Package 'jomo'

November 9, 2018

Type Package

Title Multilevel Joint Modelling Multiple Imputation

Version 2.6-5

Date 2018-11-06

Author Matteo Quartagno, James Carpenter

Maintainer Matteo Quartagno <m.quartagno@ucl.ac.uk>

Description Similarly to Schafer's package 'pan', 'jomo' is a package for multilevel joint modelling multiple imputation (Carpenter and Kenward, 2013) <doi: 10.1002/9781119942283>. Novel aspects of 'jomo' are the possibility of handling binary and categorical data through latent normal variables, the option to use cluster-specific covariance matrices and to impute compatibly with the substantive model.

License GPL-2

LazyData yes

Suggests mitml

Imports stats, lme4, survival, MASS, ordinal

NeedsCompilation yes

Repository CRAN

Date/Publication 2018-11-09 14:30:17 UTC

R topics documented:

---

cldata	*A simulated clustered dataset*

Description

A simulated dataset to test functions for imputation of clustered data.

Usage

```
data(cldata)
```

Format

A data frame with 1000 observations on the following 6 variables.

age A numeric variable with (centered) age. Fully observed.

measure A numeric variable with some measure of interest (unspecified). This is partially observed.

sex A binary variable with gender indicator. Fully observed.

social A 4-category variable with some social status indicator. This is partially observed.

city The cluster indicator vector. 10 cities are indexed 0 to 9.

id The id for individuals within each city.

Details

These are not real data, they are simulated to illustrate the use of the main functions of the package.

ExamScores	*Exam results for six inner London Education Authorities*

Description

A partially observed version of the tutorial dataset in package R2MLwiN.It includes examination results from six inner London Education Authorities (school boards).

Usage

```
data(cldata)
```

Format

A data frame with 4059 observations on the following 6 variables.

school A school identifier.

student A student ID.

normexam Students' exam score at age 16, normalised and partially observed.

sex Sex of pupil; a factor with levels boy, girl.

cons A column of 1s. Useful to add an intercept to th eimputation model.

standlrt Students' score at age 11 on the London Reading Test (LRT), standardised.

schgend Schools' gender; a factor with levels corresponding to mixed school (mixedsch), boys' school (boysch), and girls' school (girlsch).

avslrt Average LRT score in school.

schav Average LRT score in school, coded into 3 categories: low = bottom 25%, mid = middle 50%, high = top 25%.

vrband Students' score in test of verbal reasoning at age 11, a factor with 3 levels: vb1 = top 25%, vb2 = middle 50%, vb3 = bottom 25%.

Details

These fully observed verison of the data is available with package R2MLwiN.

Source

Browne, W. J. (2012) MCMC Estimation in MLwiN Version 2.26. University of Bristol: Centre for Multilevel Modelling.

Goldstein, H., Rasbash, J., Yang, M., Woodhouse, G., Pan, H., Nuttall, D., Thomas, S. (1993) A multilevel analysis of school examination results. Oxford Review of Education, 19, 425-433.

Rasbash, J., Charlton, C., Browne, W.J., Healy, M. and Cameron, B. (2009) MLwiN Version 2.1. Centre for Multilevel Modelling, University of Bristol.

---

jomo *Joint Modelling Imputation*

Description

A wrapper function linking all the functions for JM imputation. The matrix of responses Y, must be a data.frame where continuous variables are numeric and binary/categorical variables are factors.

Usage

```
jomo(Y, Y2=NULL, X=NULL, X2=NULL, Z=NULL, clus=NULL, beta.start=NULL,
     l2.beta.start=NULL, u.start=NULL, l1cov.start=NULL, l2cov.start=NULL,
     l1cov.prior=NULL, l2cov.prior=NULL, nburn=1000, nbetween=1000, nimp=5,
     a=NULL, a.prior=NULL, meth="common", output=1, out.iter=10)
```

Arguments

Y	A data.frame containing the (level-1) outcomes of the imputation model. Columns related to continuous variables have to be numeric and columns related to binary/categorical variables have to be factors.
Y2	A data.frame containing the level-2 outcomes of the imputation model. Columns related to continuous variables have to be numeric and columns related to binary/categorical variables have to be factors.
X	A data frame, or matrix, with covariates of the joint imputation model. Rows correspond to different observations, while columns are different variables. Missing values are not allowed in these variables. In case we want an intercept, a column of 1 is needed. The default is a column of 1.
X2	A data frame, or matrix, with level-2 covariates of the joint imputation model. Rows correspond to different level-1 observations, while columns are different variables. Missing values are not allowed in these variables. In case we want an intercept, a column of 1 is needed. The default is a column of 1.
Z	A data frame, or matrix, for covariates associated to random effects in the joint imputation model. Rows correspond to different observations, while columns are different variables. Missing values are not allowed in these variables. In case we want an intercept, a column of 1 is needed. The default is a column of 1.
clus	A data frame, or matrix, containing the cluster indicator for each observation. If missing, functions for single level imputation are automatically used.
beta.start	Starting value for beta, the vector(s) of level-1 fixed effects. Rows index different covariates and columns index different outcomes. For each n-category variable we have a fixed effect parameter for each of the n-1 latent normals. The default is a matrix of zeros.
l2.beta.start	Starting value for beta2, the vector(s) of level-2 fixed effects. Rows index different covariates and columns index different level-2 outcomes. For each n-category variable we have a fixed effect parameter for each of the n-1 latent normals. The default is a matrix of zeros.
u.start	A matrix where different rows are the starting values within each cluster for the random effects estimates u. The default is a matrix of zeros.
l1cov.start	Starting value for the covariance matrix. Dimension of this square matrix is equal to the number of outcomes (continuous plus latent normals) in the imputation model. The default is the identity matrix. Functions for imputation with random cluster-specific covariance matrices are an exception, because we need to pass the starting values for all of the matrices stacked one above the other.
l2cov.start	Starting value for the level 2 covariance matrix. Dimension of this square matrix is equal to the number of outcomes (continuous plus latent normals) in the imputation model times the number of random effects plus the number of level-2 outcomes. The default is an identity matrix.
l1cov.prior	Scale matrix for the inverse-Wishart prior for the covariance matrix. The default is the identity matrix.
l2cov.prior	Scale matrix for the inverse-Wishart prior for the level 2 covariance matrix. The default is the identity matrix.

nburn	Number of burn in iterations. Default is 1000.
nbetween	Number of iterations between two successive imputations. Default is 1000.
nimp	Number of Imputations. Default is 5.
a	Starting value for the degrees of freedom of the inverse Wishart distribution of the cluster-specific covariance matrices. Default is 50+D, with D being the dimension of the covariance matrices. This is used only with clustered data and when option meth is set to "random".
a.prior	Hyperparameter (Degrees of freedom) of the chi square prior distribution for the degrees of freedom of the inverse Wishart distribution for the cluster-specific covariance matrices. Default is D, with D being the dimension of the covariance matrices.
meth	Method used to deal with level 1 covariance matrix. When set to "common", a common matrix across clusters is used (functions jomo1rancon, jomo1rancat and jomo1ranmix). When set to "fixed", fixed study-specific matrices are considered (jomo1ranconhr, jomo1rancathr and jomo1ranmixhr with coption meth="fixed"). Finally, when set to "random", random study-specific matrices are considered (jomo1ranconhr, jomo1rancathr and jomo1ranmixhr with coption meth="random")
output	When set to any value different from 1 (default), no output is shown on screen at the end of the process.
out.iter	When set to K, every K iterations a dot is printed on screen. Default is 10.

Details

This is just a wrapper function to link all the functions in the package. Format of the columns of Y is crucial in order for the function to be using the right sub-function.

Value

On screen, the posterior mean of the fixed and random effects estimates and of the covariance matrices are shown. The only argument returned is the imputed dataset in long format. Column "Imputation" indexes the imputations. Imputation number 0 are the original data.

References

Carpenter J.R., Kenward M.G., (2013), Multiple Imputation and its Application. Wiley, ISBN: 978-0-470-74052-1.

Examples

```
# define all the inputs:

Y<-cldata[,c("measure","age")]
clus<-cldata[,c("city")]
nburn=as.integer(200);
nbetween=as.integer(200);
nimp=as.integer(5);
```

```
  #And finally we run the imputation function:
  imp<-jomo(Y,clus=clus,nburn=nburn,nbetween=nbetween,nimp=nimp)

  # Finally we show how to fit the model and combine estimate with Rubin's rules
  # Here we use mitml, other options are available in mice, mitools, etc etc

 #if (requireNamespace("mitml", quietly = TRUE)&requireNamespace("lme4", quietly = TRUE)) {
    #imp.mitml<-jomo2mitml.list(imp)
    #fit.i<-with(imp.mitml, lmer(measure~age+(1|clus)))
    #fit.MI<-testEstimates(fit.i, var.comp=T)
 # }

 #we could even run imputation with fixed or random cluster-specific covariance matrices:

 #imp<-jomo(Y,clus=clus,nburn=nburn,nbetween=nbetween,nimp=nimp, meth="fixed")
 #or:
 #imp<-jomo(Y,clus=clus,nburn=nburn,nbetween=nbetween,nimp=nimp, meth="random")

 #if we do not add clus as imput, functions for single level imputation are used:

 #imp<-jomo(Y)
```

---

jomo.clmm   *Joint Modelling Imputation Compatible with Cumulative Link Mixed Model*

Description

A function for substantive model compatible JM imputation, when the substantive model of interest is a cumulative link mixed model. Interactions and polynomial functions of the covariates are allowed. Data must be passed as a data.frame where continuous variables are numeric and binary/categorical variables are factors.

Usage

```
jomo.clmm(formula, data, level=rep(1,ncol(data)), beta.start=NULL,
          l2.beta.start=NULL, u.start=NULL, l1cov.start=NULL,
          l2cov.start=NULL, l1cov.prior=NULL, l2cov.prior=NULL,
          a.start=NULL, a.prior=NULL, nburn=1000, nbetween=1000,
          nimp=5, meth="common", output=1, out.iter=10)
```

Arguments

formula	an object of class formula: a symbolic description of the model to be fitted. It is possible to include in this formula interactions (through symbols '*' and '
data	A data.frame containing all the variables to include in the imputation model. Columns related to continuous variables have to be numeric and columns related to binary/categorical variables have to be factors.
level	A vector, indicating whether each variable is either a level 1 or a level 2 variable. The value assigned to the cluster indicator is irrelevant.
beta.start	Starting value for beta, the vector(s) of level-1 fixed effects for the joint model for the covariates. For each n-category variable we have a fixed effect parameter for each of the n-1 latent normals. The default is a matrix of zeros.
l2.beta.start	Starting value for beta2, the vector(s) of level-2 fixed effects for the joint model for the covariates. For each n-category variable we have a fixed effect parameter for each of the n-1 latent normals. The default is a matrix of zeros.
u.start	A matrix where different rows are the starting values within each cluster of the random effects estimates u for the joint model for the covariates. The default is a matrix of zeros.
l1cov.start	Starting value of the level-1 covariance matrix of the joint model for the covariates. Dimension of this square matrix is equal to the number of covariates (continuous plus latent normals) in the imputation model. The default is the identity matrix. Functions for imputation with random cluster-specific covariance matrices are an exception, because we need to pass the starting values for all of the matrices stacked one above the other.
l2cov.start	Starting value for the level 2 covariance matrix of the joint model for the covariates. Dimension of this square matrix is equal to the number of level-1 covariates (continuous plus latent normals) in the analysis model times the number of random effects plus the number of level-2 covariates. The default is an identity matrix.
l1cov.prior	Scale matrix for the inverse-Wishart prior for the covariance matrix. The default is the identity matrix.
l2cov.prior	Scale matrix for the inverse-Wishart prior for the level 2 covariance matrix. The default is the identity matrix.
a.start	Starting value for the degrees of freedom of the inverse Wishart distribution of the cluster-specific covariance matrices. Default is 50+D, with D being the dimension of the covariance matrices. This is used only with clustered data and when option meth is set to "random".
a.prior	Hyperparameter (Degrees of freedom) of the chi square prior distribution for the degrees of freedom of the inverse Wishart distribution for the cluster-specific covariance matrices. Default is D, with D being the dimension of the covariance matrices.
meth	Method used to deal with level 1 covariance matrix. When set to "common", a common matrix across clusters is used (functions jomo1rancon, jomo1rancat and jomo1ranmix). When set to "fixed", fixed study-specific matrices are considered (jomo1ranconhr, jomo1rancathr and jomo1ranmixhr with coption meth="fixed").

