Package ‘micemd’
November 17, 2023

Type    Package
Title   Multiple Imputation by Chained Equations with Multilevel Data
Version 1.10.0
Date    2023-11-17

Description
Addons for the 'mice' package to perform multiple imputation using chained equations with two-level data. Includes imputation methods dedicated to sporadically and systematically missing values. Imputation of continuous, binary or count variables are available. Following the recommendations of Audigier, V. et al (2018) <doi:10.1214/18-STS646>, the choice of the imputation method for each variable can be facilitated by a default choice tuned according to the structure of the incomplete dataset. Allows parallel calculation and overimputation for 'mice'.

License GPL-2 | GPL-3

Depends   R (>= 3.5.0), mice (>= 2.42)
Imports   Matrix, graphics, utils, stats, MASS, parallel, nlme, lme4,
          mvmeta (>= 0.4.7), jomo (>= 2.6-3), mvtnorm, digest, abind,
          GJRM (>= 0.2-6.4), mgcv, mixmeta, phivnorm
Suggests VIM, ggplot2, data.table, broom.mixed
RoxygenNote 6.1.0

NeedsCompilation no

Author   Vincent Audigier [aut, cre] (CNAM MSDMA team),
          Matthieu Resche-Rigon [aut] (INSERM ECSTRA team),
          Johanna Munoz Avila [ctb] (Julius Center Methods Group UMC, 2022)

Maintainer  Vincent Audigier <vincent.audigier@cnam.fr>
Repository CRAN
Date/Publication 2023-11-17 10:40:02 UTC

R topics documented:
micemd-package .......................................................... 2
CHEM97Na .............................................................. 4
find.defaultMethod .................................................. 5
Addons for the mice package to perform multiple imputation using chained equations with two-
level data. Includes imputation methods specifically handling sporadically and systematically miss-
ing values (Resche-Rigon et al. 2013). Imputation of continuous, binary or count variables are
available. Following the recommendations of Audigier, V. et al (2018), the choice of the imputation
method for each variable can be facilitated by a default choice tuned according to the structure of
the incomplete dataset. Allows parallel calculation for mice.

Author(s)

Vincent Audigier, Matthieu Resche-Rigon

Maintainer: Vincent Audigier <vincent.audigier@cnam.fr>

References

Audigier, V., White, I., Jolani, S., Debray, T., Quartagno, M., Carpenter, J., van Buuren, S. and
Resche-Rigon, M. Multiple imputation for multilevel data with continuous and binary variables

of systematically missing predictors in an individual participant data meta-analysis: a generalized

ple Imputation.

for heterogeneity and studies with missing covariates. Statistics in Medicine, 35(17):2938-2954.
doi:10.1002/sim.6837


See Also

mice

Examples

```r
require(lme4)
data(CHEM97Na)

ind.clust <- 1#index for the cluster variable

#initialisation of the argument predictorMatrix
predictor.matrix<-mice(CHEM97Na,m=1,maxit=0)$pred
predictor.matrix[ind.clust,ind.clust] <- 0
predictor.matrix[-ind.clust,ind.clust] <- -2
predictor.matrix[predictor.matrix==1] <- 2

#initialisation of the argument method
method<-find.defaultMethod(CHEM97Na,ind.clust)

#multiple imputation by chained equations (parallel calculation) [time consuming]
#res.mice <- mice.par(CHEM97Na, predictorMatrix = predictor.matrix,
#                     method=method)

#check convergence
#plot(res.mice)

#analysis (apply a generalized linear mixed effects model to each imputed dataset)
#ana <- with(res.mice, expr=glmer(Score~Sex+GSCE+(1|School),
#                                  family="poisson",
#                                  control=glmerControl(optimizer = "bobyqa")))

#check the number of generated tables
#plot(ana)

#pooling
#res.pool <- pool(ana)
#summary(res.pool)
```
**CHEM97Na**

*An incomplete two-level dataset which consists of A/AS-level examination data from England*

**Description**

This dataset is an extract of the CHEM97 dataset (Fielding, A. et al, 2003) dealing with point scores of 31,022 pupils grouped in 2,280 schools. CHEM97Na reports point score for Schools with more than 70 pupils only, i.e. 1681 pupils grouped in 18 schools. Systematically missing values and sporadically missing values have been added according to a missing completely at random (MCAR) mechanism (Little R.J.A. and Rubin D.B., 2002). Systematically missing values are values that are missing for all pupils of a same school, while sporadically missing values are values which are missing for an individual only (Resche-Rigon, et al 2013).

**Usage**

```r
data("CHEM97Na")
```

**Format**

A data frame with 1681 observations on the following 5 variables.

- **School**: a numeric indexing the School
- **Sex**: a factor with levels `M F`
- **Age**: a numeric indicating the age in months
- **GSCE**: a numeric vector indicating the point score at the General Certificate of Secondary Education
- **Score**: a numeric vector indicating the point score on A-level Chemistry in 1997

**Details**

For more details, see Fielding, A. et al (2003).

**Source**


Available at http://www.bristol.ac.uk/cmm/learning/mmsoftware/data-rev.html#chem97

**References**


See Also

matrixplot

Examples

data(CHEM97Na)

#summary
summary(CHEM97Na)

#summary per School
by(CHEM97Na,CHEM97Na$School,summary)

find.defaultMethod Suggestion of conditional imputation models to use accordingly to the incomplete dataset

Description

Provides conditionnal imputation models to use for each column of the incomplete dataset according to the number of clusters, the number of individuals per cluster and the class of the variables.

Usage

find.defaultMethod(don.na, ind.clust, I.small = 7, ni.small = 100, prop.small = 0.4)

Arguments

don.na An incomplete data frame.

ind.clust A scalar indexes the variable corresponding to the cluster indicator.

