

# Package: MatchThem (via r-universe)

August 24, 2024

**Title** Matching and Weighting Multiply Imputed Datasets

**Description** Provides essential tools for the pre-processing techniques of matching and weighting multiply imputed datasets. The package includes functions for matching within and across multiply imputed datasets using various methods, estimating weights for units in the imputed datasets using multiple weighting methods, calculating causal effect estimates in each matched or weighted dataset using parametric or non-parametric statistical models, and pooling the resulting estimates according to Rubin's rules (please see <https://journal.r-project.org/archive/2021/RJ-2021-073/> for more details).

**Version** 1.2.2

**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** 7.3.1

**NeedsCompilation** no

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

**Roxygen** list(markdown = TRUE)

**Repository** <https://farhadpishgar.r-universe.dev>

**RemoteUrl** <https://github.com/farhadpishgar/matchthem>

**RemoteRef** HEAD

**RemoteSha** f7e6110174d47a53f155647ec6e264b55e0bf4c1

## Contents

as.mimids	2
as.wimids	3
cbind	5
complete	6
is.mimids	8
is.mimipo	9
is.mimira	10
is.wimids	11
matchthem	12
mimids	15
mimipo	16
mimira	16
osteoarthritis	17
pool	18
trim	19
weightthem	21
wimids	23
with	24
<b>Index</b>	<b>27</b>

---

as.mimids	<i>Create a mimids object</i>
-----------	-------------------------------

---

## Description

Creates a mimids object from a list of matchit objects and an imputed dataset.

## Usage

```
as.mimids(x, ...)
```

## Default S3 method:  
as.mimids(x, datasets, ...)

## Arguments

x	A list of matchit objects, each the output of a call to <code>MatchIt::matchit()</code> on an imputed dataset.
...	Ignored.

`datasets` This argument specifies the datasets containing the exposure 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()` function from the **mice** package or to `amelia()` function 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).

### Details

The matched datasets are stored as though `matchthem()` was called with `approach = "within"`.

### Value

A `mimids` object.

### See Also

`matchthem()`, `mimids`, `MatchIt::matchit()`

### Examples

```
#Loading the dataset
data(osteoarthritis)

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

#Matching the multiply imputed datasets manually
match.list <- lapply(1:5, function(i) {
  MatchIt::matchit(OSP ~ AGE + SEX + BMI + RAC + SMK,
                   mice::complete(imputed.datasets, i),
                   method = 'nearest')
})

#Creating mimids object
matched.datasets <- as.mimids(match.list,
                              imputed.datasets)
```

---

as.wimids

*Create a wimids object*

---

### Description

Creates a `wimids` object from a list of `weightit` objects and an imputed dataset.

**Usage**

```
as.wimids(x, ...)

## Default S3 method:
as.wimids(x, datasets, ...)
```

**Arguments**

x	A list of <code>weightit</code> objects, each the output of a call to <code>WeightIt::weightit()</code> on an imputed dataset.
...	Ignored.
datasets	The datasets containing the exposure and covariates mentioned in the formula. This argument must be an object of the <code>mids</code> or <code>amelia</code> class, which is typically produced by a previous call to <code>mice()</code> from the <b>mice</b> package or to <code>amelia()</code> from the <b>Amelia</b> package (the <b>Amelia</b> package is designed to impute missing data in a single cross-sectional dataset or in a time-series dataset, currently, the <b>MatchThem</b> package only supports the former datasets).

**Details**

The weighted datasets are stored as though `weightthem()` was called with `approach = "within"`.

**Value**

A `wimids` object.