	Finally, when set to "random", random study-specific matrices are considered (jomo1ranconhr, jomo1rancathr and jomo1ranmixhr with option meth="random")
nburn	Number of burn in iterations. Default is 1000.
nbetween	Number of iterations between two successive imputations. Default is 1000.
nimp	Number of Imputations. Default is 5.
output	When set to any value different from 1 (default), no output is shown on screen at the end of the process.
out.iter	When set to K, every K iterations a dot is printed on screen. Default is 10.

Details

This function allows for substantive model compatible imputation when the substantive model is a cumulative link mixed-effects model. It can deal with interactions and polynomial terms through the usual lmer syntax in the formula argument. Format of the columns of data is crucial in order for the function to deal with binary/categorical covariates appropriately in the imputation algorithm.

Value

On screen, the posterior mean of the fixed effect estimates and of the residual variance are shown. The only argument returned is the imputed dataset in long format. Column "Imputation" indexes the imputations. Imputation number 0 are the original data.

References

Carpenter J.R., Kenward M.G., (2013), Multiple Imputation and its Application. Wiley, ISBN: 978-0-470-74052-1.

Examples

```
# make sure social is a factor:

cldata<-within(cldata, social<-factor(social))

# we define the data frame with all the variables

data<-cldata[,c("measure","age", "social", "city")]

# And the formula of the substantive lm model
# social as an outcome only because it is the only ordinal variable in the dataset...

formula<-as.formula(social~age+measure+(1|city))

#And finally we run the imputation function:

imp<-jomo.clmm(formula,data, nburn=2, nbetween=2, nimp=2)

# Note we are using only 2 iterations to avoid time consuming examples,
# which go against CRAN policies. In real applications we would use
```

```
# much larger burn-ins (around 1000) and at least 5 imputations.
# Check help page for function jomo to see how to fit the model and
# combine estimates with Rubin's rules
```

---

jomo.clmm.MCMCchain   *clmm Compatible JM Imputation - A tool to check convergence of the*
                             MCMC

Description

This function is similar to the jomo.clmm function, but it returns the values of all the parameters in
the model at each step of the MCMC instead of the imputations. It is useful to check the convergence
of the MCMC sampler.

Usage

```
jomo.clmm.MCMCchain(formula, data, level=rep(1,ncol(data)),
                    beta.start=NULL, l2.beta.start=NULL, u.start=NULL,
                    l1cov.start=NULL, l2cov.start=NULL, l1cov.prior=NULL,
                    l2cov.prior=NULL, a.start=NULL, a.prior=NULL,
                    betaY.start=NULL,  covuY.start=NULL,
                    uY.start=NULL, nburn=1000, meth="common",
                    start.imp=NULL, start.imp.sub=NULL, l2.start.imp=NULL,
                    output=1, out.iter=10)
```

Arguments

formula	an object of class formula: a symbolic description of the model to be fitted. It is possible to include in this formula interactions (through symbols '*' and '
data	A data.frame containing all the variables to include in the imputation model. Columns related to continuous variables have to be numeric and columns related to binary/categorical variables have to be factors.
level	A vector, indicating whether each variable is either a level 1 or a level 2 variable. The value assigned to the cluster indicator is irrelevant.
beta.start	Starting value for beta, the vector(s) of level-1 fixed effects for the joint model for the covariates. For each n-category variable we have a fixed effect parameter for each of the n-1 latent normals. The default is a matrix of zeros.
l2.beta.start	Starting value for beta2, the vector(s) of level-2 fixed effects for the joint model for the covariates. For each n-category variable we have a fixed effect parameter for each of the n-1 latent normals. The default is a matrix of zeros.
u.start	A matrix where different rows are the starting values within each cluster of the random effects estimates u for the joint model for the covariates. The default is a matrix of zeros.

l1cov.start	Starting value of the level-1 covariance matrix of the joint model for the co-variates. Dimension of this square matrix is equal to the number of covariates (continuous plus latent normals) in the imputation model. The default is the identity matrix. Functions for imputation with random cluster-specific covariance matrices are an exception, because we need to pass the starting values for all of the matrices stacked one above the other.
l2cov.start	Starting value for the level 2 covariance matrix of the joint model for the covariates. Dimension of this square matrix is equal to the number of level-1 covariates (continuous plus latent normals) in the analysis model times the number of random effects plus the number of level-2 covariates. The default is an identity matrix.
l1cov.prior	Scale matrix for the inverse-Wishart prior for the covariance matrix. The default is the identity matrix.
l2cov.prior	Scale matrix for the inverse-Wishart prior for the level 2 covariance matrix. The default is the identity matrix.
a.start	Starting value for the degrees of freedom of the inverse Wishart distribution of the cluster-specific covariance matrices. Default is 50+D, with D being the dimension of the covariance matrices. This is used only with clustered data and when option meth is set to "random".
a.prior	Hyperparameter (Degrees of freedom) of the chi square prior distribution for the degrees of freedom of the inverse Wishart distribution for the cluster-specific covariance matrices. Default is D, with D being the dimension of the covariance matrices.
meth	Method used to deal with level 1 covariance matrix. When set to "common", a common matrix across clusters is used (functions jomo1rancon, jomo1rancat and jomo1ranmix). When set to "fixed", fixed study-specific matrices are considered (jomo1ranconhr, jomo1rancathr and jomo1ranmixhr with coption meth="fixed"). Finally, when set to "random", random study-specific matrices are considered (jomo1ranconhr, jomo1rancathr and jomo1ranmixhr with option meth="random")
betaY.start	Starting value for betaY, the vector of fixed effects for the substantive analysis model. The default is the complete records estimate.
covuY.start	Starting value for covuY, the random effects covariance matrix of the substantive analysis model. The default is the complete records estimate.
uY.start	Starting value for uY, the random effects matrix of the substantive analysis model. The default is the complete records estimate.
nburn	Number of burn in iterations. Default is 1000.
output	When set to any value different from 1 (default), no output is shown on screen at the end of the process.
out.iter	When set to K, every K iterations a dot is printed on screen. Default is 10.
start.imp	Starting value for the missing data in the covariates of the substantive model. n-level categorical variables are substituted by n-1 latent normals.
l2.start.imp	Starting value for the missing data in the level-2 covariates of the substantive model. n-level categorical variables are substituted by n-1 latent normals.
start.imp.sub	Starting value for the missing data in the outcome of the substantive model. For family="binomial", these are the values of the latent normals.

Value

A list is returned; this contains the final imputed dataset (finimp) and several 3-dimensional matrices, containing all the values drawn for each parameter at each iteration: these are fixed effect parameters of the covariates beta (collectbeta), level 1 covariance matrices (collectomega), fixed effect estimates of the substantive model and associated residual variances. If there are some categorical outcomes, a further output is included in the list, finimp.latnorm, containing the final state of the imputed dataset with the latent normal variables.

Examples

```
# make sure social is a factor:

cldata<-within(cldata, social<-factor(social))

# we define the data frame with all the variables

data<-cldata[,c("measure","age", "social", "city")]

# And the formula of the substantive lm model
# social as an outcome only because it is the only ordinal variable in the dataset...

formula<-as.formula(social~age+measure+(1|city))

#And finally we run the imputation function:

imp<-jomo.clmm.MCMCchain(formula,data, nburn=100)

# We can check, for example, the convergence of the first element of beta:

# plot(c(1:100),imp$collectbeta[1,1,1:100],type="l")
```

---

jomo.coxph	*Joint Modelling Imputation Compatible with Cox Proportional Hazards Model*

Description

A function for substantive model compatible JM imputation, when the substantive model of interest is a Cox Proportional Hazards Model. Interactions and polynomial functions of the covariates are allowed. Data must be passed as a data.frame where continuous variables are numeric and binary/categorical variables are factors.

Usage

```
jomo.coxph(formula, data,  beta.start=NULL, l1cov.start=NULL, l1cov.prior=NULL,
        nburn=1000, nbetween=1000, nimp=5, output=1, out.iter=10)
```

Arguments

formula	an object of class formula: a symbolic description of the model to be fitted. It is possible to include in this formula interactions (through symbols '*' and '
data	A data.frame containing all the variables to include in the imputation model. Columns related to continuous variables have to be numeric and columns related to binary/categorical variables have to be factors.
beta.start	Starting value for beta, the vector(s) of fixed effects for the joint model for the covariates. For each n-category variable we have a fixed effect parameter for each of the n-1 latent normals. The default is a matrix of zeros.
l1cov.start	Starting value of the level-1 covariance matrix of the joint model for the co-variates. Dimension of this square matrix is equal to the number of covariates (continuous plus latent normals) in the imputation model. The default is the identity matrix.
l1cov.prior	Scale matrix for the inverse-Wishart prior for the covariance matrix. The default is the identity matrix.
nburn	Number of burn in iterations. Default is 1000.
nbetween	Number of iterations between two successive imputations. Default is 1000.
nimp	Number of Imputations. Default is 5.
output	When set to any value different from 1 (default), no output is shown on screen at the end of the process.
out.iter	When set to K, every K iterations a dot is printed on screen. Default is 10.

Details

This function allows for substantive model compatible imputation when the substantive model is a Cox PH model. It can deal with interactions and polynomial terms through the usual lm syntax in the formula argument. Format of the columns of data is crucial in order for the function to deal with binary/categorical covariates appropriately in the imputation algorithm.

Value

On screen, the posterior mean of the fixed effect estimates and of the residual variance are shown. The only argument returned is the imputed dataset in long format. Column "Imputation" indexes the imputations. Imputation number 0 are the original data.

Examples

```
#define substantive model
formula<-as.formula(Surv(time, status) ~ measure + sex + I(measure^2))

#Run imputation
if (requireNamespace("survival", quietly = TRUE)) {
  library(survival)
  imp<-jomo.coxph(formula,surdata, nburn = 100, nbetween = 100, nimp=5)
}
  # Check help page for function jomo to see how to fit the model and
  # combine estimates with Rubin's rules
```

---

jomo.coxph.MCMCchain   *coxph Compatible JM Imputation - A tool to check convergence of the MCMC*

Description

This function is similar to the jomo.coxph function, but it returns the values of all the parameters in the model at each step of the MCMC instead of the imputations. It is useful to check the convergence of the MCMC sampler.

Usage

```
jomo.coxph.MCMCchain(formula, data, beta.start = NULL, l1cov.start = NULL,
              l1cov.prior = NULL, nburn = 1000, start.imp = NULL,
              betaY.start = NULL, output = 1, out.iter = 10)
```

Arguments

formula	an object of class formula: a symbolic description of the model to be fitted. It is possible to include in this formula interactions (through symbols '*' and '
data	A data.frame containing all the variables to include in the imputation model. Columns related to continuous variables have to be numeric and columns related to binary/categorical variables have to be factors.
beta.start	Starting value for beta, the vector(s) of fixed effects for the joint model for the covariates. For each n-category variable we have a fixed effect parameter for each of the n-1 latent normals. The default is a matrix of zeros.
l1cov.start	Starting value of the level-1 covariance matrix of the joint model for the covariates. Dimension of this square matrix is equal to the number of covariates (continuous plus latent normals) in the imputation model. The default is the identity matrix.
l1cov.prior	Scale matrix for the inverse-Wishart prior for the covariance matrix. The default is the identity matrix.
betaY.start	Starting value for betaY, the vector of fixed effects for the substantive analysis model. The default is the complete records estimate.
nburn	Number of burn in iterations. Default is 1000.
output	When set to any value different from 1 (default), no output is shown on screen at the end of the process.
out.iter	When set to K, every K iterations a dot is printed on screen. Default is 10.
start.imp	Starting value for the missing data in the covariates of the substantive model. n-level categorical variables are substituted by n-1 latent normals.

