()}.
- \text{expr} \quad An expression (usually a call to a modeling function like \text{glm()}, \text{coxph()}, \text{svyglm}(), etc.) to evaluate for each imputed dataset. See Details.
- \text{cluster} \quad When a function from \text{survey} (e.g., \text{svyglm}()) is supplied in \text{expr}, whether the standard errors should incorporate clustering due to dependence between matched pairs. This is done by supplying the variable containing pair membership to the \text{ids} argument of \text{link[survey:svydesign]{svydesign}}(). If unspecified, it will be set to \text{TRUE} if subclasses (i.e., pairs) are present in the output and there are 20 or more unique subclasses. It will be ignored for matching methods that don’t return subclasses (e.g., matching with replacement).
- \ldots \quad Additional arguments to be passed to \text{expr}. 
Details

with() applies the supplied model in expr to the matched or weighed imputed datasets, automatically incorporating the (matching) weights when possible. The argument to expr should be of the form glm(y ~ z, family = quasibinomial), for example, excluding the data or weights argument, which are automatically supplied.

Functions from the survey package, such as svyglm(), are treated a bit differently. No svydesign object needs to be supplied because with() automatically constructs and supplies it with the imputed dataset and estimated weights. When cluster = TRUE (or with() detects that pairs should be clustered; see the cluster argument above), pair membership is supplied to the ids argument of svydesign().

For generalized linear models, it is always recommended to use svyglm() rather than glm() in order to correctly compute standard errors. For Cox models, coxph() will produce correct standard errors when used with weighting but svycoxph() will produce more accurate standard errors when matching is used.

Value

An object of the mimira class containing the output of the analyses.

Author(s)

Farhad Pishgar and Noah Greifer

References


See Also

matchthem()
weightthem()
mice::with.mids()

Examples

#Loading libraries
library(MatchThem)
library(survey)

#Loading the dataset
data(osteoarthritis)

#Multiply imputing the missing values
imputed.datasets <- mice::mice(osteoarthritis, m = 5)

#Matching in the multiply imputed datasets
matched.datasets <- matchthem(OSP ~ AGE + SEX + BMI + RAC + SMK,
imputed.datasets,
approach = 'within',
method = 'nearest')

#Analyzing the matched datasets
models <- with(matched.datasets,
   svyglm(KOA ~ OSP, family = binomial),
   cluster = TRUE)
Index

* classes
  mimids, 12
  mimipo, 13
  mimira, 14
  wimids, 21

* datasets
  osteoarthritis, 15

cbind, 2, 3
cbind(), 13, 21
complete, 3
distance, 11
is.mimids, 5
is.mimipo, 6
is.mimira, 7
is.wimids, 9
matchit(), 10, 11
MatchIt::matchit, 12
MatchIt::matchit(), 10, 11
matchthem, 6, 10, 13, 20
matchthem(), 23
mice::cbind.mids, 3
mice::complete, 4
mice::complete(), 4, 11, 19
mice::pool(), 16, 17
mice::with.mids(), 23
mimids, 4, 6, 11, 12, 12
mimipo, 7, 13
mimira, 8, 14

osteoaartitis, 15
pool, 7, 12, 14, 16, 20
svyglm(), 22
trim, 17
weightit(), 19
WeightIt::trim(), 18
WeightIt::weightit, 20
WeightIt::weightit(), 19
weightthem, 9, 12, 19, 21
weightthem(), 17, 23
wimids, 4, 9, 20, 21
with, 8, 12, 14, 20, 22
with(), 17