

# Package ‘MatchThem’

March 23, 2020

**Title** Matching and Weighting Multiply Imputed Datasets

**Description** Provides the necessary tools for the pre-processing techniques of matching and weighting multiply imputed datasets to control for effects of confounders and to reduce the degree of dependence on certain modeling assumptions in studying the causal associations between an exposure and an outcome. This package includes functions to perform matching within and across the multiply imputed datasets using several matching methods, to estimate weights of units in the imputed datasets using several weighting methods, to calculate 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 according to Rubin's rules (please see <<https://github.com/FarhadPishgar/MatchThem>> for details).

**Version** 0.9.3

**Maintainer** Farhad Pishgar <[Farhad.Pishgar@gmail.com](mailto:Farhad.Pishgar@gmail.com)>

**Depends** MatchIt, WeightIt, R (>= 3.5.0)

**Imports** broom, dplyr, graphics, methods, mice, stats, survey

**Suggests** Amelia, cobalt, 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** 7.0.2

**NeedsCompilation** no

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

**Repository** CRAN

**Date/Publication** 2020-03-23 16:10:02 UTC

## R topics documented:

complete	2
is.mimids	4
is.mimipo	5
is.mimira	6
is.wimids	7
matchthem	8
mimids	11
mimipo	12
mimira	12
osteoarthritis	13
pool	14
weightthem	15
wimids	18
with	19

<b>Index</b>	<b>21</b>
--------------	-----------

---

complete	<i>Extracts Imputed Datasets</i>
----------	----------------------------------

---

### Description

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

### Usage

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

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

### Arguments

data	This argument specifies an object of the mimids or wimids class.
n	This argument specifies 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	This argument specifies whether the original data with the missing values should be included. The input must be a logical value. The default is FALSE.
mild	This argument specifies whether the return value should be an object of mild class. Please note that setting mild = TRUE overrides n keywords "long", "broad", and "repeated". The default is FALSE.

all	This argument specifies whether to include observations with a zero estimated weight. The default is TRUE.
...	Additional arguments to be passed to the function.

### Details

The datasets within the `mimids` or `wimids` class objects are extracted.

### Value

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

### Author(s)

Extracted from the **mice** package written by Stef van Buuren et al. with changes

### 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. <https://www.jstatsoft.org/v45/i03/>

### See Also

[mimids](#)

[wimids](#)

### Examples

```
#Loading libraries
library(mice)
library(MatchThem)

#Loading the dataset
data(osteoarthritis)

#Multiply imputing the missing values
imputed.datasets <- mice(osteoarthritis, m = 5, maxit = 10,
  method = c("", "", "mean", "polyreg",
    "logreg", "logreg", "logreg"))

#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**

`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(mice)
library(MatchThem)

#Loading the dataset
data(dataset)

#Multiply imputing the missing values
imputed.datasets <- mice(osteoarthritis, m = 5, maxit = 10,
                        method = c("", "", "mean", "polyreg",
                                   "logreg", "logreg", "logreg"))

#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.

## Author(s)

Farhad Pishgar

## See Also

[pool](#)

[mimipo](#)

---

`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(mice)
library(MatchThem)
library(survey)

#Loading the dataset
data(osteoarthritis)

#Multiply imputing the missing values
imputed.datasets <- mice(osteoarthritis, m = 5, maxit = 10,
                        method = c("", "", "mean", "polyreg",
                                   "logreg", "logreg", "logreg"))

#Estimating weights of observations in 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(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(mice)
library(MatchThem)

#Loading the dataset
data(osteoarthritis)

#Multiply imputing the missing values
imputed.datasets <- mice(osteoarthritis, m = 5, maxit = 10,
                        method = c("", "", "mean", "polyreg",
                                   "logreg", "logreg", "logreg"))

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

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

```

---

matchthem

*Matches Multiply Imputed Datasets*


---

**Description**

matchthem() function enables parametric models for causal inference to work better by selecting matched subsets of the control and treated subgroups of imputed datasets of a mids or amelia class object.

**Usage**

```

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

```

**Arguments**

**formula** This argument takes the usual syntax of R formula,  $z \sim x_1 + x_2$ , where  $z$  is a binary exposure indicator and  $x_1$  and  $x_2$  are the potential confounders. Both



the exposure indicator and the potential confounders must be contained in the imputed datasets, which is specified as `datasets` (see below). All of the usual R syntax for formula works. For example, `x1:x2` represents the first order interaction term between `x1` and `x2` and `I(x1^2)` represents the square term of `x1`. See `help(formula)` for details.