Value

A list is returned; this contains the final imputed dataset (finimp) and several 3-dimensional matrices, containing all the values drawn for each parameter at each iteration: these are fixed effect parameters of the covariates beta (collectbeta), level 1 covariance matrices (collectomega), fixed effect estimates of the substantive model. If there are some categorical outcomes, a further output is included in the list, finimp.latnorm, containing the final state of the imputed dataset with the latent normal variables.

Examples

```
    # define substantive model

    formula<-as.formula(Surv(time, status) ~ measure + sex + I(measure^2))

    #Run imputation

if (requireNamespace("survival", quietly = TRUE)) {
  library(survival)
  imp<-jomo.coxph.MCMCchain(formula,surdata, nburn = 100)
  }
```

---

jomo.glm   *Joint Modelling Imputation Compatible with glm Model*

Description

A function for substantive model compatible JM imputation, when the substantive model of interest is a simple generalized linear regression model. Interactions and polynomial functions of the covariates are allowed. Data must be passed as a data.frame where continuous variables are numeric and binary/categorical variables are factors.

Usage

```
 jomo.glm(formula, data, beta.start=NULL, l1cov.start=NULL,
       l1cov.prior=NULL,nburn=1000, nbetween=1000, nimp=5,
       output=1, out.iter=10, family="binomial")
```

Arguments

formula   an object of class formula: a symbolic description of the model to be fitted. It is possible to include in this formula interactions (through symbols '*' and '

data   A data.frame containing all the variables to include in the imputation model. Columns related to continuous variables have to be numeric and columns related to binary/categorical variables have to be factors.

beta.start	Starting value for beta, the vector(s) of fixed effects for the joint model for the covariates. For each n-category variable we have a fixed effect parameter for each of the n-1 latent normals. The default is a matrix of zeros.
l1cov.start	Starting value of the level-1 covariance matrix of the joint model for the covariates. Dimension of this square matrix is equal to the number of covariates (continuous plus latent normals) in the imputation model. The default is the identity matrix.
l1cov.prior	Scale matrix for the inverse-Wishart prior for the covariance matrix. The default is the identity matrix.
nburn	Number of burn in iterations. Default is 1000.
nbetween	Number of iterations between two successive imputations. Default is 1000.
nimp	Number of Imputations. Default is 5.
output	When set to any value different from 1 (default), no output is shown on screen at the end of the process.
out.iter	When set to K, every K iterations a dot is printed on screen. Default is 10.
family	One of either "gaussian"" or "binomial". For binomial family, a probit link is assumed.

Details

This function allows for substantive model compatible imputation when the substantive model is a simple linear regression model. It can deal with interactions and polynomial terms through the usual lm syntax in the formula argument. Format of the columns of data is crucial in order for the function to deal with binary/categorical covariates appropriately in the imputation algorithm.

Value

On screen, the posterior mean of the fixed effect estimates and of the residual variance are shown. The only argument returned is the imputed dataset in long format. Column "Imputation" indexes the imputations. Imputation number 0 are the original data.

References

Carpenter J.R., Kenward M.G., (2013), Multiple Imputation and its Application. Wiley, ISBN: 978-0-470-74052-1.

Examples

```
 # make sure sex is a factor:

 sldata<-within(sldata, sex<-factor(sex))

 # we define the data frame with all the variables

 data<-sldata[,c("measure","age", "sex")]

 # And the formula of the substantive lm model
```

```
# sex as an outcome only because it is the only binary variable in the dataset...

formula<-as.formula(sex~age+measure)

#And finally we run the imputation function:

imp<-jomo.glm(formula,data, nburn=10, nbetween=10, nimp=2)

# Note we are using only 10 iterations to avoid time consuming examples,
# which go against CRAN policies. In real applications we would use
# much larger burn-ins (around 1000) and at least 5 imputations.
# Check help page for function jomo to see how to fit the model and
# combine estimates with Rubin's rules
```

---

jomo.glm.MCMCchain *glm Compatible JM Imputation - A tool to check convergence of the MCMC*

Description

This function is similar to the jomo.glm function, but it returns the values of all the parameters in the model at each step of the MCMC instead of the imputations. It is useful to check the convergence of the MCMC sampler.

Usage

```
jomo.glm.MCMCchain(formula, data, beta.start=NULL, l1cov.start=NULL,
l1cov.prior=NULL, betaY.start=NULL, nburn=1000,
start.imp=NULL, start.imp.sub=NULL, output=1, out.iter=10,
family="binomial")
```

Arguments

formula	an object of class formula: a symbolic description of the model to be fitted. It is possible to include in this formula interactions (through symbols '*' and '
data	A data.frame containing all the variables to include in the imputation model. Columns related to continuous variables have to be numeric and columns related to binary/categorical variables have to be factors.
start.imp	Starting value for the imputed covariates. n-level categorical variables are substituted by n-1 latent normals.
start.imp.sub	Starting value for the imputations of the outcome. When using binomial family, this is the value of the latent normal.
beta.start	Starting value for beta, the vector(s) of fixed effects for the joint model for the covariates. For each n-category variable we have a fixed effect parameter for each of the n-1 latent normals. The default is a matrix of zeros.

l1cov.start   Starting value of the level-1 covariance matrix of the joint model for the co-
                   variates. Dimension of this square matrix is equal to the number of covariates
                   (continuous plus latent normals) in the imputation model. The default is the
                   identity matrix.

l1cov.prior   Scale matrix for the inverse-Wishart prior for the covariance matrix. The default
                   is the identity matrix.

betaY.start   Starting value for betaY, the vector of fixed effects for the substantive analysis
                   model. The default is the complete records estimate.

nburn   Number of burn in iterations. Default is 1000.

output   When set to any value different from 1 (default), no output is shown on screen
                   at the end of the process.

out.iter   When set to K, every K iterations a dot is printed on screen. Default is 10.

family   One of either "gaussian"" or "binomial". For binomial family, a probit link is
                   assumed.

Value

A list is returned; this contains the final imputed dataset (finimp) and several 3-dimensional ma-
trices, containing all the values drawn for each parameter at each iteration: these are fixed effect
parameters of the covariates beta (collectbeta), level 1 covariance matrices (collectomega), fixed
effect estimates of the substantive model and associated residual variances. If there are some cate-
gorical outcomes, a further output is included in the list, finimp.latnorm, containing the final state
of the imputed dataset with the latent normal variables.

Examples

```
# make sure sex is a factor:

sldata<-within(sldata, sex<-factor(sex))

# we define the data frame with all the variables

data<-sldata[,c("measure","age", "sex")]

# And the formula of the substantive lm model
# sex as an outcome only because it is the only binary variable in the dataset...

formula<-as.formula(sex~age+measure)

#And finally we run the imputation function:

imp<-jomo.glm.MCMCchain(formula,data, nburn=10)

# Note we are using only 10 iterations to avoid time consuming examples,
# which go against CRAN policies. In real applications we would use
# much larger burn-ins (around 1000, to say the least).

# We can check, for example, the convergence of the first element of beta:
```

```
plot(c(1:10),imp$collectbeta[1,1,1:10],type="l")
```

---

jomo.glmer	*Joint Modelling Imputation Compatible with Generalized Linear Mixed Model*

Description

A function for substantive model compatible JM imputation, when the substantive model of interest is a generalized linear mixed-effects regression model. Interactions and polynomial functions of the covariates are allowed. Data must be passed as a data.frame where continuous variables are numeric and binary/categorical variables are factors.

Usage

```
jomo.glmer(formula, data, level=rep(1,ncol(data)), beta.start=NULL,
          l2.beta.start=NULL, u.start=NULL, l1cov.start=NULL,
          l2cov.start=NULL, l1cov.prior=NULL, l2cov.prior=NULL,
          a.start=NULL, a.prior=NULL, nburn=1000, nbetween=1000,
          nimp=5, meth="common", output=1, out.iter=10,
          family="binomial")
```

Arguments

formula	an object of class formula: a symbolic description of the model to be fitted. It is possible to include in this formula interactions (through symbols '*' and '
data	A data.frame containing all the variables to include in the imputation model. Columns related to continuous variables have to be numeric and columns related to binary/categorical variables have to be factors.
level	A vector, indicating whether each variable is either a level 1 or a level 2 variable. The value assigned to the cluster indicator is irrelevant.
beta.start	Starting value for beta, the vector(s) of level-1 fixed effects for the joint model for the covariates. For each n-category variable we have a fixed effect parameter for each of the n-1 latent normals. The default is a matrix of zeros.
l2.beta.start	Starting value for beta2, the vector(s) of level-2 fixed effects for the joint model for the covariates. For each n-category variable we have a fixed effect parameter for each of the n-1 latent normals. The default is a matrix of zeros.
u.start	A matrix where different rows are the starting values within each cluster of the random effects estimates u for the joint model for the covariates. The default is a matrix of zeros.

l1cov.start   Starting value of the level-1 covariance matrix of the joint model for the co-
                   variates. Dimension of this square matrix is equal to the number of covariates
                   (continuous plus latent normals) in the imputation model. The default is the
                   identity matrix. Functions for imputation with random cluster-specific covari-
                   ance matrices are an exception, because we need to pass the starting values for
                   all of the matrices stacked one above the other.

l2cov.start   Starting value for the level 2 covariance matrix of the joint model for the covari-
                   ates. Dimension of this square matrix is equal to the number of level-1 covariates
                   (continuous plus latent normals) in the analysis model times the number of ran-
                   dom effects plus the number of level-2 covariates. The default is an identity
                   matrix.

l1cov.prior   Scale matrix for the inverse-Wishart prior for the covariance matrix. The default
                   is the identity matrix.

l2cov.prior   Scale matrix for the inverse-Wishart prior for the level 2 covariance matrix. The
                   default is the identity matrix.

a.start   Starting value for the degrees of freedom of the inverse Wishart distribution
                   of the cluster-specific covariance matrices. Default is 50+D, with D being the
                   dimension of the covariance matrices. This is used only with clustered data and
                   when option meth is set to "random".

a.prior   Hyperparameter (Degrees of freedom) of the chi square prior distribution for the
                   degrees of freedom of the inverse Wishart distribution for the cluster-specific
                   covariance matrices. Default is D, with D being the dimension of the covariance
                   matrices.

meth   Method used to deal with level 1 covariance matrix. When set to "common",
                   a common matrix across clusters is used (functions jomo1rancon, jomo1rancat
                   and jomo1ranmix). When set to "fixed", fixed study-specific matrices are con-
                   sidered (jomo1ranconhr, jomo1rancathr and jomo1ranmixhr with coption meth="fixed").
                   Finally, when set to "random", random study-specific matrices are considered
                   (jomo1ranconhr, jomo1rancathr and jomo1ranmixhr with option meth="random")

nburn   Number of burn in iterations. Default is 1000.

nbetween   Number of iterations between two successive imputations. Default is 1000.

nimp   Number of Imputations. Default is 5.

output   When set to any value different from 1 (default), no output is shown on screen
                   at the end of the process.

out.iter   When set to K, every K iterations a dot is printed on screen. Default is 10.

family   One of either "gaussian"" or "binomial". For binomial family, a probit link is
                   assumed.

Details

This function allows for substantive model compatible imputation when the substantive model is a
linear mixed-effects model. It can deal with interactions and polynomial terms through the usual
lmer syntax in the formula argument. Format of the columns of data is crucial in order for the
function to deal with binary/categorical covariates appropriately in the imputation algorithm.

Value

On screen, the posterior mean of the fixed effect estimates and of the residual variance are shown. The only argument returned is the imputed dataset in long format. Column "Imputation" indexes the imputations. Imputation number 0 are the original data.

References

Carpenter J.R., Kenward M.G., (2013), Multiple Imputation and its Application. Wiley, ISBN: 978-0-470-74052-1.