**See Also**

`weightthem()`, `wimids`, `WeightIt::weightit()`

**Examples**

```
#Loading the dataset
data(osteoarthritis)

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

#Matching the multiply imputed datasets manually
weight.list <- lapply(1:5, function(i) {
  WeightIt::weightit(OSP ~ AGE + SEX + BMI + RAC + SMK,
                     mice::complete(imputed.datasets, i),
                     method = 'glm',
                     estimand = 'ATT')
})

#Creating wimids object
weighted.datasets <- as.wimids(weight.list,
                               imputed.datasets)
```

---

`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 multiple 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

```
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 argument should be a `mimids` or `wimids` object. Additional `data.frames`, `matrixes`, `factors`, or `vectors` can be supplied. These can be given as named arguments.

`deparse.level` Ignored.

### Value

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

### Author(s)

Farhad Pishgar and Noah Greifer

### See Also

[cbind\(\)](#)

### Examples

```
#Loading libraries
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)

```

---

complete

*Extracts Multiply Imputed Datasets*

---

## Description

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

## Usage

```

## 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

<code>data</code>	A <code>mimids</code> or <code>wimids</code> object; the output of a call to <code>matchthem()</code> or <code>weightthem()</code> .
<code>action</code>	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 <code>mild</code> object of the multiply imputed datasets), "long" (produces a dataset with multiply imputed datasets stacked vertically), and "broad" (produces a dataset with multiply imputed datasets stacked horizontally). The default is 1.
<code>include</code>	Whether the original data with the missing values should be included. The input must be a logical value. The default is <code>FALSE</code> .
<code>mild</code>	Whether the return value should be an object of <code>mild</code> class. Please note that setting <code>mild = TRUE</code> overrides action keywords of "long", "broad", and "repeated". The default is <code>FALSE</code> .

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

Stef van Buuren and Karin Groothuis-Oudshoorn (2011). `mice`: Multivariate Imputation by Chained Equations in R. *Journal of Statistical Software*, 45(3): 1-67. doi:[10.18637/jss.v045.i03](https://doi.org/10.18637/jss.v045.i03)

## See Also

[mimids](#)

[wimids](#)

[mice::complete\(\)](#)

## Examples

```
#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**

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

**Arguments**

<code>object</code>	This argument specifies the object that should be checked to see if it 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](#)

[inherits\(\)](#)

**Examples**

```
#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.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 it 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.

## Author(s)

Farhad Pishgar

## See Also

[pool\(\)](#)  
[mimipo\(\)](#)

## Examples

```
#Loading libraries  
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 = 'glm',
        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.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 it 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(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 = 'glm',
                                estimand = "ATT")

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

#Checking the 'models' object
is.mimira(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 it 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 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 = 'glm',
                                estimand = "ATT")

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

---

`matchthem`*Matches Multiply Imputed Datasets*

---

**Description**

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

**Usage**

```
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 $z \sim x_1 + x_2$ , where $z$ is the exposure and $x_1$ and $x_2$ are the covariates to be balanced, which is passed directly to <code>MatchIt::matchit()</code> to specify the propensity score model or treatment and covariates to be used in matching. See <code>MatchIt::matchit()</code> for details.
datasets	This argument specifies the datasets containing the exposure and the potential confounders called in the formula. This argument must be an object of the <code>mids</code> or <code>amelia</code> class, which is typically produced by a previous call to <code>mice()</code> function from the <b>mice</b> package or to <code>amelia()</code> function from the <b>Amelia</b> package (the <b>Amelia</b> package is designed to impute missing data in a single cross-sectional dataset or in a time-series dataset, currently, the <b>MatchThem</b> package only supports the former datasets).
approach	The approach that should be used to combine information in multiply imputed datasets. Currently, "within" (performing matching within each dataset) and "across" (estimating propensity scores within each dataset, averaging them across datasets, and performing matching using 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" (optimal full matching), "genetic" (genetic matching), "subclass" (subclassification), "cem" (coarsened exact matching), "optimal" (optimal pair matching), "quick" (generalized full matching), and ("cardinality") (cardinality and profile matching) methods are available. Only methods that produce a propensity score ("nearest", "full", "genetic", "subclass", "optimal", and "quick") are compatible with the "across" approach. The default is "nearest" for nearest neighbor matching. See <code>MatchIt::matchit()</code> 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 estimating propensity scores using logistic regression. See <code>MatchIt::matchit()</code> and <code>MatchIt::distance</code> for details and allowable options.
link, distance.options, discard, reestimate	Arguments passed to <code>MatchIt::matchit()</code> to control estimation of the distance measure (e.g., propensity scores).
...	Additional arguments passed to <code>MatchIt::matchit()</code> .

