Package 'SensMice'

October 4, 2010

Type Package

Title Multivariate Imputation by Chained Equations (Iteration Step for sensitivity analysis)

Version 1.0

Date 2010-07-07

Author Noemie

Maintainer Noemie Resseguier <noemie.resseguier@yahoo.fr>

Description Takes a mids object, and produces a new object of class mids

Depends mice

License GPL

URL http://lertim.fr/Members/rgiorgi/DossierPublic/fonctions-r-s/

Encoding latin1

Collate 'SensMice.r'

R topics documented:

`sens.mice` *Multivariate Imputation by Chained Equations (Iteration Step for sensitivity analysis).*

Description

Takes a `mids` object, and produces a new object of class `mids`

Usage

```
sens.mice(IM, ListMethod=ListMethod, SupPar=SupPar)
```

Arguments

IM	An object of class mids, typically produces by a previous call to mice() or mice.mids()
ListMethod	A vector of strings with length ncol(IM$data), specifying the colum(s) in data which has (have) to be imputed with a different imputation model
SupPar	A vector of numbers, specifying the supplementary parameters to be added to the imputation model(s)

Details

This function is based on the mice function's principle, which uses the MICE algorithm. mice allows to generate multiple imputations for incomplete multivariate data by Gibbs sampling. The algorithm imputes an incomplete colum (the target column) by generating 'plausible' synthetic values given other columns in the data. Each incomplete column must act as a target column, and has its own set of predictors. The default set of predictors for a given target consists of all others colums in the data. For predictors that are incomplete themselves, the most recently generated imputations are used to complete the predictors prior to imputation of the target column.

Built-in elementary imputation methods are: pmm Bayesian linear regression (numeric) norm Predictive mean matching (numeric) norm.nob Linear regression ignoring model error (numeric) mean Unconditional mean imputation (numeric) 2l.norm Two-level normal imputation (numeric) logreg Logistic regression (factor, 2 categories) polyreg Polytomous (unordered) logistic regression (factor, >=2 categories) lda Linear discriminant analysis (factor, >=2 categories) sample Random sample from the observed values (any)

This function enables to impute missing values under the hypothesis of MNAR data, for one or more variable(s). A sensitivity analysis can be performed using the mids object returned by the function.

It goes through 3 steps:

- 1. Estimate the parameters of the imputation model under ignorable missing data hypothesis using the function mice. The fitted imputation model depends on the type of the variable which contains missing values, i.e. bayesian linear regression for numeric variables, logistic regression for binary variables, polytomous unordered logistic regression for categorical variables.

- 2. Modify the imputation model according to the explored scenario by specifying supplementary parameters as arguments for the sens.mice function. Without fitting a mixture model, we appeal to its principle as proposed by Rubin. Indeed, the addition of this(these) supplementary parameter(s) allows to specify that the distribution of the variable of interest is different among subjects with missing value and among subjects witouht missing value, conditionnally on all variables included in the imputation model. The direction and the size of this difference is expressed by the supplementary parameter in the imputation model.

- 3. Impute the missing data using the function sens.mice, resulting into a mids object that contains m newly imputed data sets under MNAR hypothesis. For this step, missing values are imputed with the usual MICE algorithm, using the previously modified imputation model.

The sensitivity analysis can then be performed on the returned mids object, which contains the newly imputed data, according to the assumption that values for the variabl(s) imputed with the sens.mice function come from two different distributions (for responders and incomplete responders).

Built-in elementary imputation methods are:

- pmm: Predictive mean matching (numeric)

- norm: Bayesian linear regression (numeric)

- logreg: Logistic regression (factor, 2 categories)

- polyreg: Polytomous (unordered) regression (factor, >= 2 categories)

ListMethod is a vector of strings with length ncol(IM$data), specifying the column(s) in data which has (have) to be imputed with an imputation model accounting for MNAR data. For variables which have to be imputed with a modified imputation model, the method has to be specified as "MyFunc". For variables which do not have to be imputed with a different imputation model, the method has to be be specified as "".

SupPar is a vector of numbers, specifying the supplementary parameters to be added to the imputation model(s). For logreg and polyreg methods, the supplementary parameters are expressed as odds-ratios (correponding to the excess of risk to present the modality of interest for non responders as compared to responders). The value for the reference need not be specified. For pmm and norm methods, the supplementary parameters are the difference between the expected values in responders and non responders.