Examples

```
# make sure sex is a factor:

cldata<-within(cldata, sex<-factor(sex))

# we define the data frame with all the variables

data<-cldata[,c("measure","age", "sex", "city")]

# And the formula of the substantive lm model
# sex as an outcome only because it is the only binary variable in the dataset...

formula<-as.formula(sex~age+measure+(1|city))

#And finally we run the imputation function:

imp<-jomo.glmer(formula,data, nburn=2, nbetween=2, nimp=2)

# Note we are using only 2 iterations to avoid time consuming examples,
# which go against CRAN policies. In real applications we would use
# much larger burn-ins (around 1000) and at least 5 imputations.
# Check help page for function jomo to see how to fit the model and
# combine estimates with Rubin's rules
```

---

jomo.glmer.MCMCchain   *glmer Compatible JM Imputation - A tool to check convergence of the MCMC*

Description

This function is similar to the jomo.glmer function, but it returns the values of all the parameters in the model at each step of the MCMC instead of the imputations. It is useful to check the convergence of the MCMC sampler.

Usage

```
jomo.glmer.MCMCchain(formula, data, level=rep(1,ncol(data)),
                   beta.start=NULL, l2.beta.start=NULL, u.start=NULL,
                   l1cov.start=NULL, l2cov.start=NULL, l1cov.prior=NULL,
                   l2cov.prior=NULL, a.start=NULL, a.prior=NULL,
                   betaY.start=NULL, covuY.start=NULL,
                   uY.start=NULL, nburn=1000, meth="common",
                   start.imp=NULL, start.imp.sub=NULL, l2.start.imp=NULL,
                   output=1, out.iter=10, family="binomial")
```

Arguments

formula	an object of class formula: a symbolic description of the model to be fitted. It is possible to include in this formula interactions (through symbols '*' and '
data	A data.frame containing all the variables to include in the imputation model. Columns related to continuous variables have to be numeric and columns related to binary/categorical variables have to be factors.
level	A vector, indicating whether each variable is either a level 1 or a level 2 variable. The value assigned to the cluster indicator is irrelevant.
beta.start	Starting value for beta, the vector(s) of level-1 fixed effects for the joint model for the covariates. For each n-category variable we have a fixed effect parameter for each of the n-1 latent normals. The default is a matrix of zeros.
l2.beta.start	Starting value for beta2, the vector(s) of level-2 fixed effects for the joint model for the covariates. For each n-category variable we have a fixed effect parameter for each of the n-1 latent normals. The default is a matrix of zeros.
u.start	A matrix where different rows are the starting values within each cluster of the random effects estimates u for the joint model for the covariates. The default is a matrix of zeros.
l1cov.start	Starting value of the level-1 covariance matrix of the joint model for the covariates. Dimension of this square matrix is equal to the number of covariates (continuous plus latent normals) in the imputation model. The default is the identity matrix. Functions for imputation with random cluster-specific covariance matrices are an exception, because we need to pass the starting values for all of the matrices stacked one above the other.
l2cov.start	Starting value for the level 2 covariance matrix of the joint model for the covariates. Dimension of this square matrix is equal to the number of level-1 covariates (continuous plus latent normals) in the analysis model times the number of random effects plus the number of level-2 covariates. The default is an identity matrix.
l1cov.prior	Scale matrix for the inverse-Wishart prior for the covariance matrix. The default is the identity matrix.
l2cov.prior	Scale matrix for the inverse-Wishart prior for the level 2 covariance matrix. The default is the identity matrix.

a.start   Starting value for the degrees of freedom of the inverse Wishart distribution of the cluster-specific covariance matrices. Default is 50+D, with D being the dimension of the covariance matrices. This is used only with clustered data and when option meth is set to "random".

a.prior   Hyperparameter (Degrees of freedom) of the chi square prior distribution for the degrees of freedom of the inverse Wishart distribution for the cluster-specific covariance matrices. Default is D, with D being the dimension of the covariance matrices.

meth   Method used to deal with level 1 covariance matrix. When set to "common", a common matrix across clusters is used (functions jomo1rancon, jomo1rancat and jomo1ranmix). When set to "fixed", fixed study-specific matrices are considered (jomo1ranconhr, jomo1rancathr and jomo1ranmixhr with coption meth="fixed"). Finally, when set to "random", random study-specific matrices are considered (jomo1ranconhr, jomo1rancathr and jomo1ranmixhr with option meth="random")

betaY.start   Starting value for betaY, the vector of fixed effects for the substantive analysis model. The default is the complete records estimate.

covuY.start   Starting value for covuY, the random effects covariance matrix of the substantive analysis model. The default is the complete records estimate.

uY.start   Starting value for uY, the random effects matrix of the substantive analysis model. The default is the complete records estimate.

nburn   Number of burn in iterations. Default is 1000.

output   When set to any value different from 1 (default), no output is shown on screen at the end of the process.

out.iter   When set to K, every K iterations a dot is printed on screen. Default is 10.

start.imp   Starting value for the missing data in the covariates of the substantive model. n-level categorical variables are substituted by n-1 latent normals.

l2.start.imp   Starting value for the missing data in the level-2 covariates of the substantive model. n-level categorical variables are substituted by n-1 latent normals.

start.imp.sub Starting value for the missing data in the outcome of the substantive model. For family="binomial", these are the values of the latent normals.

family   One of either "gaussian"" or "binomial". For binomial family, a probit link is assumed.

Value

A list is returned; this contains the final imputed dataset (finimp) and several 3-dimensional matrices, containing all the values drawn for each parameter at each iteration: these are fixed effect parameters of the covariates beta (collectbeta), level 1 covariance matrices (collectomega), fixed effect estimates of the substantive model and associated residual variances. If there are some categorical outcomes, a further output is included in the list, finimp.latnorm, containing the final state of the imputed dataset with the latent normal variables.

Examples

```
# make sure sex is a factor:

cldata<-within(cldata, sex<-factor(sex))

# we define the data frame with all the variables

data<-cldata[,c("measure","age", "sex", "city")]

# And the formula of the substantive lm model
# sex as an outcome only because it is the only binary variable in the dataset...

formula<-as.formula(sex~age+measure+(1|city))

#And finally we run the imputation function:

imp<-jomo.glmer.MCMCchain(formula,data, nburn=100)

# We can check, for example, the convergence of the first element of beta:

# plot(c(1:100),imp$collectbeta[1,1,1:100],type="l")
```

---

jomo.lm   *Joint Modelling Imputation Compatible with Linear Regression Model*

Description

A function for substantive model compatible JM imputation, when the substantive model of inter-est is a simple linear regression model. Interactions and polynomial functions of the covariates are allowed. Data must be passed as a data.frame where continuous variables are numeric and binary/categorical variables are factors.

Usage

```
jomo.lm(formula, data,  beta.start=NULL, l1cov.start=NULL,
        l1cov.prior=NULL, nburn=1000, nbetween=1000, nimp=5,
        output=1, out.iter=10)
```

Arguments

formula	an object of class formula: a symbolic description of the model to be fitted. It is possible to include in this formula interactions (through symbols '*' and '
data	A data.frame containing all the variables to include in the imputation model. Columns related to continuous variables have to be numeric and columns related to binary/categorical variables have to be factors.

beta.start	Starting value for beta, the vector(s) of fixed effects for the joint model for the covariates. For each n-category variable we have a fixed effect parameter for each of the n-1 latent normals. The default is a matrix of zeros.
l1cov.start	Starting value of the level-1 covariance matrix of the joint model for the covariates. Dimension of this square matrix is equal to the number of covariates (continuous plus latent normals) in the imputation model. The default is the identity matrix.
l1cov.prior	Scale matrix for the inverse-Wishart prior for the covariance matrix. The default is the identity matrix.
nburn	Number of burn in iterations. Default is 1000.
nbetween	Number of iterations between two successive imputations. Default is 1000.
nimp	Number of Imputations. Default is 5.
output	When set to any value different from 1 (default), no output is shown on screen at the end of the process.
out.iter	When set to K, every K iterations a dot is printed on screen. Default is 10.

Details

This function allows for substantive model compatible imputation when the substantive model is a simple linear regression model. It can deal with interactions and polynomial terms through the usual lm syntax in the formula argument. Format of the columns of data is crucial in order for the function to deal with binary/categorical covariates appropriately in the imputation algorithm.

Value

On screen, the posterior mean of the fixed effect estimates and of the residual variance are shown. The only argument returned is the imputed dataset in long format. Column "Imputation" indexes the imputations. Imputation number 0 are the original data.

References

Carpenter J.R., Kenward M.G., (2013), Multiple Imputation and its Application. Wiley, ISBN: 978-0-470-74052-1.

Examples

```
 # make sure sex is a factor:

sldata<-within(sldata, sex<-factor(sex))

# we define the data frame with all the variables

data<-sldata[,c("measure","age", "sex")]

# And the formula of the substantive lm model
```

```
formula<-as.formula(measure~sex+age+I(age^2))

#And finally we run the imputation function:

imp<-jomo.lm(formula,data, nburn=100, nbetween=100)

# Note we are using only 100 iterations to avoid time consuming examples,
# which go against CRAN policies.
# If we were interested in a model with interactions:

formula2<-as.formula(measure~sex*age)
imp2<-jomo.lm(formula2,data, nburn=100, nbetween=100)

# The analysis and combination steps are as for all the other functions
# (see e.g. help file for function jomo)
```

---

jomo.lm.MCMCchain	*lm Compatible JM Imputation - A tool to check convergence of the MCMC*

Description

This function is similar to the jomo.lm function, but it returns the values of all the parameters in the model at each step of the MCMC instead of the imputations. It is useful to check the convergence of the MCMC sampler.

Usage

```
jomo.lm.MCMCchain(formula, data, beta.start=NULL, l1cov.start=NULL,
l1cov.prior=NULL, betaY.start=NULL, varY.start=NULL, nburn=1000,
start.imp=NULL, start.imp.sub=NULL, output=1, out.iter=10)
```

Arguments

formula	an object of class formula: a symbolic description of the model to be fitted. It is possible to include in this formula interactions (through symbols '*' and '
data	A data.frame containing all the variables to include in the imputation model. Columns related to continuous variables have to be numeric and columns related to binary/categorical variables have to be factors.
beta.start	Starting value for beta, the vector(s) of fixed effects for the joint model for the covariates. For each n-category variable we have a fixed effect parameter for each of the n-1 latent normals. The default is a matrix of zeros.

l1cov.start   Starting value of the level-1 covariance matrix of the joint model for the co-variates. Dimension of this square matrix is equal to the number of covariates (continuous plus latent normals) in the imputation model. The default is the identity matrix.

l1cov.prior   Scale matrix for the inverse-Wishart prior for the covariance matrix. The default is the identity matrix.

betaY.start   Starting value for betaY, the vector of fixed effects for the substantive analysis model. The default is the complete records estimate.

varY.start   Starting value for varY, the residual variance of the substantive analysis model. The default is the complete records estimate.

nburn   Number of burn in iterations. Default is 1000.

output   When set to any value different from 1 (default), no output is shown on screen at the end of the process.

out.iter   When set to K, every K iterations a dot is printed on screen. Default is 10.

start.imp   Starting value for the missing data in the covariates of the substantive model. n-level categorical variables are substituted by n-1 latent normals.

start.imp.sub Starting value for the missing data in the outcome of the substantive model.

Value

A list is returned; this contains the final imputed dataset (finimp) and several 3-dimensional matrices, containing all the values drawn for each parameter at each iteration: these are fixed effect parameters of the covariates beta (collectbeta), level 1 covariance matrices (collectomega), fixed effect estimates of the substantive model and associated residual variances. If there are some categorical outcomes, a further output is included in the list, finimp.latnorm, containing the final state of the imputed dataset with the latent normal variables.

Examples

```
 # make sure sex is a factor:

sldata<-within(sldata, sex<-factor(sex))

# we define the data frame with all the variables

data<-sldata[,c("measure","age", "sex")]

# And the formula of the substantive lm model

formula<-as.formula(measure~sex+age+I(age^2))

#And finally we run the imputation function:

imp<-jomo.lm.MCMCchain(formula,data, nburn=100)

# Note we are using only 100 iterations to avoid time consuming examples,
```

```
# which go against CRAN policies.

# We can check, for example, the convergence of the first element of beta:

plot(c(1:100),imp$collectbeta[1,1,1:100],type="l")
```

---

jomo.lmer   *Joint Modelling Imputation Compatible with Linear Mixed-effects Re-*
                                    gression Model

Description

A function for substantive model compatible JM imputation, when the substantive model of interest
is a linear mixed-effects regression model. Interactions and polynomial functions of the covariates
are allowed. Data must be passed as a data.frame where continuous variables are numeric and
binary/categorical variables are factors.