<code>datasets</code>	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 <code>mids</code> or <code>amelia</code> class, which is typically produced by a previous call to <code>mice()</code> or <code>mice.mids()</code> functions 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).
<code>approach</code>	This argument specifies a matching approach. Currently, "within" (calculating distance measures within each imputed dataset and matching observations based on them) and "across" (calculating distance measures within each imputed dataset, averaging distance measure for each observation across imputed datasets, and matching based on the averaged measures) approaches are available. The default is "within" which has been shown to produce unbiased results.
<code>method</code>	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 the "nearest", "full", "subclass", and "optimal" matching methods are compatible with the "across" approach). The default is "nearest". Note that within each of these matching methods, <b>MatchThem</b> offers a variety of options.
<code>distance</code>	This argument specifies the method that should be used to estimate the distance measure (the "mahalanobis" method for distance measure is not compatible with the "across" approach). The default is logistic regression, "logit". A variety of other methods are available (please see the <b>MatchIt</b> package reference manual < <a href="https://cran.r-project.org/package=MatchIt">https://cran.r-project.org/package=MatchIt</a> > for more details).
<code>distance.options</code>	This optional argument specifies the arguments that are passed to the model for estimating the distance measure. The input to this argument should be a list.
<code>discard</code>	This argument specifies whether to discard observations that fall outside some measure of support of the distance score before matching and not allow them to be used at all in the matching procedure. Note that discarding observations may change the quantity of interest being estimated. The current options are "none" (discarding no observations before matching), "both" (discarding all observations, both the control and treated observations, that are outside the support of the distance measure), "control" (discarding only control observations outside the support of the distance measure of the treated observations), and "treat" (discarding only treated observations outside the support of the distance measure of the control observations). The default is "none".
<code>reestimate</code>	This argument specifies whether the model for estimating the distance measure should be reestimated after observations are discarded. The input must be a logical value. The default is FALSE.

... Additional arguments to be passed to the matching method (please see the **MatchIt** package reference manual <<https://cran.r-project.org/package=MatchIt>> for more details).

### Details

The matching is done using the `matchthem(z ~ x1, ...)` command, where `z` is the exposure indicator and `x1` represents the potential confounder to be used in the matching model. There are a number of matching options. The default syntax is `matchthem(formula, datasets, approach = "within", method = "nearest", distance = "logit", ...)`. Summaries of the results can be seen graphically using `plot()` or numerically using `summary()` functions. The `print()` function also prints out the output.

### Value

This function returns an object of the `mimids` (matched multiply imputed datasets) class, that includes matched subsets of the imputed datasets primarily passed to the function by the `datasets` argument.

### 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. <http://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. <https://www.jstatsoft.org/v45/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. <http://j.mp/2o0rtGs>

### See Also

[mimids](#)

[with](#)

[pool](#)

### Examples

```
#Loading libraries
library(mice)
library(MatchThem)

#Loading the dataset
data(osteoarthritis)
```

```
#Multiply imputing the missing values
imputed.datasets <- mice(osteoarthritis, m = 5, maxit = 10,
                        method = c("", "", "mean", "polyreg",
                                   "logreg", "logreg", "logreg"))

#Matching the multiply imputed datasets
matched.datasets <- matchthem(OSP ~ AGE + SEX + BMI + RAC + SMK, imputed.datasets,
                              approach = 'within', 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 has methods for the `print()`, `summary()`, `plot()`, and `merge()` functions (please see **MatchIt** 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. <https://www.jstatsoft.org/v45/i03/>

## See Also

[matchthem](#)

---

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. <https://www.jstatsoft.org/v45/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. <https://www.jstatsoft.org/v45/i03/>

**See Also**

[with](#)

---

osteoarthritis

*Data of 2,585 Participants in the OAI Project*

---

**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), osteoporosis status at baseline (OSP, 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 characteristics 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, 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()` function pools estimates from  $n$  repeated data analyses. The typical sequence of steps to do a matching procedure on the imputed datasets are:

1. Impute the missing values by 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 each imputed dataset using a matching model by the `matchthem()` function, resulting in an object of the `mimids` class;
3. Check the extent of balance of covariates across the matched datasets;
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**

<code>object</code>	This argument specifies an object of the <code>mimira</code> class (produced by a previous call to <code>with()</code> function).
<code>dfcom</code>	This argument specifies 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 warning The function cannot extract the <code>dfcom</code> from the datasets, hence, large sample is assumed. is printed and 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.

**Value**

This function returns an object of the `mimipo` class.

**Author(s)**

Extracted from the **mice** package written by Stef van Buuren et al. with changes

**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. <https://www.jstatsoft.org/v45/i03/>

**See Also**

[with](#)

**Examples**

```
#Loading libraries
library(mice)
library(MatchThem)

#Loading the dataset
data(osteoarthritis)

#Multiply imputing the missing values
imputed.datasets <- mice(osteoarthritis, m = 5, maxit = 10,
                        method = c("", "", "mean", "polyreg",
                                   "logreg", "logreg", "logreg"))

#Matching 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(data = matched.datasets,
              exp = glm(KOA ~ OSP, family = binomial))

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

---

weightthem

*Weights Multiply Imputed Datasets*

---

**Description**

`weightthem()` function enables parametric models for causal inference to work better by estimating weights of the control and treated units in each imputed dataset of a `mids` or `amelia` class object.