I.small A scalar that is used as threshold to consider the number of observed clusters (fully observed or partially observed) as small. Default is I.small=7.

ni.small A scalar that is used as threshold to consider the number individuals per clusters (with observed values) as small. Default is ni.small=100.

prop.small A scalar that is used as threshold to consider the number of small clusters as small. Default is prop.small=0.4.
Details

Provides conditionnal imputation models to use for each column of the incomplete dataset according to the number of clusters, the number of individuals per cluster and the class of the variable (Audigier, V. et al 2017). Returned methods can be: 2l.stage.bin (binary), 2l.stage.norm (continuous), 2l.stage.pois (integer), 2l.glm.bin (binary), 2l.glm.norm (continuous), 2l.glm.pois (integer), 2l.jomo (continuous or binary). For a given variable, the method retained is chosen according to the following decision tree:

<table>
<thead>
<tr>
<th>Few observed clusters</th>
<th>Many observed clusters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Few observed values per cluster</td>
<td>Many observed values per cluster</td>
</tr>
<tr>
<td>continuous</td>
<td>2l.glm.norm</td>
</tr>
<tr>
<td>binary</td>
<td>2l.glm.bin</td>
</tr>
<tr>
<td>integer</td>
<td>2l.glm.pois</td>
</tr>
</tbody>
</table>

For instance, with few observed clusters (i.e. less than \(I \cdot \text{small}\)), and many observed values per cluster (i.e. less than \(\text{prop} \cdot \text{small} \) clusters with less than \(ni \cdot \text{small}\) observed values), imputation of a continuous variable according to the method 2l.stage.norm will be suggested.

Value

A vector of strings with length `ncol(data)`.

Author(s)

Vincent Audigier <vincent.audigier@cnam.fr>

References

IPDNa


See Also

mice, mice.par

Examples

data(CHEM97Na)

ind.clust <- 1#index for the cluster variable

#initialisation of the argument predictorMatrix
predictor.matrix <- mice(CHEM97Na, m = 1, maxit = 0)$pred
predictor.matrix[ind.clust,ind.clust] <- 0
predictor.matrix[-ind.clust,ind.clust] <- -2
predictor.matrix[predictor.matrix==1] <- 2

#initialisation of the argument method
method <- find.defaultMethod(CHEM97Na, ind.clust)
print(method)

#multiple imputation by chained equations (parallel calculation)
#res.mice <- mice.par(CHEM97Na, m = 3, predictorMatrix = predictor.matrix, method = method)

IPDNa

A simulated Individual Patient Data (IPD) meta-analysis with missing values.

Description

This dataset is a simulated version of an IPD meta-analysis consisting of 28 studies focusing on risk factors in acute heart failure (GREAT, 2013). Each study includes a list of patient characteristics and potential risk factors. Each of them is incomplete, leading to sporadically missing values (Resche-Rigon, et al 2013). In addition, some variables have been collected on some studies only, leading to systematically missing values. More details on the original dataset are provided in Audigier et al. (2018). To mimic the real data, a general location model has been fitted on each study (Schafer, 1997). Then, each study has been generated according to the estimated parameters. Finally, missing values have been allocated similarly to the original dataset.
Usage

data("IPDNa")

Format

A data frame with 11685 observations on the following 10 variables.

- `centre` a numeric indexing the center where the study is conducted
- `gender` a factor with levels 0 1
- `bmi` a numeric vector indicating the body mass index
- `age` a numeric vector indicating the age
- `sbp` a numeric vector indicating the systolic blood pressure
- `dbp` a numeric vector indicating the diastolic blood pressure
- `hr` a numeric vector indicating the heart rate
- `lvef` a numeric vector indicating the ventricular ejection fraction
- `bnp` a numeric vector indicating the level of the brain natriuretic peptide biomarker
- `afib` a factor with levels 0 1 indicating the atrial fibrillation

Details

For more details, see Audigier et al. (2018)

Source


References


Examples

data(IPDNa)

#summary
summary(IPDNa)

#summary per study
by(IPDNa, IPDNa$centre, summary)
**Description**

Imputes univariate two-level binary variable from a logistic model. The imputation method is based on a two-stage estimator: at step 1, a logistic regression model is fitted to each observed cluster; at step 2, estimates obtained from each cluster are combined according to a linear random effect model.

**Usage**

```r
mice.impute.2l.2stage.bin(y, ry, x, type, method_est = "mm", ...)
```

**Arguments**

- `y`: Incomplete data vector of length `n`
- `ry`: Vector of missing data pattern (FALSE=missing, TRUE=observed)
- `x`: Matrix (`n x p`) of complete covariates.
- `type`: Vector of length `ncol(x)` identifying random and class variables. Random variables are identified by a '2'. The class variable (only one is allowed) is coded as '-2'. Random variables also include the fixed effect.
- `method_est`: Vector of string given the version of the estimator to use. Choose `method_est="reml"` for restricted maximum likelihood estimator or `method_est="mm"` for the method of moments. By default `method_est="mm"`.
- `...`: Other named arguments.

**Details**

Imputes univariate two-level continuous variable from a heteroscedastic normal model. The imputation method is based on a two-stage estimator: at step 1, a linear regression model is fitted to each observed cluster; at step 2, estimates obtained from each cluster are combined according to a linear random effect model. Two possibilities are available to combine estimates at stage 2: by default, parameters of the linear random effect model are estimated according to the method of moments (MM), otherwise, parameters of the linear random effect model can be estimated according to the restricted maximum likelihood estimator (REML). The variability on the parameters of the imputation is propagated according to an asymptotic strategy requiring a large number of clusters. Compared to the REML version, the MM version is quicker to perform, but it provides less theoretical guarantees. Nevertheless, simulation studies show that both versions lead to similar inferences (Audigier et al, 2018; Resche-Rigon, M. and White, I. R., 2016).