**Details**

If an `amelia` object is supplied to `datasets`, it will 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" approach, the estimated propensity scores are averaged across multiply imputed datasets 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 multiply imputed dataset.

**Author(s)**

Farhad Pishgar and Noah Greifer

**References**

Daniel Ho, Kosuke Imai, Gary King, and Elizabeth Stuart (2007). Matching as Nonparametric Preprocessing for Reducing Model Dependence in Parametric Causal Inference. *Political Analysis*, 15(3): 199-236. <https://gking.harvard.edu/files/abs/matchp-abs.shtml>

Stef van Buuren and Karin Groothuis-Oudshoorn (2011). `mice`: Multivariate Imputation by Chained Equations in R. *Journal of Statistical Software*, 45(3): 1-67. doi:10.18637/jss.v045.i03

Gary King, James Honaker, Anne Joseph, and Kenneth Scheve (2001). Analyzing Incomplete Political Science Data: An Alternative Algorithm for Multiple Imputation. *American Political Science Review*, 95: 49-69. <https://gking.harvard.edu/files/abs/evil-abs.shtml>

**See Also**

[mimids](#)  
[with\(\)](#)  
[pool\(\)](#)  
[weightthem\(\)](#)  
[MatchIt::matchit\(\)](#)

**Examples**

```
#1

#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 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

*Matched Multiply Imputed Datasets*

---

## Description

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

Stef van Buuren and Karin Groothuis-Oudshoorn (2011). `mice`: Multivariate Imputation by Chained Equations in R. *Journal of Statistical Software*, 45(3): 1-67. doi:10.18637/jss.v045.i03

## See Also

`matchthem()`, `as.mimids()`

---

mimipo

*Multiply Imputed Pooled Outcome*

---

### Description

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

Stef van Buuren and Karin Groothuis-Oudshoorn (2011). *mice: Multivariate Imputation by Chained Equations in R*. *Journal of Statistical Software*, 45(3): 1-67. doi:10.18637/jss.v045.i03

### 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**

Stef van Buuren and Karin Groothuis-Oudshoorn (2011). *mice*: Multivariate Imputation by Chained Equations in R. *Journal of Statistical Software*, 45(3): 1-67. doi:[10.18637/jss.v045.i03](https://doi.org/10.18637/jss.v045.i03)

**See Also**

[with\(\)](#)

---

osteoarthritis	<i>Data of 2,585 Participants in the Osteoarthritis Initiative (OAI) Project</i>
----------------	--

---

**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 unit. 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); and

**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

*Pools Estimates by Rubin's Rules*

---

**Description**

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

1. Multiply impute the missing values using the `mice()` function (from the **mice** package) or the `amelia()` function (from the **Amelia** package), resulting in a multiply imputed dataset (an object of the `mids` or `amelia` class);
2. Match or weight each multiply imputed dataset using `matchthem()` or `weightthem()`, resulting in an object of the `mimids` or `wimids` class;
3. Check the extent of balance of covariates in the datasets (using functions from the **cobalt** package);
4. Fit the statistical model of interest on each 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**

<code>object</code>	An object of the <code>mimira</code> class (produced by a previous call to <code>with()</code> ).
<code>dfcom</code>	A positive number representing the degrees of freedom in the data analysis. The default is <code>NULL</code> , 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 from the `mimipo` class. Methods for `mimipo` objects (e.g., `print()`, `summary()`, etc.) are imported from the **mice** package.

## References

Stef van Buuren and Karin Groothuis-Oudshoorn (2011). *mice*: Multivariate Imputation by Chained Equations in R. *Journal of Statistical Software*, 45(3): 1-67. doi:10.18637/jss.v045.i03

## See Also

`with()`  
`mice::pool()`

## Examples