Usage

```
jomo.lmer(formula, data, level=rep(1,ncol(data)), beta.start=NULL,
         l2.beta.start=NULL, u.start=NULL, l1cov.start=NULL, l2cov.start=NULL,
           l1cov.prior=NULL, l2cov.prior=NULL, a.start=NULL, a.prior=NULL,
         nburn=1000, nbetween=1000, nimp=5, meth="common", output=1, out.iter=10)
```

Arguments

formula	an object of class formula: a symbolic description of the model to be fitted. It is possible to include in this formula interactions (through symbols '*' and '
data	A data.frame containing all the variables to include in the imputation model. Columns related to continuous variables have to be numeric and columns related to binary/categorical variables have to be factors.
level	A vector, indicating whether each variable is either a level 1 or a level 2 variable. The value assigned to the cluster indicator is irrelevant.
beta.start	Starting value for beta, the vector(s) of level-1 fixed effects for the joint model for the covariates. For each n-category variable we have a fixed effect parameter for each of the n-1 latent normals. The default is a matrix of zeros.
l2.beta.start	Starting value for beta2, the vector(s) of level-2 fixed effects for the joint model for the covariates. For each n-category variable we have a fixed effect parameter for each of the n-1 latent normals. The default is a matrix of zeros.
u.start	A matrix where different rows are the starting values within each cluster of the random effects estimates u for the joint model for the covariates. The default is a matrix of zeros.

l1cov.start	Starting value of the level-1 covariance matrix of the joint model for the co-variates. Dimension of this square matrix is equal to the number of covariates (continuous plus latent normals) in the imputation model. The default is the identity matrix. Functions for imputation with random cluster-specific covariance matrices are an exception, because we need to pass the starting values for all of the matrices stacked one above the other.
l2cov.start	Starting value for the level 2 covariance matrix of the joint model for the covariates. Dimension of this square matrix is equal to the number of level-1 covariates (continuous plus latent normals) in the analysis model times the number of random effects plus the number of level-2 covariates. The default is an identity matrix.
l1cov.prior	Scale matrix for the inverse-Wishart prior for the covariance matrix. The default is the identity matrix.
l2cov.prior	Scale matrix for the inverse-Wishart prior for the level 2 covariance matrix. The default is the identity matrix.
a.start	Starting value for the degrees of freedom of the inverse Wishart distribution of the cluster-specific covariance matrices. Default is 50+D, with D being the dimension of the covariance matrices. This is used only with clustered data and when option meth is set to "random".
a.prior	Hyperparameter (Degrees of freedom) of the chi square prior distribution for the degrees of freedom of the inverse Wishart distribution for the cluster-specific covariance matrices. Default is D, with D being the dimension of the covariance matrices.
meth	Method used to deal with level 1 covariance matrix. When set to "common", a common matrix across clusters is used (functions jomo1rancon, jomo1rancat and jomo1ranmix). When set to "fixed", fixed study-specific matrices are considered (jomo1ranconhr, jomo1rancathr and jomo1ranmixhr with coption meth="fixed"). Finally, when set to "random", random study-specific matrices are considered (jomo1ranconhr, jomo1rancathr and jomo1ranmixhr with option meth="random")
nburn	Number of burn in iterations. Default is 1000.
nbetween	Number of iterations between two successive imputations. Default is 1000.
nimp	Number of Imputations. Default is 5.
output	When set to any value different from 1 (default), no output is shown on screen at the end of the process.
out.iter	When set to K, every K iterations a dot is printed on screen. Default is 10.

Details

This function allows for substantive model compatible imputation when the substantive model is a linear mixed-effects model. It can deal with interactions and polynomial terms through the usual lmer syntax in the formula argument. Format of the columns of data is crucial in order for the function to deal with binary/categorical covariates appropriately in the imputation algorithm.

Value

On screen, the posterior mean of the fixed effect estimates and of the residual variance are shown. The only argument returned is the imputed dataset in long format. Column "Imputation" indexes the imputations. Imputation number 0 are the original data.

References

Carpenter J.R., Kenward M.G., (2013), Multiple Imputation and its Application. Wiley, ISBN: 978-0-470-74052-1.

Examples

```
# make sure sex is a factor:

cldata<-within(cldata, sex<-factor(sex))

# we define the data frame with all the variables

data<-cldata[,c("measure","age", "sex", "city")]
mylevel<-c(1,1,1,1)

# And the formula of the substantive lm model

formula<-as.formula(measure~sex+age+I(age^2)+(1|city))

#And finally we run the imputation function:

imp<-jomo.lmer(formula,data, level=mylevel, nburn=10, nbetween=10)

# Note we are using only 10 iterations to avoid time consuming examples,
# which go against CRAN policies.
# If we were interested in a model with interactions:

# formula2<-as.formula(measure~sex*age+(1|city))
# imp2<-jomo.lmer(formula2,data, level=mylevel, nburn=10, nbetween=10)

# The analysis and combination steps are as for all the other functions
# (see e.g. help file for function jomo)
```

---

jomo.lmer.MCMCchain   *lmer Compatible JM Imputation - A tool to check convergence of the MCMC*

Description

This function is similar to the jomo.lmer function, but it returns the values of all the parameters in the model at each step of the MCMC instead of the imputations. It is useful to check the convergence of the MCMC sampler.

Usage

```
jomo.lmer.MCMCchain(formula, data, level=rep(1,ncol(data)), beta.start=NULL,
                    l2.beta.start=NULL, u.start=NULL, l1cov.start=NULL,
                    l2cov.start=NULL, l1cov.prior=NULL, l2cov.prior=NULL,
                    a.start=NULL, a.prior=NULL, betaY.start=NULL,
                    varY.start=NULL, covuY.start=NULL, uY.start=NULL,
                    nburn=1000, meth="common", start.imp=NULL,
                    start.imp.sub=NULL, l2.start.imp=NULL, output=1,
                    out.iter=10)
```

Arguments

formula	an object of class formula: a symbolic description of the model to be fitted. It is possible to include in this formula interactions (through symbols '*' and '
data	A data.frame containing all the variables to include in the imputation model. Columns related to continuous variables have to be numeric and columns related to binary/categorical variables have to be factors.
level	A vector, indicating whether each variable is either a level 1 or a level 2 variable. The value assigned to the cluster indicator is irrelevant.
beta.start	Starting value for beta, the vector(s) of level-1 fixed effects for the joint model for the covariates. For each n-category variable we have a fixed effect parameter for each of the n-1 latent normals. The default is a matrix of zeros.
l2.beta.start	Starting value for beta2, the vector(s) of level-2 fixed effects for the joint model for the covariates. For each n-category variable we have a fixed effect parameter for each of the n-1 latent normals. The default is a matrix of zeros.
u.start	A matrix where different rows are the starting values within each cluster of the random effects estimates u for the joint model for the covariates. The default is a matrix of zeros.
l1cov.start	Starting value of the level-1 covariance matrix of the joint model for the covariates. Dimension of this square matrix is equal to the number of covariates (continuous plus latent normals) in the imputation model. The default is the identity matrix. Functions for imputation with random cluster-specific covariance matrices are an exception, because we need to pass the starting values for all of the matrices stacked one above the other.
l2cov.start	Starting value for the level 2 covariance matrix of the joint model for the covariates. Dimension of this square matrix is equal to the number of level-1 covariates (continuous plus latent normals) in the analysis model times the number of random effects plus the number of level-2 covariates. The default is an identity matrix.

l1cov.prior	Scale matrix for the inverse-Wishart prior for the covariance matrix. The default is the identity matrix.
l2cov.prior	Scale matrix for the inverse-Wishart prior for the level 2 covariance matrix. The default is the identity matrix.
a.start	Starting value for the degrees of freedom of the inverse Wishart distribution of the cluster-specific covariance matrices. Default is 50+D, with D being the dimension of the covariance matrices. This is used only with clustered data and when option meth is set to "random".
a.prior	Hyperparameter (Degrees of freedom) of the chi square prior distribution for the degrees of freedom of the inverse Wishart distribution for the cluster-specific covariance matrices. Default is D, with D being the dimension of the covariance matrices.
meth	Method used to deal with level 1 covariance matrix. When set to "common", a common matrix across clusters is used (functions jomo1rancon, jomo1rancat and jomo1ranmix). When set to "fixed", fixed study-specific matrices are considered (jomo1ranconhr, jomo1rancathr and jomo1ranmixhr with coption meth="fixed"). Finally, when set to "random", random study-specific matrices are considered (jomo1ranconhr, jomo1rancathr and jomo1ranmixhr with option meth="random")
betaY.start	Starting value for betaY, the vector of fixed effects for the substantive analysis model. The default is the complete records estimate.
varY.start	Starting value for varY, the residual variance of the substantive analysis model. The default is the complete records estimate.
covuY.start	Starting value for covuY, the random effects covariance matrix of the substantive analysis model. The default is the complete records estimate.
uY.start	Starting value for uY, the random effects matrix of the substantive analysis model. The default is the complete records estimate.
nburn	Number of burn in iterations. Default is 1000.
output	When set to any value different from 1 (default), no output is shown on screen at the end of the process.
out.iter	When set to K, every K iterations a dot is printed on screen. Default is 10.
start.imp	Starting value for the missing data in the covariates of the substantive model. n-level categorical variables are substituted by n-1 latent normals.
l2.start.imp	Starting value for the missing data in the level-2 covariates of the substantive model. n-level categorical variables are substituted by n-1 latent normals.
start.imp.sub	Starting value for the missing data in the outcome of the substantive model.

Value

A list is returned; this contains the final imputed dataset (finimp) and several 3-dimensional matrices, containing all the values drawn for each parameter at each iteration: these are fixed effect parameters of the covariates beta (collectbeta), level 1 covariance matrices (collectomega), fixed effect estimates of the substantive model and associated residual variances. If there are some categorical outcomes, a further output is included in the list, finimp.latnorm, containing the final state of the imputed dataset with the latent normal variables.

Examples

```
# make sure sex is a factor:

cldata<-within(cldata, sex<-factor(sex))

# we define the data frame with all the variables

data<-cldata[,c("measure","age", "sex", "city")]
mylevel<-c(1,1,1,1)

# And the formula of the substantive lm model

formula<-as.formula(measure~sex+age+I(age^2)+(1|city))

#And finally we run the imputation function:

imp<-jomo.lmer.MCMCchain(formula,data, level=mylevel, nburn=100)

# Note we are using only 100 iterations to avoid time consuming examples,
# which go against CRAN policies.

# We can check, for example, the convergence of the first element of beta:

plot(c(1:100),imp$collectbeta[1,1,1:100],type="l")
```

---

jomo.MCMCchain   *JM Imputation - A tool to check convergence of the MCMC*

Description

This function is similar to the jomo function, but it returns the values of all the parameters in the model at each step of the MCMC instead of the imputations. It is useful to check the convergence of the MCMC sampler.