**Value**

A vector of length `nmis` with imputations.
Author(s)

Vincent Audigier <vincent.audigier@cnam.fr>

References


See Also

mice,mice.impute.2l.glm.bin,mice.impute.2l.jomo

mice.impute.2l.2stage.heckman

Imputation based on Heckman model for multilevel data.

Description

Imputes both outcome or predictor incomplete variables that follow an indirectly non-ignorable Missing Not at Random (MNAR) mechanism, i.e., the likelihood of a missing value in the incomplete variable depends on other unobserved variable(s) that are also correlated with the incomplete variable. This imputation is based on Heckman’s selection model and is suitable for multilevel databases, such as individual participant data, with both systematic or sporadic missing data.

Usage

mice.impute.2l.2stage.heckman(y, ry, x, wy = NULL, type, pmm = FALSE, ypmm = NULL, meta_method = "reml", pred_std = FALSE,...)

Arguments

y Vector to be imputed
ry A logical vector of length length(y) indicating the subset y[ry] of elements in y to which the imputation model is fitted. The ry generally distinguishes the observed (TRUE) and missing values (FALSE) in y.
x A numeric design matrix with length(y) rows with predictors for y. Matrix x may have no missing values.
wy A logical vector of length length(y). A TRUE value indicates locations in y for which imputations are created.
mice.impute.2l.2stage.heckman

**type**
Type of the variable in the prediction model, which can be one of the following:
- No predictor (0), Cluster variable (-2), Predictor in both the outcome and selection equation (2), Predictor only in the selection equation (-3), Predictor only in the outcome equation (-4). In this method all predictors are considered random variables that also included the fixed effect.

**pmm**
A logical value that specifies whether the predictive mean matching method is applied. (default = "FALSE"). This method is only applicable to missing continuous variables.

**ypmm**
A continuous vector of donor values for y used in the predictive mean matching method. If ypmm is not provided, the observable values of y are used as donors.

**meta_method**
A character value that indicates the method for estimating meta-analysis random effects: "ml" (maximum likelihood), "reml" (restricted maximum likelihood) or "mm" (method of moments).

**pred_std**
A logical value that indicates whether internally standardize the set of predictor variables (default = FALSE).

... Other named arguments. Not used.

**Details**
This function imputes systematically and sporadically missing binary and continuous univariate variables that follow an MNAR mechanism according to the Heckman selection model. It is specifically designed for clustered datasets. The imputation method employs a two-stage approach in which the Heckman model parameters at the cluster level are estimated using the copula method.

**Value**
Vector with imputed data, of type binary or continuous type

**Note**
Missing binary variables should be included as two-level factor type variables in the incomplete dataset. The cluster variable should be included as a numeric variable in the dataset. When the cluster variable is not specified, the imputation method defaults to a simple Heckman model, which does not take in account the hierarchical structure. In cases where the Heckman model cannot be estimated at the hierarchical level, the imputation method reverts to the simple Heckman model.

**Author(s)**
Julius Center Methods Group UMC, 2022 <J.MunozAvila@umcutrecht.nl>

**References**

See Also

mice

Examples

```r
require(mice)
require(nlme)
require(broom.mixed)
require(parallel)

# Load dataset
data(Obesity)

# Define imputation methods for each incomplete variables
meth <- find.defaultMethod(Obesity, ind.clust = 1)

# Modify some of the proposed imputation methods
# Deterministic imputation
meth['BMI'] <- "~ I(Weight / (Height)^2)"
meth['Age'] <- "2l.2stage.pmm"

# Set method, here Weight variable is assumed an MNAR variable
# Weight imputed with the Heckman method
meth['Weight'] <- "2l.2stage.heckman"

# Set type of predictor variable,
# All covariates are included in both outcome and selection equation
ini <- mice(Obesity, maxit = 0)
pred <- ini$pred
pred[, "Time"] <- 0
pred[, "Cluster"] <- -2
pred[pred == 1] <- 2

# Time was used as exclusion restriction variable
pred['Weight', "Time"] <- -3
# Deterministic imputation, to avoid circular predictions
pred[c("Height", "Weight"), "BMI"] <- 0

# Imputation of continuous variables (time consumming)

# nnodes <- detectCores()
# imp <- mice.par(Obesity, meth = meth, pred = pred, m=10, seed = 123,
#                n.nodes = nnodes)
# summary(complete(imp,"long")$Weight)

# Imputation of continuous variables using the predictor mean matching method.
# Imputed values fall within the range of observable variables.

# imp_pmm <- mice.par(Obesity, meth = meth, pred = pred, m = 10,
#                    seed = 123, pmm=TRUE, n.nodes = nnodes)
# summary(complete(imp_pmm,"long")$Weight)
```
# Fit the model

```r
# model_MNAR <- with(imp,lme( BMI ~ Age + FamOb + Gender,random=~1+Age|Cluster))
# model_MNAR_pmm <- with(imp_pmm,lme( BMI ~ Age + FamOb + Gender,random=~1+Age|Cluster))

# summary(pool(model_MNAR))
# summary(pool(model_MNAR_pmm))
```

---

### Description

Imputes univariate two-level continuous variable from a heteroscedastic normal model. The imputation method is based on a two-stage estimator: at step 1, a linear regression model is fitted to each observed cluster; at step 2, estimates obtained from each cluster are combined according to a linear random effect model.