```
#Loading libraries
#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 = 'glm')

#Analyzing the weighted datasets
models <- with(weighted.datasets,
               WeightIt::glm_weightit(KOA ~ OSP,
                                       family = binomial))

#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(x, at = 0, lower = FALSE, ...)
```

**Arguments**

x	A wimids object; the output of a call to <code>weightthem()</code> .
at	numeric; either the quantile of the weights above which weights are to be trimmed. A single number between .5 and 1, or the number of weights to be trimmed (e.g., <code>at = 3</code> for the top 3 weights to be set to the 4th largest weight).
lower	logical; whether also to trim at the lower quantile (e.g., for <code>at = .9</code> , trimming at both .1 and .9, or for <code>at = 3</code> , trimming the top and bottom 3 weights). Default is FALSE to only trim the higher weights.
...	Ignored.

**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 from the `wimids` class, 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 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 = 'glm',
                                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 multiply imputed datasets, given as mids or amelia objects, by running `WeightIt::weightit()` on each of the multiply imputed datasets with the supplied arguments.

## Usage

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

## 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 <code>WeightIt::weightit()</code> to specify the propensity score model or treatment and covariates to be used to estimate the weights. See <code>WeightIt::weightit()</code> 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 <code>mice()</code> from the <b>mice</b> package or to <code>amelia()</code> from the <b>Amelia</b> package (the <b>Amelia</b> package is designed to impute missing data in a single cross-sectional dataset or in a time-series dataset, currently, the <b>MatchThem</b> package only supports the former datasets).
approach	The approach used to combine information in multiply imputed datasets. Currently, "within" (estimating weights within each dataset), "across" (estimating propensity scores within each dataset, averaging them across datasets, and computing a single set of weights based on that to be applied to all datasets), and "apw" (or averaging the probability weights, estimating weights within each dataset and averaging them across 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 <code>WeightIt::weightit()</code> for allowable options. Only methods that produce a propensity score ("glm", "gbm", "ipt", "cbps", "super", and "bart") are compatible with the "across" approach). The default is "glm" propensity score weighting using logistic regression propensity scores.
...	Additional arguments to be passed to <code>weightit()</code> . see <code>WeightIt::weightit()</code> for more details.

## Details

If an amelia object is supplied to datasets, it will 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 dataset, 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" approach, the estimated propensity scores are averaged across imputations and re-supplied to another set of calls to weightit().

### Value

An object of the `wimids()` (weighted 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 `weightit()` on each multiply imputed dataset.

### Author(s)

Farhad Pishgar and Noah Greifer

### References

Stef van Buuren and Karin Groothuis-Oudshoorn (2011). `mice`: Multivariate Imputation by Chained Equations in R. *Journal of Statistical Software*, 45(3): 1-67. doi:[10.18637/jss.v045.i03](https://doi.org/10.18637/jss.v045.i03)

### See Also

[wimids](#)  
[with\(\)](#)  
[pool\(\)](#)  
[matchthem\(\)](#)  
[WeightIt::weightit\(\)](#)

### Examples

```
#1

#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 = 'glm',
                                estimand = 'ATT')

#2

#Loading the dataset
data(osteoarthritis)
```

```
#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 = 'glm',
                                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

Stef van Buuren and Karin Groothuis-Oudshoorn (2011). `mice`: Multivariate Imputation by Chained Equations in R. *Journal of Statistical Software*, 45(3): 1-67. doi:[10.18637/jss.v045.i03](https://doi.org/10.18637/jss.v045.i03)

## See Also

[weightthem\(\)](#), [as.wimids\(\)](#)