Usage

```
jomo.MCMCchain(Y, Y2=NULL, X=NULL, X2=NULL, Z=NULL, clus=NULL,
               beta.start=NULL, l2.beta.start=NULL, u.start=NULL,
               l1cov.start=NULL, l2cov.start=NULL, l1cov.prior=NULL,
             l2cov.prior=NULL, start.imp=NULL, l2.start.imp=NULL,
   nburn=1000, a=NULL, a.prior=NULL, meth="common",output=1, out.iter=10)
```

Arguments

Y	A data.frame containing the outcomes of the imputation model, i.e. the partially observed level 1 variables. Columns related to continuous variables have to be numeric and columns related to binary/categorical variables have to be factors.
Y2	A data.frame containing the level-2 outcomes of the imputation model, i.e. the partially observed level-2 variables. Columns related to continuous variables have to be numeric and columns related to binary/categorical variables have to be factors.
X	A data frame, or matrix, with covariates of the joint imputation model. Rows correspond to different observations, while columns are different variables. Missing values are not allowed in these variables. In case we want an intercept, a column of 1 is needed. The default is a column of 1.
X2	A data frame, or matrix, with level-2 covariates of the joint imputation model. Rows correspond to different level-1 observations, while columns are different variables. Missing values are not allowed in these variables. In case we want an intercept, a column of 1 is needed. The default is a column of 1.
Z	A data frame, or matrix, for covariates associated to random effects in the joint imputation model. Rows correspond to different observations, while columns are different variables. Missing values are not allowed in these variables. In case we want an intercept, a column of 1 is needed. The default is a column of 1.
clus	A data frame, or matrix, containing the cluster indicator for each observation. If missing, functions for single level imputation are automatically used.
beta.start	Starting value for beta, the vector(s) of level-1 fixed effects. Rows index different covariates and columns index different outcomes. For each n-category variable we have a fixed effect parameter for each of the n-1 latent normals. The default is a matrix of zeros.
l2.beta.start	Starting value for beta2, the vector(s) of level-2 fixed effects. Rows index different covariates and columns index different level-2 outcomes. For each n-category variable we have a fixed effect parameter for each of the n-1 latent normals. The default is a matrix of zeros.
u.start	A matrix where different rows are the starting values within each cluster for the random effects estimates u. The default is a matrix of zeros.
l1cov.start	Starting value for the covariance matrix. Dimension of this square matrix is equal to the number of outcomes (continuous plus latent normals) in the imputation model. The default is the identity matrix. Functions for imputation with random cluster-specific covariance matrices are an exception, because we need to pass the starting values for all of the matrices stacked one above the other.
l2cov.start	Starting value for the level 2 covariance matrix. Dimension of this square matrix is equal to the number of outcomes (continuous plus latent normals) in the imputation model times the number of random effects plus the number of level-2 outcomes. The default is an identity matrix.
l1cov.prior	Scale matrix for the inverse-Wishart prior for the covariance matrix. The default is the identity matrix.

l2cov.prior	Scale matrix for the inverse-Wishart prior for the level 2 covariance matrix. The default is the identity matrix.
start.imp	Starting value for the imputed dataset. n-level categorical variables are substituted by n-1 latent normals.
l2.start.imp	Starting value for the level-2 imputed variables. n-level categorical variables are substituted by n-1 latent normals.
nburn	Number of iterations. Default is 1000.
a	Starting value for the degrees of freedom of the inverse Wishart distribution of the cluster-specific covariance matrices. Default is 50+D, with D being the dimension of the covariance matrices. This is used only with clustered data and when option meth is set to "random".
a.prior	Hyperparameter (Degrees of freedom) of the chi square prior distribution for the degrees of freedom of the inverse Wishart distribution for the cluster-specific covariance matrices. Default is D, with D being the dimension of the covariance matrices.
meth	Method used to deal with level 1 covariance matrix. When set to "common", a common matrix across clusters is used (functions jomo1rancon, jomo1rancat and jomo1ranmix). When set to "fixed", fixed study-specific matrices are considered (jomo1ranconhr, jomo1rancathr and jomo1ranmixhr with coption meth="fixed"). Finally, when set to "random", random study-specific matrices are considered (jomo1ranconhr, jomo1rancathr and jomo1ranmixhr with option meth="random")
output	When set to any value different from 1 (default), no output is shown on screen at the end of the process.
out.iter	When set to K, every K iterations a dot is printed on screen. Default is 10.

Value

A list is returned; this contains the final imputed dataset (finimp) and several 3-dimensional matrices, containing all the values drawn for each parameter at each iteration: these are, potentially, fixed effect parameters beta (collectbeta), random effects (collectu), level 1 (collectomega) and level 2 covariance matrices (collectcovu) and level-2 fixed effect parameters. If there are some categorical outcomes, a further output is included in the list, finimp.latnorm, containing the final state of the imputed dataset with the latent normal variables.

Examples

```
# define all the inputs:

Y<-cldata[,c("measure","age")]
clus<-cldata[,c("city")]
nburn=as.integer(200);

#And finally we run the imputation function:
imp<-jomo.MCMCchain(Y,clus=clus,nburn=nburn)
#We can check the convergence of the first element of beta:
```

```
plot(c(1:nburn),imp$collectbeta[1,1,1:nburn],type="l")

#Or similarly we can check the convergence of any element of the level 2 covariance matrix:

plot(c(1:nburn),imp$collectcovu[1,2,1:nburn],type="l")
```

---

jomo.polr   *Joint Modelling Imputation Compatible with Proportional Odds Ordinal Probit Regression*

Description

A function for substantive model compatible JM imputation, when the substantive model of interest is a simple ordinal regression model. Interactions and polynomial functions of the covariates are allowed. Data must be passed as a data.frame where continuous variables are numeric and binary/categorical variables are factors.

Usage

```
jomo.polr(formula, data, beta.start=NULL, l1cov.start=NULL,
        l1cov.prior=NULL,nburn=1000, nbetween=1000, nimp=5,
        output=1, out.iter=10)
```

Arguments

formula	an object of class formula: a symbolic description of the model to be fitted. It is possible to include in this formula interactions (through symbols '*' and '
data	A data.frame containing all the variables to include in the imputation model. Columns related to continuous variables have to be numeric and columns related to binary/categorical variables have to be factors.
beta.start	Starting value for beta, the vector(s) of fixed effects for the joint model for the covariates. For each n-category variable we have a fixed effect parameter for each of the n-1 latent normals. The default is a matrix of zeros.
l1cov.start	Starting value of the level-1 covariance matrix of the joint model for the covariates. Dimension of this square matrix is equal to the number of covariates (continuous plus latent normals) in the imputation model. The default is the identity matrix.
l1cov.prior	Scale matrix for the inverse-Wishart prior for the covariance matrix. The default is the identity matrix.
nburn	Number of burn in iterations. Default is 1000.
nbetween	Number of iterations between two successive imputations. Default is 1000.
nimp	Number of Imputations. Default is 5.
output	When set to any value different from 1 (default), no output is shown on screen at the end of the process.
out.iter	When set to K, every K iterations a dot is printed on screen. Default is 10.

Details

This function allows for substantive model compatible imputation when the substantive model is a simple ordinal regression model. It can deal with interactions and polynomial terms through the usual lm syntax in the formula argument. Format of the columns of data is crucial in order for the function to deal with binary/categorical covariates appropriately in the imputation algorithm.

Value

On screen, the posterior mean of the fixed effect estimates and of the residual variance are shown. The only argument returned is the imputed dataset in long format. Column "Imputation" indexes the imputations. Imputation number 0 are the original data.

References

Carpenter J.R., Kenward M.G., (2013), Multiple Imputation and its Application. Wiley, ISBN: 978-0-470-74052-1.

Examples

```
 # make sure social is a factor:

sldata<-within(sldata, social<-factor(social))

# we define the data frame with all the variables

data<-sldata[,c("measure","age", "social")]

# And the formula of the substantive lm model
# social as an outcome only because it is the only binary variable in the dataset...

formula<-as.formula(social~age+measure)

#And finally we run the imputation function:

imp<-jomo.polr(formula,data, nburn=100, nbetween=100, nimp=2)

# Note we are using only 100 iterations to avoid time consuming examples,
# which go against CRAN policies. In real applications we would use
# much larger burn-ins (around 1000) and at least 5 imputations.
# Check help page for function jomo to see how to fit the model and
# combine estimates with Rubin's rules
```

---

jomo.polr.MCMCchain   *polr Compatible JM Imputation - A tool to check convergence of the MCMC*

Description

This function is similar to the jomo.polr function, but it returns the values of all the parameters in the model at each step of the MCMC instead of the imputations. It is useful to check the convergence of the MCMC sampler.

Usage

```
jomo.polr.MCMCchain(formula, data, beta.start=NULL, l1cov.start=NULL,
l1cov.prior=NULL, betaY.start=NULL, nburn=1000,
start.imp=NULL, start.imp.sub=NULL, output=1, out.iter=10)
```

Arguments

formula	an object of class formula: a symbolic description of the model to be fitted. It is possible to include in this formula interactions (through symbols '*' and '
data	A data.frame containing all the variables to include in the imputation model. Columns related to continuous variables have to be numeric and columns related to binary/categorical variables have to be factors.
start.imp	Starting value for the imputed covariates. n-level categorical variables are substituted by n-1 latent normals.
start.imp.sub	Starting value for the imputations of the outcome. When using binomial family, this is the value of the latent normal.
beta.start	Starting value for beta, the vector(s) of fixed effects for the joint model for the covariates. For each n-category variable we have a fixed effect parameter for each of the n-1 latent normals. The default is a matrix of zeros.
l1cov.start	Starting value of the level-1 covariance matrix of the joint model for the covariates. Dimension of this square matrix is equal to the number of covariates (continuous plus latent normals) in the imputation model. The default is the identity matrix.
l1cov.prior	Scale matrix for the inverse-Wishart prior for the covariance matrix. The default is the identity matrix.
betaY.start	Starting value for betaY, the vector of fixed effects for the substantive analysis model. The default is the complete records estimate.
nburn	Number of burn in iterations. Default is 1000.
output	When set to any value different from 1 (default), no output is shown on screen at the end of the process.
out.iter	When set to K, every K iterations a dot is printed on screen. Default is 10.

Value

A list is returned; this contains the final imputed dataset (finimp) and several 3-dimensional matrices, containing all the values drawn for each parameter at each iteration: these are fixed effect parameters of the covariates beta (collectbeta), level 1 covariance matrices (collectomega), fixed effect estimates of the substantive model and associated residual variances. If there are some categorical outcomes, a further output is included in the list, finimp.latnorm, containing the final state of the imputed dataset with the latent normal variables.

Examples

```
# make sure social is a factor:

sldata<-within(sldata, social<-factor(social))

# we define the data frame with all the variables

data<-sldata[,c("measure","age", "social")]

# And the formula of the substantive lm model
# social as an outcome only because it is the only ordinal variable in the dataset...

formula<-as.formula(social~age+measure)

#And finally we run the imputation function:

imp<-jomo.polr.MCMCchain(formula,data, nburn=100)

# Note we are using only 100 iterations to avoid time consuming examples,
# which go against CRAN policies. In real applications we would use
# much larger burn-ins (around 1000, to say the least).

# We can check, for example, the convergence of the first element of beta:

plot(c(1:100),imp$collectbeta[1,1,1:100],type="l")
```

---

jomo.smc   *Joint Modelling Substantive Model Compatible Imputation*

Description

A wrapper function for all the substantive model compatible JM imputation functions. The substantive model of interest is either lm, glm, polr, lmer, clmm, glmer or coxph. Interactions and polynomial functions of the covariates are allowed. Data must be passed as a data.frame where continuous variables are numeric and binary/categorical variables are factors.