### Usage

```r
mice.impute.2l.2stage.norm(y, ry, x, type, method_est = "mm", ...)  
```

### Arguments

- `y`: Incomplete data vector of length `n`
- `ry`: Vector of missing data pattern (FALSE=missing, TRUE=observed)
- `x`: Matrix `(n x p)` of complete covariates.
- `type`: Vector of length `ncol(x)` identifying random and class variables. Random variables are identified by a '2'. The class variable (only one is allowed) is coded as '-2'. Random variables also include the fixed effect.
- `method_est`: Vector of string given the version of the estimator to used. Choose `method_est="reml"` for restricted maximum likelihood estimator or `method_est="mm"` for the method of moments. By default `method_est="mm"`.
- `...`: Other named arguments.

### Details

Imputes univariate two-level continuous variable from a heteroscedastic normal model. The imputation method is based on a two-stage estimator: at step 1, a linear regression model is fitted to each observed cluster; at step 2, estimates obtained from each cluster are combined according to a linear random effect model. Two possibilities are available to combine estimates at stage 2: by default, parameters of the linear random effect model are estimated according to the method of moments (MM), otherwise, parameters of the linear random effect model can be estimated according to the restricted maximum likelihood estimator (REML). The variability on the parameters of the
imputation is propagated according to an asymptotic strategy requiring a large number of clusters. Compared to the REML version, the MM version is quicker to perform, but it provides less theoretical guarantees. Nevertheless, simulation studies show that both versions lead to similar inferences (Resche-Rigon, M. and White, I. R. (2016)).

**Value**

A vector of length \( n_{\text{mis}}\) with imputations.

**Author(s)**

Vincent Audigier <vincent.audigier@cnam.fr>

**References**


**See Also**

mice, mice.impute.2l.2stage.pmm, mice.impute.2l.glm.norm, mice.impute.2l.jomo

---

**mice.impute.2l.2stage.pmm**

*Predictive mean matching imputation for two-level variable*

**Description**

Similarly to mice.impute.2l.stage.norm, this function imputes univariate two-level continuous variable from a heteroscedastic normal model. The difference consists in replacing missing values by observed values instead of adding a parametric noise to the prediction of a linear model with random effects (as done in mice.impute.2l.stage.norm.mm and mice.impute.2l.stage.norm.reml).

**Usage**

```r
mice.impute.2l.2stage.pmm(y, ry, x, type,
    method_est = "mm",
    incluster = FALSE,
    kpmm = 5, ...)
```
Arguments

- **y**: Incomplete data vector of length n.
- **ry**: Vector of missing data pattern (FALSE=missing, TRUE=observed).
- **x**: Matrix \((n \times p)\) of complete covariates.
- **type**: Vector of length \(\text{ncol}(x)\) identifying random and class variables. Random variables are identified by a ’2’. The class variable (only one is allowed) is coded as ’-2’. Random variables also include the fixed effect.
- **method_est**: Vector of string given the version of the estimator to used. Choose \(\text{method_est}="\text{reml}"\) for restricted maximum likelihood estimator or \(\text{method_est}="\text{mm}"\) for the method of moments. By default \(\text{method_est}="\text{mm}"\).
- **incluster**: Boolean indicating if the imputed values are drawn from the cluster or from the full dataset. By default imputed values are drawn from all available clusters \(\text{incluster}=\text{FALSE}\).
- **kpmm**: The size of the donor pool among which a draw is made. The default is \(k = 5\).
- **...**: Other named arguments.

Details

Imputes univariate two-level continuous variable from observed values. The imputation method is based on a two-stage estimator: at step 1, a linear regression model is fitted to each observed cluster; at step 2, estimates obtained from each cluster are combined according to a linear random effect model. To combine estimates at stage 2, parameters of the linear random effect model are estimated according to the method of moments or according to the restricted maximum likelihood estimator. The variability on the parameters of the imputation is propagated according to an asymptotic strategy requiring a large number of clusters. The sample variability is reflected by using a predictive mean matching approach, meaning that missing values are imputed by a draw from observed values. The pool of \(k\) donors is defined according to the Manhattan distance between the prediction of the observation which is imputed and the predictions of other available observations (matching of type 2). The pool can be restricted to the cluster of the individual that is imputed or from all clusters. By drawing values inside the cluster, the heteroscedasticity assumption is preserved. Otherwise, the sample variability of imputed values is the same for all clusters, which strengthen the homoscedasticity assumption. Among the pool of \(k\) donors, the selected one is drawn at random.

Value

Numeric vector of length \(\text{sum}(\text{!ry})\) with imputations

Note

This method is experimental.

Author(s)

Vincent Audigier <vincent.audigier@cnam.fr>
References


See Also

mice.impute.2l.2stage.norm

mice.impute.2l.2stage.pois

Imputation by a two-level Poisson model based on a two-stage estimator

Description

Imputes univariate two-level count variable from a Poisson model. The imputation method is based on a two-stage estimator: at step 1, a Poisson regression model is fitted to each observed cluster; at step 2, estimates obtained from each cluster are combined according to a linear random effect model.

Usage

mice.impute.2l.2stage.pois(y, ry, x, type, method_est = "mm", ...)

Arguments

y Incomplete data vector of length n
ry Vector of missing data pattern (FALSE=missing, TRUE=observed)
x Matrix (n x p) of complete covariates.
type Vector of length ncol(x) identifying random and class variables. Random variables are identified by a '2'. The class variable (only one is allowed) is coded as '-2'. Random variables also include the fixed effect.
method_est Vector of string given the version of the estimator to used. Choose method_est="rem1" for restricted maximum likelihood estimator or method_est="mm" for the method of moments. By default method_est="mm".
... Other named arguments.
mice.impute.2l.glm.bin

Details

Imputes univariate two-level count variable from a Poisson model. The imputation method is based on a two-stage estimator: at step 1, a Poisson regression model is fitted to each observed cluster; at step 2, estimates obtained from each cluster are combined according to a linear random effect model. Two possibilities are available to combine estimates at stage 2: by default, parameters of the linear random effect model are estimated according to the method of moments (MM), otherwise, parameters of the linear random effect model can be estimated according to the restricted maximum likelihood estimator (REML). The variability on the parameters of the imputation is propagated according to an asymptotic strategy requiring large samples and a large number of clusters. Compared to the REML version, the MM version is quicker to perform, but it provides less theoretical guarantees. Nevertheless, simulation studies show that both versions lead to similar inferences (Audigier et al, 2018; Resche-Rigon and White, 2016).