---

with	<i>Evaluates an Expression in Matched or Weighted Multiply Imputed Datasets</i>
------	---

---

### Description

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

1. Multiply impute the missing values using the `mice()` function (from the **mice** package) or the `amelia()` function (from the **Amelia** package), resulting in a multiply imputed dataset (an object of the `mids` or `amelia` class);
2. Match or weight each multiply imputed dataset using `matchthem()` or `weightthem()`, resulting in an object of the `mimids` or `wimids` class;
3. Check the extent of balance of covariates in the datasets (using functions from the **cobalt** package);
4. Fit the statistical model of interest on each 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

```
## S3 method for class 'mimids'
with(data, expr, cluster, ...)
```

```
## S3 method for class 'wimids'
with(data, expr, ...)
```

### Arguments

<code>data</code>	A <code>mimids</code> or <code>wimids</code> object, typically produced by a previous call to the <code>matchthem()</code> or <code>weightthem()</code> .
<code>expr</code>	An expression (usually a call to a modeling function like <code>glm()</code> , <code>coxph()</code> , <code>svyglm()</code> , etc.) to evaluate in each (matched or weighted) multiply imputed dataset. See Details.
<code>cluster</code>	When a function from <b>survey</b> (e.g., <code>survey::svyglm()</code> ) is supplied in <code>expr</code> , 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 <code>ids</code> argument of <code>link[survey:svydesign]{svydesign()}</code> . If unspecified, it will be set to <code>TRUE</code> 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).
<code>...</code>	Additional arguments to be passed to <code>expr</code> .



## Details

`with()` applies the supplied model in `expr` to the (matched or weighted) multiply 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()`.

After weighting using `weightthem()`, `glm_weightit()` should be used as the modeling function to fit generalized linear models. It correctly produces robust standard errors that account for estimation of the weights, if possible. See `WeightIt::glm_weightit()` for details. Otherwise, `svyglm()` should be used rather than `glm()` in order to correctly compute standard errors. For Cox models, `coxph()` will produce approximately correct standard errors when used with weighting but `svycoxph()` will produce more accurate standard errors when matching is used.

## Value

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

## Author(s)

Farhad Pishgar and Noah Greifer

## References

Stef van Buuren and Karin Groothuis-Oudshoorn (2011). `mice`: Multivariate Imputation by Chained Equations in R. *Journal of Statistical Software*, 45(3): 1-67. doi:10.18637/jss.v045.i03

## See Also

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

## Examples

```
#Loading libraries
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,
```



# Index

## \* classes

mimids, 15  
mimipo, 16  
mimira, 16  
wimids, 23

## \* datasets

osteoarthritis, 17

as.mimids, 2  
as.mimids(), 15  
as.wimids, 3  
as.wimids(), 23

cbind, 5  
cbind(), 5, 15, 23  
complete, 6

inherits(), 8  
is.mimids, 8  
is.mimipo, 9  
is.mimira, 10  
is.wimids, 11

MatchIt::distance, 13  
MatchIt::matchit(), 2, 3, 12–14  
matchthem, 12  
matchthem(), 3, 6, 8, 15, 22, 24, 25  
mice::complete(), 7, 13, 21  
mice::pool(), 18, 19  
mice::with.mids(), 25  
mimids, 3, 5–8, 14, 15  
mimids(), 14  
mimipo, 16  
mimipo(), 9  
mimira, 16, 18  
mimira(), 10

osteoarthritis, 17

pool, 18  
pool(), 9, 14, 16, 22

survey::svyglm(), 24

trim, 19

WeightIt::glm\_weightit(), 25  
WeightIt::trim(), 20  
WeightIt::weightit(), 4, 21, 22  
weightthem, 21  
weightthem(), 4, 6, 12, 14, 20, 23–25  
wimids, 4–7, 22, 23  
wimids(), 12, 22  
with, 24  
with(), 10, 14, 17, 19, 22