Usage

```
jomo.smc(formula, data, level=rep(1,ncol(data)), beta.start=NULL,
l2.beta.start=NULL, u.start=NULL, l1cov.start=NULL, l2cov.start=NULL,
l1cov.prior=NULL, l2cov.prior=NULL, a.start=NULL, a.prior=NULL,
nburn=1000, nbetween=1000, nimp=5, meth="common", family="binomial",
output=1, out.iter=10, model)
```

Arguments

formula	an object of class formula: a symbolic description of the model to be fitted. It is possible to include in this formula interactions (through symbols '*' and '
data	A data.frame containing all the variables to include in the imputation model. Columns related to continuous variables have to be numeric and columns related to binary/categorical variables have to be factors.
level	If the dataset is multilevel, this must be a vector indicating whether each variable is either a level 1 or a level 2 variable. The value assigned to the cluster indicator is irrelevant.
beta.start	Starting value for beta, the vector(s) of level-1 fixed effects for the joint model for the covariates. For each n-category variable we have a fixed effect parameter for each of the n-1 latent normals. The default is a matrix of zeros.
l2.beta.start	Starting value for beta2, the vector(s) of level-2 fixed effects for the joint model for the covariates. For each n-category variable we have a fixed effect parameter for each of the n-1 latent normals. The default is a matrix of zeros.
u.start	A matrix where different rows are the starting values within each cluster of the random effects estimates u for the joint model for the covariates. The default is a matrix of zeros.
l1cov.start	Starting value of the level-1 covariance matrix of the joint model for the covariates. Dimension of this square matrix is equal to the number of covariates (continuous plus latent normals) in the imputation model. The default is the identity matrix. Functions for imputation with random cluster-specific covariance matrices are an exception, because we need to pass the starting values for all of the matrices stacked one above the other.
l2cov.start	Starting value for the level 2 covariance matrix of the joint model for the covariates. Dimension of this square matrix is equal to the number of level-1 covariates (continuous plus latent normals) in the analysis model times the number of random effects plus the number of level-2 covariates. The default is an identity matrix.
l1cov.prior	Scale matrix for the inverse-Wishart prior for the covariance matrix. The default is the identity matrix.
l2cov.prior	Scale matrix for the inverse-Wishart prior for the level 2 covariance matrix. The default is the identity matrix.
a.start	Starting value for the degrees of freedom of the inverse Wishart distribution of the cluster-specific covariance matrices. Default is 50+D, with D being the dimension of the covariance matrices. This is used only with clustered data and when option meth is set to "random".

a.prior	Hyperparameter (Degrees of freedom) of the chi square prior distribution for the degrees of freedom of the inverse Wishart distribution for the cluster-specific covariance matrices. Default is D, with D being the dimension of the covariance matrices.
meth	Method used to deal with level 1 covariance matrix. When set to "common", a common matrix across clusters is used (functions jomo1rancon, jomo1rancat and jomo1ranmix). When set to "fixed", fixed study-specific matrices are considered (jomo1ranconhr, jomo1rancathr and jomo1ranmixhr with coption meth="fixed"). Finally, when set to "random", random study-specific matrices are considered (jomo1ranconhr, jomo1rancathr and jomo1ranmixhr with option meth="random")
nburn	Number of burn in iterations. Default is 1000.
nbetween	Number of iterations between two successive imputations. Default is 1000.
nimp	Number of Imputations. Default is 5.
output	When set to any value different from 1 (default), no output is shown on screen at the end of the process.
out.iter	When set to K, every K iterations a dot is printed on screen. Default is 10.
model	The type of model we want to impute compatibly with. It can currently be one of lm, glm (binomial), polr, coxph, lmer, clmm or glmer (binomial).
family	One of either "gaussian"" or "binomial". For binomial family, a probit link is assumed.

Details

This function allows for substantive model compatible imputation. It can deal with interactions and polynomial terms through the usual lmer syntax in the formula argument. Format of the columns of data is crucial in order for the function to deal with binary/categorical covariates appropriately in the imputation algorithm.

Value

On screen, the posterior mean of the fixed effect estimates and of the residual variance are shown. The only argument returned is the imputed dataset in long format. Column "Imputation" indexes the imputations. Imputation number 0 are the original data.

References

Carpenter J.R., Kenward M.G., (2013), Multiple Imputation and its Application. Wiley, ISBN: 978-0-470-74052-1.

Examples

```
# make sure sex is a factor:

cldata<-within(cldata, sex<-factor(sex))

# we define the data frame with all the variables
```

```
data<-cldata[,c("measure","age", "sex", "city")]
mylevel<-c(1,1,1,1)

# And the formula of the substantive lm model

formula<-as.formula(measure~sex+age+I(age^2)+(1|city))

#And finally we run the imputation function:

imp<-jomo.smc(formula,data, level=mylevel, nburn=100, nbetween=100, model="lmer")

# Note we are using only 100 iterations to avoid time consuming examples,
# which go against CRAN policies.
# If we were interested in a model with interactions:

# formula2<-as.formula(measure~sex*age+(1|city))
# imp2<-jomo.smc(formula2,data, level=mylevel, nburn=100, nbetween=100, model="lmer")

# The analysis and combination steps are as for all the other functions
# (see e.g. help file for function jomo)
```

---

jomo.smc.MCMCchain   *Substantive Model Compatible JM Imputation - A tool to check con-*
                            vergence of the MCMC

Description

This function is similar to the jomo.smc function, but it returns the values of all the parameters in the
model at each step of the MCMC instead of the imputations. It is useful to check the convergence
of the MCMC sampler.

Usage

```
jomo.smc.MCMCchain(formula, data, level=rep(1,ncol(data)), beta.start=NULL,
l2.beta.start=NULL, u.start=NULL, l1cov.start=NULL, l2cov.start=NULL,
l1cov.prior=NULL, l2cov.prior=NULL, a.start=NULL, a.prior=NULL,
betaY.start=NULL, varY.start=NULL, covuY.start=NULL, uY.start=NULL,
nburn=1000,  meth="common", family="binomial",
start.imp=NULL, start.imp.sub=NULL, l2.start.imp=NULL, output=1,
out.iter=10, model)
```

Arguments

formula	an object of class formula: a symbolic description of the model to be fitted. It is possible to include in this formula interactions (through symbols '*' and '
data	A data.frame containing all the variables to include in the imputation model. Columns related to continuous variables have to be numeric and columns related to binary/categorical variables have to be factors.
level	If the dataset is multilevel, this must be a vector indicating whether each variable is either a level 1 or a level 2 variable. The value assigned to the cluster indicator is irrelevant.
beta.start	Starting value for beta, the vector(s) of level-1 fixed effects for the joint model for the covariates. For each n-category variable we have a fixed effect parameter for each of the n-1 latent normals. The default is a matrix of zeros.
l2.beta.start	Starting value for beta2, the vector(s) of level-2 fixed effects for the joint model for the covariates. For each n-category variable we have a fixed effect parameter for each of the n-1 latent normals. The default is a matrix of zeros.
u.start	A matrix where different rows are the starting values within each cluster of the random effects estimates u for the joint model for the covariates. The default is a matrix of zeros.
l1cov.start	Starting value of the level-1 covariance matrix of the joint model for the covariates. Dimension of this square matrix is equal to the number of covariates (continuous plus latent normals) in the imputation model. The default is the identity matrix. Functions for imputation with random cluster-specific covariance matrices are an exception, because we need to pass the starting values for all of the matrices stacked one above the other.
l2cov.start	Starting value for the level 2 covariance matrix of the joint model for the covariates. Dimension of this square matrix is equal to the number of level-1 covariates (continuous plus latent normals) in the analysis model times the number of random effects plus the number of level-2 covariates. The default is an identity matrix.
l1cov.prior	Scale matrix for the inverse-Wishart prior for the covariance matrix. The default is the identity matrix.
l2cov.prior	Scale matrix for the inverse-Wishart prior for the level 2 covariance matrix. The default is the identity matrix.
a.start	Starting value for the degrees of freedom of the inverse Wishart distribution of the cluster-specific covariance matrices. Default is 50+D, with D being the dimension of the covariance matrices. This is used only with clustered data and when option meth is set to "random".
a.prior	Hyperparameter (Degrees of freedom) of the chi square prior distribution for the degrees of freedom of the inverse Wishart distribution for the cluster-specific covariance matrices. Default is D, with D being the dimension of the covariance matrices.
meth	Method used to deal with level 1 covariance matrix. When set to "common", a common matrix across clusters is used (functions jomo1rancon, jomo1rancat and jomo1ranmix). When set to "fixed", fixed study-specific matrices are considered (jomo1ranconhr, jomo1rancathr and jomo1ranmixhr with coption meth="fixed").

	Finally, when set to "random", random study-specific matrices are considered (jomo1ranconhr, jomo1rancathr and jomo1ranmixhr with option meth="random")
betaY.start	Starting value for betaY, the vector of fixed effects for the substantive analysis model. The default is the complete records estimate.
varY.start	Starting value for varY, the residual variance of the substantive analysis model. The default is the complete records estimate.
covuY.start	Starting value for covuY, the random effects covariance matrix of the substantive analysis model. The default is the complete records estimate.
uY.start	Starting value for uY, the random effects matrix of the substantive analysis model. The default is the complete records estimate.
nburn	Number of burn in iterations. Default is 1000.
output	When set to any value different from 1 (default), no output is shown on screen at the end of the process.
out.iter	When set to K, every K iterations a dot is printed on screen. Default is 10.
start.imp	Starting value for the missing data in the covariates of the substantive model. n-level categorical variables are substituted by n-1 latent normals.
l2.start.imp	Starting value for the missing data in the level-2 covariates of the substantive model. n-level categorical variables are substituted by n-1 latent normals.
start.imp.sub	Starting value for the missing data in the outcome of the substantive model.
model	The type of model we want to impute compatibly with. It can currently be one of lm, glm (binomial), polr, coxph, lmer, clmm or glmer (binomial).
family	One of either "gaussian"" or "binomial". For binomial family, a probit link is assumed.

Value

A list is returned; this contains the final imputed dataset (finimp) and several 3-dimensional matrices, containing all the values drawn for each parameter at each iteration: these are fixed effect parameters of the covariates beta (collectbeta), level 1 covariance matrices (collectomega), fixed effect estimates of the substantive model and associated residual variances. If there are some categorical outcomes, a further output is included in the list, finimp.latnorm, containing the final state of the imputed dataset with the latent normal variables.

Examples

```
# make sure sex is a factor:

cldata<-within(cldata, sex<-factor(sex))

# we define the data frame with all the variables

data<-cldata[,c("measure","age", "sex", "city")]
mylevel<-c(1,1,1,1)

# And the formula of the substantive lm model
```

```
formula<-as.formula(measure~sex+age+I(age^2)+(1|city))

#And finally we run the imputation function:

imp<-jomo.smc.MCMCchain(formula,data, level=mylevel, nburn=100, model="lmer")

# Note we are using only 100 iterations to avoid time consuming examples,
# which go against CRAN policies.

# We can check, for example, the convergence of the first element of beta:

plot(c(1:100),imp$collectbeta[1,1,1:100],type="l")
```

---

jomo1   *JM Imputation of single level data*

Description

A wrapper function linking the 3 single level JM Imputation functions. The matrix of responses Y, must be a data.frame where continuous variables are numeric and binary/categorical variables are factors.

Usage

```
jomo1 (Y, X=NULL, beta.start=NULL, l1cov.start=NULL, l1cov.prior=NULL,
      nburn=100, nbetween=100, nimp=5, output=1, out.iter=10)
```

Arguments

Y	A data.frame containing the outcomes of the imputation model. Columns related to continuous variables have to be numeric and columns related to binary/categorical variables have to be factors.
X	A data frame, or matrix, with covariates of the joint imputation model. Rows correspond to different observations, while columns are different variables. Missing values are not allowed in these variables. In case we want an intercept, a column of 1 is needed. The default is a column of 1.
beta.start	Starting value for beta, the vector(s) of fixed effects. Rows index different covariates and columns index different outcomes. For each n-category variable we have a fixed effect parameter for each of the n-1 latent normals. The default is a matrix of zeros.
l1cov.start	Starting value for the covariance matrix. Dimension of this square matrix is equal to the number of outcomes (continuous plus latent normals) in the imputation model. The default is the identity matrix.
l1cov.prior	Scale matrix for the inverse-Wishart prior for the covariance matrix. The default is the identity matrix.

nburn	Number of burn in iterations. Default is 100.
nbetween	Number of iterations between two successive imputations. Default is 100.
nimp	Number of Imputations. Default is 5.
output	When set to any value different from 1 (default), no output is shown on screen at the end of the process.
out.iter	When set to K, every K iterations a dot is printed on screen. Default is 10.

Details

This is just a wrapper function to link jomo1con, jomo1cat and jomo1mix. Format of the columns of Y is crucial in order for the function to be using the right sub-function.

Value

On screen, the posterior mean of the fixed effects estimates and of the covariance matrix are shown. The only argument returned is the imputed dataset in long format. Column "Imputation" indexes the imputations. Imputation number 0 are the original data.

References

Carpenter J.R., Kenward M.G., (2013), Multiple Imputation and its Application. Chapter 3-5, Wiley, ISBN: 978-0-470-74052-1.