Value

A vector of length \( nmis \) with imputations.

Author(s)

Vincent Audigier <vincent.audigier@cnam.fr>

References


See Also

mice, mice.impute.2l.glm.pois

mice.impute.2l.glm.bin

Imputation of univariate missing data using a Bayesian logistic mixed model based on non-informative prior distributions

Description

Imputes univariate missing data using a Bayesian logistic mixed model based on non-informative prior distributions. The method is dedicated to a binary outcome stratified in several clusters. Should be used with few clusters and few individuals per cluster. Can be very slow to perform otherwise.
mice.impute.2l.glm.bin

Usage

mice.impute.2l.glm.bin(y, ry, x, type, ...)

Arguments

y  Incomplete data vector of length n
ry  Vector of missing data pattern (FALSE=missing, TRUE=observed)
x  Matrix (n x p) of complete covariates.
type  Vector of length ncol(x) identifying random and class variables. Random variables are identified by a '2'. The class variable (only one is allowed) is coded as '-2'. Random variables also include the fixed effect.
...  Other named arguments.

Details

Imputes univariate missing data using a Bayesian logistic mixed model based on non-informative prior distributions. The variability on the parameters of the imputation is propagated according to an explicit Bayesian modelling. More precisely, improper prior distributions are used for regression coefficients and covariance matrix of random effects. The method is recommended for datasets with a small number of clusters and a small number of individuals per cluster. Otherwise, the method can be very slow to perform.

Value

A vector of length nmis with imputations.

Author(s)

Vincent Audigier <vincent.audigier@cnam.fr> from the R code of Shahab Jolani.

References


See Also

mice, mice.impute.2l.2stage.bin, mice.impute.2l.jomo
Imputation of univariate missing data using a Bayesian linear mixed model based on non-informative prior distributions

**Description**

Imputes univariate missing data using a Bayesian linear mixed model based on non-informative prior distributions. The method is dedicated to a continuous outcome stratified in several clusters. Should be used with few clusters and few individuals per cluster. Can be very slow to perform otherwise.

**Usage**

```r
mice.impute.2l.glm.norm(y, ry, x, type, ...)  
```

**Arguments**

- `y` Incomplete data vector of length `n`
- `ry` Vector of missing data pattern (FALSE=missing, TRUE=observed)
- `x` Matrix (`n x p`) of complete covariates.
- `type` Vector of length `nrow(x)` identifying random and class variables. Random variables are identified by a ’2’. The class variable (only one is allowed) is coded as ’-2’. Random variables also include the fixed effect.
- `...` Other named arguments.

**Details**

Imputes univariate two-level continuous variable from a homoscedastic normal model. The variability on the parameters of the imputation is propagated according to an explicit Bayesian modelling. More precisely, improper prior distributions are used for regression coefficients and variances components. The method is recommended for datasets with a small number of clusters and a small number of individuals per cluster. Otherwise, confidence intervals after applying analysis methods on the multiply imputed dataset tend to be anti-conservative. In addition, the imputation can be highly time consuming.

**Value**

A vector of length `nmis` with imputations.

**Author(s)**

Vincent Audigier (<vincent.audigier@cnam.fr> from the R code of Shahab Jolani.)
mice.impute.2l.glm.pois

References


See Also

mice,mice.impute.2l.2stage.norm,mice.impute.2l.jomo

mice.impute.2l.glm.pois

Imputation of count variable using a Bayesian mixed model based on non-informative prior distributions

Description

Imputes univariate missing data using a Bayesian mixed model (Poisson regression) based on non-informative prior distributions. The method is dedicated to a count outcome stratified in several clusters. Should be used with few clusters and few individuals per cluster. Can be very slow to perform otherwise.

Usage

mice.impute.2l.glm.pois(y, ry, x, type,...)

Arguments

y  Incomplete data vector of length n
ry  Vector of missing data pattern (FALSE=missing, TRUE=observed)
x  Matrix (n x p) of complete covariates.
type  Vector of length ncol(x) identifying random and class variables. Random variables are identified by a ’2’. The class variable (only one is allowed) is coded as ’-2’. Random variables also include the fixed effect.
...  Other named arguments.
**Details**

Imputes univariate missing data using a Bayesian mixed model (Poisson regression) based on non-informative prior distributions. The variability on the parameters of the imputation is propagated according to an explicit Bayesian modelling. More precisely, improper prior distributions are used for regression coefficients and variances components. The method is recommended for datasets with a small number of clusters and a small number of individuals per cluster. Otherwise, the method can be very slow to perform.

**Value**

A vector of length \( n_{mis} \) with imputations.

**Author(s)**

Vincent Audigier &lt;vincent.audigier@cnam.fr&gt; from the R code of Shahab Jolani.

**References**


**See Also**

mice, mice.impute.2l.2stage.pois

---

**mice.impute.2l.jomo**

*Imputation of univariate missing data by a Bayesian multivariate generalized model based on conjugate priors*

**Description**

Univariate imputation by a Bayesian multivariate generalized model based on conjugate priors. Can be used for a continuous or binary incomplete variable. For continuous variables, the modelling assumes heteroscedasticity for errors. For a binary variable, a probit link and a latent variables framework are used. The method should be used for a variable with large number of clusters and a large number of individuals per cluster.

**Usage**

mice.impute.2l.jomo(y, ry, x, type, nburn = 200, ...)