**Usage**

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

**Arguments**

- |          |   |
|----------|---|
| formula  | This argument takes the usual syntax of R formula, $z \sim x1 + x2$ , where $z$ is a binary exposure indicator and $x1$ and $x2$ are the potential confounders. Both the exposure indicator and the potential confounders must be contained in the imputed datasets, which is specified as <code>datasets</code> (see below). All of the usual R syntax for formula works. For example, $x1 : x2$ represents the first order interaction term between $x1$ and $x2$ and $I(x1^2)$ represents the square term of $x1$ . See <code>help(formula)</code> 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 <code>mids</code> or <code>amelia</code> class, which is typically produced by a previous call to <code>mice()</code> or <code>mice.mids()</code> functions 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 | This argument specifies a matching approach. Currently, "within" (calculating distance measures within each imputed dataset and weighting observations based on them) and "across" (calculating distance measures within each imputed dataset, averaging distance measure for each observation across imputed datasets, and weighting based on the averaged measures) approaches are available. The default is "within" which has been shown to produce unbiased results.   |
| method   | This argument specifies the method that should be used to estimate weights. Currently, "ps" (propensity score weighting using generalized linear models), "gbm" (propensity score weighting using generalized boosted modeling), "cbps" (covariate balancing propensity score weighting), "npcbps" (non-parametric covariate balancing propensity score weighting), "eбал" (entropy balancing), "ebcw" (empirical balancing calibration weighting), "optweight" (optimization-based weighting), "super" (propensity score weighting using SuperLearner), and "user-defined" (weighting using a user-defined weighting function) are available (only the "ps", "gbm", "cbps", and "super" weighting methods are compatible with the "across" approach). The default is "ps". Note that within each of these weighting methods, <b>MatchThem</b> offers a variety of options. |
| estimand | This argument specifies the desired estimand. For binary and multinomial treatments, can be "ATE", "ATT", "ATC", and, for some weighting methods, "ATO" or  |





```
#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 has methods for the `print()`, `summary()`, and `merge()` functions (please see **WeightIt** 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. <https://www.jstatsoft.org/v45/i03/>

## See Also

[weightthem](#)

---

with

*Evaluates an Expression in Matched or Weighted Imputed Datasets*


---

## Description

`with()` function performs a statistical computation on the `n` imputed datasets of the `mimids` or `wimids` objects. The typical sequence of steps to do a matching or weighting procedure on the imputed datasets are:

1. Impute the missing values by 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 imputed datasets using a matching or weighting model by the `matchthem()` or `weightthem()` function, resulting in an object of the `mimids` or `wimids` class;
3. Check the extent of balance of covariates across the datasets;
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, ...)

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

## Arguments

<code>data</code>	This argument specifies an object of the <code>mimids</code> or <code>wimids</code> class, typically produced by a previous call to the <code>matchthem()</code> or <code>weightthem()</code> .
<code>expr</code>	This argument specifies an expression of the usual syntax of R formula (it also accepts expressions from <b>survey</b> package, like <code>svyglm()</code> , please note that you shouldn't include the <code>weights = weights</code> argument, see the package vignette for details).
<code>...</code>	Additional arguments to be passed to <code>expr</code> .

## Details

`with()` performs a computation on the imputed datasets.

## Value

This function returns an object of the `mimira` class.

**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. <https://www.jstatsoft.org/v45/i03/>

**See Also**

[matchthem](#)

[weightthem](#)

**Examples**

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

#Loading the dataset
data(osteoarthritis)

#Multiply imputing the missing values
imputed.datasets <- mice(osteoarthritis, m = 5, maxit = 10,
  method = c("", "", "mean", "polyreg",
    "logreg", "logreg", "logreg"))

#Estimating weights of observations in 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(data = weighted.datasets,
  exp = svyglm(KOA ~ OSP, family = binomial))
```

# Index

## \*Topic **classes**

mimids, [11](#)

mimipo, [12](#)

mimira, [12](#)

wimids, [18](#)

## \*Topic **datasets**

osteoarthritis, [13](#)

complete, [2](#)

is.mimids, [4](#)

is.mimipo, [5](#)

is.mimira, [6](#)

is.wimids, [7](#)

matchthem, [4](#), [8](#), [11](#), [20](#)

mimids, [3](#), [4](#), [10](#), [11](#)

mimipo, [5](#), [12](#)

mimira, [6](#), [12](#)

osteoarthritis, [13](#)

pool, [5](#), [10](#), [12](#), [14](#), [17](#)

weightthem, [7](#), [15](#), [18](#), [20](#)

wimids, [3](#), [7](#), [17](#), [18](#)

with, [6](#), [10](#), [13](#), [15](#), [17](#), [19](#)