Value

Returns an object of class mids (multiply imputed data set) with usual components

Author(s)

Noemie Resseguier, with contributions of Roch Giorgi, David Hajage, Yann De Rycke and Xavier Paoletti

References

Resseguier, N., Giorgi, R. and Paoletti, X. (submitted) *How to perform a senstivity analysis exploring the impact of missing not at random data under different sceanrios of non response mechanism with the R software.*

Rubin, D.B. *Multiple Imputation for Nonresponse in Surveys.* New York: John Wiley & Sons, 1987.

van Buuren, S., Groothuis-Oudshoorn, K. *MICE: Multivariate Imputation by Chained Equations in R.*

See Also

mice, mids

Examples

```
###### In a descriptive context : data set : popmis

# Do multiple imputation
data(popmis)
MYpopmis <- popmis[, c("popular", "sex", "texp", "teachpop")]
IM <- mice(MYpopmis, method=c("pmm", "", "", ""), seed=13, maxit=5, m=5)

# Describe popular

# Construction of the completed imputed data sets
temp1 <- complete(IM, action=1)
temp2 <- complete(IM, action=2)
temp3 <- complete(IM, action=3)
temp4 <- complete(IM, action=4)
temp5 <- complete(IM, action=5)
```

```
# Description of the popularity score
library(mitools)
miData <- imputationList(list(temp1, temp2, temp3, temp4, temp5))
MEAN <- with(miData, mean(popular))
VAR <- with(miData, var(popular))
MIcombine(MEAN, VAR)

# Imputation with a supplementary parameter -0.5 on the popular variable
IMHyp0.5 <- sens.mice(IM, ListMethod = c("MyFunc", "", "", ""), SupPar = c(-0.5))

# Describe popular

# Construction of the completed imputed data sets
temp1Hyp0.5 <- complete(IMHyp0.5, action=1)
temp2Hyp0.5 <- complete(IMHyp0.5, action=2)
temp3Hyp0.5 <- complete(IMHyp0.5, action=3)
temp4Hyp0.5 <- complete(IMHyp0.5, action=4)
temp5Hyp0.5 <- complete(IMHyp0.5, action=5)

# Description of the popularity score
library(mitools)
miDataHyp0.5 <- imputationList(list(temp1Hyp0.5, temp2Hyp0.5, temp3Hyp0.5,
temp4Hyp0.5, temp5Hyp0.5))
MEANHyp0.5 <- with(miDataHyp0.5, mean(popular))
VARHyp0.5 <- with(miDataHyp0.5, var(popular))
MIcombine(MEANHyp0.5, VARHyp0.5)

###### In a causal context : data set CHAIN

library(mi)
data(CHAIN)

MyData <- CHAIN[apply(CHAIN, 1, function(x) !all(is.na(x))), ]

### VL2cat = binary variable (<400 c/mL vs >= 400 c/mL)
MyData$VL2cat <- NA
MyData$VL2cat <- ifelse((MyData$h39b.W1 < log(400) & !is.na(MyData$h39b.W1)),
0, MyData$VL2cat)
MyData$VL2cat <- ifelse((MyData$h39b.W1 >= log(400) & !is.na(MyData$h39b.W1)),
1, MyData$VL2cat)

# Do multiple imputation

MyDataVL2cat <- MyData[, c("age.W1", "c28.W1", "pcs.W1", "mcs37.W1", "b05.W1",
"haartadhere.W1", "VL2cat")]
MyDataVL2cat$mcs37.W1 <- as.factor(MyDataVL2cat$mcs37.W1)
MyDataVL2cat$b05.W1 <- as.factor(MyDataVL2cat$b05.W1)
MyDataVL2cat$haartadhere.W1 <- as.factor(MyDataVL2cat$haartadhere.W1)
MyDataVL2cat$VL2cat <- as.factor(MyDataVL2cat$VL2cat)
MyDataVL2catMI <- mice(MyDataVL2cat, method = c("pmm", "pmm", "pmm", "logreg",