---
Arguments

- **y**: Incomplete data vector of length \( n \)
- **ry**: Vector of missing data pattern (FALSE=missing, TRUE=observed)
- **x**: Matrix \((n \times p)\) of complete covariates.
- **type**: Vector of length \( ncol(x) \) identifying random and class variables. Random variables are identified by a '2'. The class variable (only one is allowed) is coded as '-2'. Random variables also include the fixed effect.
- **nburn**: A scalar indicating the number of iterations for the Gibbs sampler. Default is \( nburn=200 \)
- **...**: Other named arguments.

Details

Contrary to the approach developed in the R jomo package, the imputation is here sequentially performed through a FCS approach, instead of imputing all variables simultaneously. The motivation for such a method is that jomo presents some advantages over other imputation methods, but not always for any type of variables (binary or continuous). By proposing a FCS version of jomo, we allow imputation of mixed variables (continuous and binary), while taking the best of jomo and of other imputation methods. To impute one variable according to this method, other variables are assumed to be full, like in any FCS approach. The imputation function is a direct use of the R function jomo1ran from the jomo package. The argument \( meth \) is tuned to "random" to allow covariance matrices drawn from an inverse Wishart distribution. Only intercept are considered in covariates \((X=NULL \text{ and } Z=NULL)\), while the multivariate outcome corresponds to all variables of the datasets.

Value

A vector of length \( nmis \) with imputations.

Author(s)

Vincent Audigier <vincent.audigier@cnam.fr> from the R code of Matteo Quartagno.

References


See Also

mice, jomo1ran
Parallel calculations for Multivariate Imputation by Chained Equations using the R package parallel.

Usage

```r
mice.par(don.na, m = 5, method = NULL, predictorMatrix, where = NULL,
visitSequence = NULL, blots = NULL, post = NULL, blocks, formulas,
defaultMethod = c("pmm", "logreg", "polyreg", "polr"), maxit = 5,
seed = NA, data.init = NULL, nnodes = 5, path.outfile = NULL, ...)
```

Arguments

don.na A data frame or a matrix containing the incomplete data. Missing values are coded as NA.

m Number of multiple imputations. The default is m=5.

method Can be either a single string, or a vector of strings with length ncol(data), specifying the elementary imputation method to be used for each column in data. If specified as a single string, the same method will be used for all columns. The default imputation method (when no argument is specified) depends on the measurement level of the target column and are specified by the defaultMethod argument. Columns that need not be imputed have the empty method "". See details for more information.

predictorMatrix A square matrix of size ncol(data) containing 0/1 data specifying the set of predictors to be used for each target column. Rows correspond to target variables (i.e. variables to be imputed), in the sequence as they appear in data. A value of '1' means that the column variable is used as a predictor for the target variable (in the rows). The diagonal of predictorMatrix must be zero. The default for predictorMatrix is that all other columns are used as predictors (sometimes called massive imputation). Note: For two-level imputation codes '2' and '-2' are also allowed.

where A data frame or matrix with logicals of the same dimensions as data indicating where in the data the imputations should be created. The default, where = is.na(data), specifies that the missing data should be imputed. The where argument may be used to overimpute observed data, or to skip imputations for selected missing values.

visitSequence A vector of integers of arbitrary length, specifying the column indices of the visiting sequence. The visiting sequence is the column order that is used to impute the data during one pass through the data. A column may be visited more than once. All incomplete columns that are used as predictors should be visited, or else the function will stop with an error. The default sequence 1:ncol(data)
implies that columns are imputed from left to right. It is possible to specify one of the keywords 'roman' (left to right), 'arabic' (right to left), 'monotone' (sorted in increasing amount of missingness) and 'revmonotone' (reverse of monotone). The keyword should be supplied as a string and may be abbreviated.

**blots**
A named list of `alist`'s that can be used to pass down arguments to lower level imputation function. The entries of element `blots[[blockname]]` are passed down to the function called for block `blockname`.

**post**
A vector of strings with length `ncol(data)`, specifying expressions. Each string is parsed and executed within the `sampler()` function to postprocess imputed values. The default is to do nothing, indicated by a vector of empty strings ''. 

**blocks**
List of vectors with variable names per block. List elements may be named to identify blocks. Variables within a block are imputed by a multivariate imputation method (see `method` argument). By default each variable is placed into its own block, which is effectively fully conditional specification (FCS) by univariate models (variable-by-variable imputation). Only variables whose names appear in blocks are imputed. The relevant columns in the `where` matrix are set to FALSE of variables that are not block members. A variable may appear in multiple blocks. In that case, it is effectively re-imputed each time that it is visited.

**formulas**
A named list of formula's, or expressions that can be converted into formula's by `as.formula`. List elements correspond to blocks. The block to which the list element applies is identified by its name, so list names must correspond to block names. The `formulas` argument is an alternative to the `predictorMatrix` argument that allows for more flexibility in specifying imputation models, e.g., for specifying interaction terms.

**defaultMethod**
A vector of three strings containing the default imputation methods for numerical columns, factor columns with 2 levels, and columns with (unordered or ordered) factors with more than two levels, respectively. If nothing is specified, the following defaults will be used: `pmm`, predictive mean matching (numeric data) `logreg`, logistic regression imputation (binary data, factor with 2 levels) `polyreg`, polytomous regression imputation for unordered categorical data (factor >= 2 levels) `polr`, proportional odds model for (ordered, >= 2 levels)

**maxit**
A scalar giving the number of iterations. The default is 5.

**seed**
An integer that is used as argument by the `set.seed()` for offsetting the random number generator. Default is to leave the random number generator alone.

**data.init**
A data frame of the same size and type as `data`, without missing data, used to initialize imputations before the start of the iterative process. The default `NULL` implies that starting imputation are created by a simple random draw from the data. Note that specification of `data.init` will start the `m` Gibbs sampling streams from the same imputations.

**nnodes**
A scalar indicating the number of nodes for parallel calculation. Default value is 5.

**path.outfile**
A vector of strings indicating the path for redirection of print messages. Default value is `NULL`, meaning that silent imputation is performed. Otherwise, print messages are saved in the files `path.outfile/output.txt`. One file per node is generated.

... Named arguments that are passed down to the elementary imputation functions.
Details
Performs multiple imputation of \( m \) tables in parallel by generating \( m \) seeds, and then by performing multiple imputation by chained equations in parallel from each one. The output is the same as the `mice` function of the mice package.

Value
Returns an S3 object of class `mids` (multiply imputed data set)

Author(s)
Vincent Audigier <vincent.audigier@cnam.fr>

References


See Also
`mice`, `parallel`

Examples

```
# # nhanes (one level data)
#
# data(nhanes, package = "mice")
# imp <- mice.par(nhanes)
# fit <- with(data = imp, exp = lm(bmi ~ hyp + chl))
# summary(pool(fit))
```

```
# CHEM97Na (Two levels data with 1681 observations and 5 variables)
```
data(CHEM97Na)

ind.clust<-1  # index for the cluster variable

# initialisation of the argument predictorMatrix
predictor.matrix<-mice(CHEM97Na,m=1,maxit=0)$pred
predictor.matrix[ind.clust,ind.clust]<-0
predictor.matrix[-ind.clust,ind.clust]<- -2
predictor.matrix[predictor.matrix==1]<-2

# initialisation of the argument method
method<-find.defaultMethod(CHEM97Na,ind.clust)

# multiple imputation by chained equations (parallel calculation) [1 minute]
# (the imputation process can be followed by opening output.txt files in the working directory)
# res.mice<-mice.par(CHEM97Na,
# predictorMatrix = predictor.matrix,
# method=method,
# path.outfile=getwd())

# multiple imputation by chained equations (without parallel calculation) [4.8 minutes]
# res.mice<-mice(CHEM97Na,
# predictorMatrix = predictor.matrix,
# method=method)

IPDNa (Two levels data with 11,685 observations and 10 variables)

data(IPDNa)

ind.clust<-1  # index for the cluster variable

# initialisation of the argument predictorMatrix
predictor.matrix<-mice(IPDNa,m=1,maxit=0)$pred
predictor.matrix[ind.clust,ind.clust]<-0
predictor.matrix[-ind.clust,ind.clust]<- -2
predictor.matrix[predictor.matrix==1]<-2

# initialisation of the argument method
method<-find.defaultMethod(IPDNa,ind.clust)

# multiple imputation by chained equations (parallel calculation)
# res.mice<-mice.par(IPDNa,
# predictorMatrix = predictor.matrix,
# method=method,
# path.outfile=getwd())
Description

This synthetic dataset was generated from an online survey on obesity, which collected information on the dietary behavior of 2111 participants. We made the assumption that the data was gathered from five distinct locations or clusters. To account for potential selection bias in the responses related to weight, we simulated the values and observability of this variable using the Heckman selection model within a hierarchical structure.

Additionally, we assumed that in one of the locations, the weight variable was systematically missing. We also introduced missing values for some other variables in the dataset using a Missing at Random (MAR) mechanism.

Format

A dataframe with 2111 observations with the following variables:

- Gender: a factor variable with two levels: 1 ("Female"), 0 ("Male").
- Age: a numeric variable indicating the subject’s age in years.
- Height: a numeric value with Height in meters.
- FamOb: a factor variable describing the subject’s family history of obesity with two levels: 1("Yes"), 0("No").
- Weight: a numeric variable indicating the subject’s weight in kilograms.
- Time: a numeric variable indicating the time taken by the subject to respond to the surveys questions in minutes.
- BMI: a numeric variable with the subject’s body mass index.
- Cluster: a numeric variable indexing the cluster.

Details


Source

Synthetic data based on the data retrieved from "https://www.kaggle.com/datasets/fabinmndez/obesitydata/"

References

Palechor, F. M., & de la Hoz Manotas, A. (2019). Dataset for estimation of obesity levels based on eating habits and physical condition in individuals from Colombia, Peru and Mexico. Data in brief, 25, 104344.

Examples

```r
library(mice)
library(ggplot2)
library(data.table)
```
data(Obesity)
summary(Obesity)
md.pattern(Obesity)

# Missingness per region (Weight)
dataNA <- setDT(Obesity)[, .(nNA = sum(is.na(Weight)), n=.N), by = Cluster]
dataNA[, propNA:=nNA/n]
dataNA

# Density per region (Weight)
Obesity$Cluster <- as.factor(Obesity$Cluster)
ggplot(Obesity, aes(x = Weight, group=Cluster)) +
  geom_histogram(aes(color = Cluster, fill= Cluster),
                position = "identity", bins = 30) +
  facet_grid(Cluster~.)

overimpute

Overimputation diagnostic plot

Description
Assess the fit of the predictive distribution after performing multiple imputation with mice

Usage
overimpute(res.mice, plotvars = NULL, plotinds = NULL, 
           nnodes = 5, path.outfile = NULL, alpha = 0.1)

Arguments

res.mice              An object of class mids
plotvars             column index of the variables overimputed
plotinds             row index of the individuals overimputed
nnodes               A scalar indicating the number of nodes for parallel calculation. Default value is 5.
path.outfile         A vector of strings indicating the path for redirection of print messages. Default value is NULL, meaning that silent imputation is performed. Otherwise, print messages are saved in the files path.outfile/output.txt. One file per node is generated.
alpha                alpha level for prediction intervals
Details

This function imputes each observed values from each of the parameters of the imputation model obtained from the mice procedure. The comparison between the "overimputed" values and the observed values is made by building a confidence interval for each observed value using the quantiles of the overimputed values (Blackwell et al. (2015)). Note that confidence intervals built with quantiles require a large number of imputations. If the model fits the data well, then the 90% confidence interval should contain the observed value in 90% of the cases (the proportion of intervals containing the observed value is reported in the title of each graph). The function overimpute takes as an input the output of the mice or mice.par function (res.mice), the indices of the incomplete continuous variables that are plotted (plotvars), the indices of individuals (can be useful for time consuming imputation methods), the number of nodes for parallel computation, and the path for exporting print message generated during the parallel process.