"polyreg", "polyreg", "logreg"), seed = 13)

# Construction of the completed imputed data sets
mi1VL2cat <- complete(MyDataVL2catMI, action=1)
mi2VL2cat <- complete(MyDataVL2catMI, action=2)
mi3VL2cat <- complete(MyDataVL2catMI, action=3)
```

```
mi4VL2cat <- complete(MyDataVL2catMI, action=4)
mi5VL2cat <- complete(MyDataVL2catMI, action=5)
miDataVL2cat <- imputationList(list(mi1VL2cat, mi2VL2cat, mi3VL2cat,
mi4VL2cat, mi5VL2cat))

# Description of VL2cat and fit of the model
library(mitools)
tablesVL2cat <- with(miDataVL2cat, prop.table(table(VL2cat)))

fitVL2cat <- with(miDataVL2cat, glm(mcs37.W1 ~ as.numeric(age.W1) +
as.numeric(pcs.W1) + as.factor(haartadhere.W1) + as.factor(VL2cat) +
as.numeric(c28.W1) + as.factor(b05.W1), family=binomial))

coefsVL2cat <- MIextract(fitVL2cat, fun=coef)
varsVL2cat <- MIextract(fitVL2cat, fun=vcov)
resVL2cat <- summary(MIcombine(coefsVL2cat, varsVL2cat))
devfitVL2cat <- MIextract(fitVL2cat, fun=deviance)

fitVL2catPVAL <- with(miDataVL2cat, glm(mcs37.W1 ~ as.numeric(age.W1) +
as.numeric(pcs.W1) + as.factor(haartadhere.W1) +  as.numeric(c28.W1) +
as.factor(b05.W1), family=binomial))
devfitVL2catPVAL <- MIextract(fitVL2catPVAL, fun=deviance)

pvalVL2cat <- mean(1-pchisq(unlist(devfitVL2catPVAL)-unlist(devfitVL2cat), 1))

# Imputation with a supplementary parameter 1.2 on the VL2cat variable
MyDataVL2catMI.SCEN1 <- sens.mice(MyDataVL2catMI, ListMethod = c("", "", "",
"", "", "", "MyFunc"), SupPar = c(1.2))

# Construction of the completed imputed data sets
mi1.SCEN1VL2cat <- complete(MyDataVL2catMI.SCEN1, action=1)
mi2.SCEN1VL2cat <- complete(MyDataVL2catMI.SCEN1, action=2)
mi3.SCEN1VL2cat <- complete(MyDataVL2catMI.SCEN1, action=3)
mi4.SCEN1VL2cat <- complete(MyDataVL2catMI.SCEN1, action=4)
mi5.SCEN1VL2cat <- complete(MyDataVL2catMI.SCEN1, action=5)
miData.SCEN1VL2cat <- imputationList(list(mi1.SCEN1VL2cat, mi2.SCEN1VL2cat,
mi3.SCEN1VL2cat, mi4.SCEN1VL2cat, mi5.SCEN1VL2cat))

# Description of VL2cat and fit of the model
tablesSCEN1VL2cat <- with(miData.SCEN1VL2cat, prop.table(table(VL2cat)))

fitVL2cat.SCEN1 <- with(miData.SCEN1VL2cat, glm(mcs37.W1 ~ as.numeric(age.W1) +
as.numeric(pcs.W1) + as.factor(haartadhere.W1) + as.factor(VL2cat) +
as.numeric(c28.W1) + as.factor(b05.W1), family=binomial))

coefs.SCEN1VL2cat <- MIextract(fitVL2cat.SCEN1, fun=coef)
vars.SCEN1VL2cat <- MIextract(fitVL2cat.SCEN1, fun=vcov)
res.SCEN1VL2cat <- summary(MIcombine(coefs.SCEN1VL2cat, vars.SCEN1VL2cat))
devfitVL2cat.SCEN1 <- MIextract(fitVL2cat.SCEN1, fun=deviance)

fitVL2catPVAL.SCEN1 <- with(miData.SCEN1VL2cat, glm(mcs37.W1 ~ as.numeric(age.W1) +
as.numeric(pcs.W1) + as.factor(haartadhere.W1) + as.numeric(c28.W1) +
as.factor(b05.W1), family=binomial))

devfitVL2catPVAL.SCEN1 <- MIextract(fitVL2catPVAL.SCEN1, fun=deviance)

pvalVL2cat.SCEN1 <- mean(1-pchisq(unlist(devfitVL2catPVAL.SCEN1)-
```

```
unlist(devfitVL2cat.SCEN1), 1))
```

Index