Value

A list of two matrices

res.plot 7-columns matrix that contains (1) the variable which is overimputed, (2) the observed value of the observation, (3) the mean of the overimputations, (4) the lower bound of the confidence interval of the overimputations, (5) the upper bound of the confidence interval of the overimputations, (6) the proportion of the other variables that were missing for that observation in the original data, and (7) the color for graphical representation.

res.values A matrix with overimputed values for each cell. The number of columns corresponds to the number of values generated (i.e. the number of imputed tables)

Author(s)

Vincent Audigier <vincent.audigier@cnam.fr>

References


See Also

mice,parallel, mice.par

Examples

require(parallel)
nnodes<-detectCores()-1#number of nodes
m<-1000#nb generated values per observation

################
#one level data
################
require(mice)
data(nhanes)
#res.mice<-mice.par(nhanes,m = m,nnodes = nnodes)
#res.over<-overimpute(res.mice, nnodes = nnodes)

################
#two level data (time consuming)
################
data(CHEM97Na)

ind.clust<-1 #index for the cluster variable

#initialisation of the argument predictorMatrix
predictor.matrix<-mice(CHEM97Na,m=1,maxit=0)$pred
predictor.matrix[ind.clust,ind.clust]<-0
predictor.matrix[-ind.clust,ind.clust]< -2
predictor.matrix[predictor.matrix==1]<2

#initialisation of the argument method
method<-find.defaultMethod(CHEM97Na,ind.clust)

#multiple imputation by chained equations (time consuming)
#res.mice<-mice.par(CHEM97Na,
#    predictorMatrix = predictor.matrix,
#    method=method,m=m,nnodes = nnodes)

#overimputation on 30 individuals
#res.over<-overimpute(res.mice,
#    nnodes=nnodes,
#    plotinds=sample(x = seq(nrow(CHEM97Na)),size = 30))

---

plot.mira

Graphical investigation for the number of generated datasets

**Description**

The plot method for a mira object plots the confidence interval length against the number of multiply imputed datasets from 2 to \( m \). This is a graphical tool to check if the variability due to the simulation of the multiple imputation process can be substantially reduced by increasing the number of generated datasets \( m \).

**Usage**

```r
## S3 method for class 'mira'
plot(x, ...)
```
Arguments

- **x**: An object of class `mira`.
- **...**: Extra arguments for `plot.mira`.

Author(s)

Vincent Audigier <vincent.audigier@cnam.fr>

References


See Also

- `mice`
- `mira`

Examples

```r
require(nlme)
data(CHEM97Na)

ind.clust<-1#index for the cluster variable

#initialisation of the argument predictorMatrix
predictor.matrix<-mice(CHEM97Na,m=1,maxit=0)$pred
predictor.matrix[ind.clust,ind.clust]<-0
predictor.matrix[-ind.clust,ind.clust]<- -2
predictor.matrix[predictor.matrix==1]<-2

#initialisation of the argument method
method<-c("", "2l.2stage.bin", "2l.2stage.pois", "2l.2stage.norm", "") #quickest methods

#multiple imputation by chained equations (parallel calculation)
res.mice<-mice.par(CHEM97Na,m=15,predictorMatrix = predictor.matrix,method=method)

#analysis (apply a linear mixed effects model to each imputed dataset)
ana<-with(res.mice,expr=lme(fixed=formula(Score~Sex+GSCE+Age),
    random=formula(~1|School),method="REML",
    control=list(maxIter=100,msMaxIter=100,niterEM=25)))

#graphical investigation for the number of generated datasets m
#plot(ana)
```
Index

* datasets
  CHEM97Na, 4
  IPDNa, 7
  Obesity, 27
  * imputation
    micemd-package, 2
  * meta-analyse
    micemd-package, 2
  * mice
    find.defaultMethod, 5
    mice.impute.2l.2stage.bin, 9
    mice.impute.2l.2stage.norm, 13
    mice.impute.2l.2stage.pois, 16
    mice.impute.2l.glm.bin, 17
    mice.impute.2l.glm.norm, 19
    mice.impute.2l.glm.pois, 20
    mice.impute.2l.jomo, 21
    mice.par, 23
    overimpute, 28
  * multilevel
    micemd-package, 2
  * package
    micemd-package, 2

CHEM97Na, 4
find.defaultMethod, 5
IPDNa, 7
jomo1ran, 22
matrixplot, 5
mice, 3, 7, 10, 12, 14, 17, 18, 20–22, 25, 29, 31
mice.impute.2l.2stage.bin, 9, 18
mice.impute.2l.2stage.heckman, 10
mice.impute.2l.2stage.norm, 13, 16, 20
mice.impute.2l.2stage.pmm, 14, 14
mice.impute.2l.2stage.pois, 16, 21
mice.impute.2l.glm.bin, 10, 17
mice.impute.2l.glm.norm, 14, 19
mice.impute.2l.glm.pois, 17, 20
mice.impute.2l.jomo, 10, 14, 18, 20, 21
mice.par, 7, 23, 29
micemd (micemd-package), 2
micemd-package, 2
mids, 25
mira, 31
Obesity, 27
overimpute, 28
parallel, 25, 29
plot.mira, 30, 31