Examples

```
# define all the inputs:

Y<-sldata[,c("measure","age")]
nburn=as.integer(200);
nbetween=as.integer(200);
nimp=as.integer(5);

# Then we run the function:

imp<-jomo1(Y,nburn=nburn,nbetween=nbetween,nimp=nimp)

  # Check help page for function jomo to see how to fit the model and
  # combine estimates with Rubin's rules
```

---

jomo1.MCMCchain   *JM Imputation of single level data - A tool to check convergence of the MCMC*

Description

This function is similar to jomo1, but it returns the values of all the parameters in the model at each step of the MCMC instead of the imputations. It is useful to check the convergence of the MCMC sampler.

Usage

```
jomo1.MCMCchain(Y, X=NULL, beta.start=NULL, l1cov.start=NULL, l1cov.prior=NULL,
start.imp=NULL, nburn=100, output=1, out.iter=10)
```

Arguments

Y	A data.frame containing the outcomes of the imputation model. Columns related to continuous variables have to be numeric and columns related to binary/categorical variables have to be factors.
X	A data frame, or matrix, with covariates of the joint imputation model. Rows correspond to different observations, while columns are different variables. Missing values are not allowed in these variables. In case we want an intercept, a column of 1 is needed. The default is a column of 1.
beta.start	Starting value for beta, the vector(s) of fixed effects. Rows index different covariates and columns index different outcomes. For each n-category variable we have a fixed effect parameter for each of the n-1 latent normals. The default is a matrix of zeros.
l1cov.start	Starting value for the covariance matrix. Dimension of this square matrix is equal to the number of outcomes (continuous plus latent normals) in the imputation model. The default is the identity matrix.
l1cov.prior	Scale matrix for the inverse-Wishart prior for the covariance matrix. The default is the identity matrix.
start.imp	Starting value for the imputed dataset. n-level categorical variables are substituted by n-1 latent normals.
nburn	Number of iterations. Default is 100.
output	When set to any value different from 1 (default), no output is shown on screen at the end of the process.
out.iter	When set to K, every K iterations a dot is printed on screen. Default is 10.

Value

A list with three elements is returned: the final imputed dataset (finimp) and three 3-dimensional matrices, containing all the values for beta (collectbeta) and omega (collectomega). If there are some categorical outcomes, a further output is included in the list, finimp.latnorm, containing the final state of the imputed dataset with the latent normal variables.

Examples

```
# define all the inputs:
```

```
Y<-sldata[,c("measure","age")]
nburn=as.integer(200);

# Then we run the function:

imp<-jomo1.MCMCchain(Y,nburn=nburn)

#We can check the convergence of the first element of beta:

plot(c(1:nburn),imp$collectbeta[1,1,1:nburn],type="l")

#Or similarly we can check the convergence of any element of omega:

plot(c(1:nburn),imp$collectomega[1,2,1:nburn],type="l")
```

---

jomo1cat   *JM Imputation of single level data with categorical variables*

Description

Impute a single level dataset with categorical variables as outcomes. A joint multivariate model
for partially observed data is assumed and imputations are generated through the use of a Gibbs
sampler where the covariance matrix is updated with a Metropolis-Hastings step. Fully observed
categorical covariates can be included in the imputation model as covariates as well, but in that case
dummy variables have to be created first.

Usage

```
jomo1cat(Y.cat, Y.numcat, X=NULL, beta.start=NULL, l1cov.start=NULL,
l1cov.prior=NULL, nburn=100, nbetween=100, nimp=5,output=1, out.iter=10)
```

Arguments

Y.cat	A data frame, or matrix, with categorical (or binary) responses of the joint im- putation model. Rows correspond to different observations, while columns are different variables. Missing values are coded as NA.
Y.numcat	A vector with the number of categories in each categorical (or binary) variable.
X	A data frame, or matrix, with covariates of the joint imputation model. Rows correspond to different observations, while columns are different variables. Miss- ing values are not allowed in these variables. In case we want an intercept, a column of 1 is needed. The default is a column of 1.
beta.start	Starting value for beta, the vector(s) of fixed effects. Rows index different co- variates and columns index different outcomes. For each n-category variable we have a fixed effect parameter for each of the n-1 latent normals. The default is a matrix of zeros.

l1cov.start	Starting value for the covariance matrix. Dimension of this square matrix is equal to the number of outcomes (continuous plus latent normals) in the imputation model. The default is the identity matrix.
l1cov.prior	Scale matrix for the inverse-Wishart prior for the covariance matrix. The default is the identity matrix.
nburn	Number of burn in iterations. Default is 100.
nbetween	Number of iterations between two successive imputations. Default is 100.
nimp	Number of Imputations. Default is 5.
output	When set to any value different from 1 (default), no output is shown on screen at the end of the process.
out.iter	When set to K, every K iterations a dot is printed on screen. Default is 10.

Details

The Gibbs sampler algorithm used is described in detail in Chapter 5 of Carpenter and Kenward (2013). Regarding the choice of the priors, a flat prior is considered for beta and for the covariance matrix. A Metropolis Hastings step is implemented to update the covariance matrix, as described in the book. Binary or continuous covariates in the imputation model may be considered without any problem, but when considering a categorical covariate it has to be included with dummy variables (binary indicators) only.

Value

On screen, the posterior mean of the fixed effects estimates and of the covariance matrix are shown. The only argument returned is the imputed dataset in long format. Column "Imputation" indexes the imputations. Imputation number 0 are the original data.

References

Carpenter J.R., Kenward M.G., (2013), Multiple Imputation and its Application. Chapter 5, Wiley, ISBN: 978-0-470-74052-1.

Examples

```
# make sure sex is a factor:

sldata<-within(sldata, sex<-factor(sex))


# we define all the inputs:
# nimp, nburn and nbetween are smaller than they should. This is
#just because of CRAN policies on the examples.

Y.cat=sldata[,c("social"), drop=FALSE]
Y.numcat=matrix(4,1,1)
X=data.frame(rep(1,300),sldata[,c("sex")])
colnames(X)<-c("const", "sex")
beta.start<-matrix(0,2,3)
l1cov.start<-diag(1,3)
```

```
l1cov.prior=diag(1,3);
nburn=as.integer(100);
nbetween=as.integer(100);
nimp=as.integer(5);

# Finally we run the sampler:

imp<-jomo1cat(Y.cat,Y.numcat,X,beta.start,l1cov.start,l1cov.prior,nburn,nbetween,nimp)

#See one of the imputed values:

cat("Original value was missing (",imp[16,1],"), imputed value:", imp[316,1])

  # Check help page for function jomo to see how to fit the model and
  # combine estimates with Rubin's rules
```

---

jomo1cat.MCMCchain   *JM Imputation of single level data with categorical variables - A tool to check convergence of the MCMC*

Description

This function is similar to jomo1cat, but it returns the values of all the parameters in the model at each step of the MCMC instead of the imputations. It is useful to check the convergence of the MCMC sampler.

Usage

```
jomo1cat.MCMCchain(Y.cat, Y.numcat, X=NULL, beta.start=NULL,
l1cov.start=NULL, l1cov.prior=NULL, start.imp=NULL,
nburn=100, output=1, out.iter=10)
```

Arguments

Y.cat	A data frame, or matrix, with categorical (or binary) responses of the joint imputation model. Rows correspond to different observations, while columns are different variables. Missing values are coded as NA.
Y.numcat	A vector with the number of categories in each categorical (or binary) variable.
X	A data frame, or matrix, with covariates of the joint imputation model. Rows correspond to different observations, while columns are different variables. Missing values are not allowed in these variables. In case we want an intercept, a column of 1 is needed. The default is a column of 1.
beta.start	Starting value for beta, the vector(s) of fixed effects. Rows index different covariates and columns index different outcomes. For each n-category variable we have a fixed effect parameter for each of the n-1 latent normals. The default is a matrix of zeros.

l1cov.start   Starting value for the covariance matrix. Dimension of this square matrix is equal to the number of outcomes (continuous plus latent normals) in the imputation model. The default is the identity matrix.

l1cov.prior   Scale matrix for the inverse-Wishart prior for the covariance matrix. The default is the identity matrix.

start.imp   Starting value for the imputed dataset. n-level categorical variables are substituted by n-1 latent normals.

nburn   Number of iterations. Default is 100.

output   When set to any value different from 1 (default), no output is shown on screen at the end of the process.

out.iter   When set to K, every K iterations a dot is printed on screen. Default is 10.

Value

A list with four elements is returned: the final imputed dataset (finimp) and three 3-dimensional matrices, containing all the values drawn at each iteration for fixed effect parameters beta (collect-beta) and covariance matrix omega (collectomega). Finally, in finimp.latnorm, it is stored the final state of the imputed dataset with the latent normals in place of the categorical variables.

Examples

```
# make sure sex is a factor:

sldata<-within(sldata, sex<-factor(sex))

# we define all the inputs:
#  nburn is smaller than necessary. This is
#just because of CRAN policies on the examples.

Y.cat=sldata[,c("social"), drop=FALSE]
Y.numcat=matrix(4,1,1)
X=data.frame(rep(1,300),sldata[,c("sex")])
colnames(X)<-c("const", "sex")
beta.start<-matrix(0,2,3)
l1cov.start<-diag(1,3)
l1cov.prior=diag(1,3);
nburn=as.integer(100);

# Finally we run the sampler:

imp<-jomo1cat.MCMCchain(Y.cat,Y.numcat,X,beta.start,l1cov.start,l1cov.prior,nburn=nburn)

#We can check the convergence of the first element of beta:

plot(c(1:nburn),imp$collectbeta[1,1,1:nburn],type="l")
```

---

jomo1con   *JM Imputation of single level data with continuous variables only*

Description

Impute a single level dataset with continuous outcomes only. A joint multivariate model for partially observed data is assumed and imputations are generated through the use of a Gibbs sampler. Categorical covariates may be considered, but they have to be included with dummy variables.

Usage

```
jomo1con(Y, X=NULL, beta.start=NULL, l1cov.start=NULL, l1cov.prior=NULL,
nburn=100, nbetween=100, nimp=5, output=1, out.iter=10)
```

Arguments

Y	A data frame, or matrix, with responses of the joint imputation model. Rows correspond to different observations, while columns are different variables. Missing values are coded as NA.
X	A data frame, or matrix, with covariates of the joint imputation model. Rows correspond to different observations, while columns are different variables. Missing values are not allowed in these variables. In case we want an intercept, a column of 1 is needed. The default is a column of 1.
beta.start	Starting value for beta, the vector(s) of fixed effects. Rows index different covariates and columns index different outcomes. The default is a matrix of zeros.
l1cov.start	Starting value for the covariance matrix. Dimension of this square matrix is equal to the number of outcomes in the imputation model. The default is the identity matrix.
l1cov.prior	Scale matrix for the inverse-Wishart prior for the covariance matrix. The default is the identity matrix.
nburn	Number of burn in iterations. Default is 100.
nbetween	Number of iterations between two successive imputations. Default is 100.
nimp	Number of Imputations. Default is 5.
output	When set to any value different from 1 (default), no output is shown on screen at the end of the process.
out.iter	When set to K, every K iterations a dot is printed on screen. Default is 10.

Details

The Gibbs sampler algorithm used is described in detail in Chapter 3 of Carpenter and Kenward (2013). Regarding the choice of the priors, a flat prior is considered for beta, while an inverse-Wishart prior is given to the covariance matrix, with p-1 degrees of freedom, aka the minimum possible, to guarantee the greatest uncertainty. Binary or continuous covariates in the imputation model may be considered without any problem, but when considering a categorical covariate it has to be included through dummy variables (binary indicators) only.

Value

On screen, the posterior mean of the fixed effects estimates and of the covariance matrix are shown. The only argument returned is the imputed dataset in long format. Column "Imputation" indexes the imputations. Imputation number 0 are the original data.

References

Carpenter J.R., Kenward M.G., (2013), Multiple Imputation and its Application. Chapter 3, Wiley, ISBN: 978-0-470-74052-1.

Examples

```
#We define all the inputs:

Y=sldata[,c("measure", "age")]
X=data.frame(rep(1,300),sldata[,c("sex")])
colnames(X)<-c("const", "sex")
beta.start<-matrix(0,2,2)
l1cov.start<-diag(1,2)
l1cov.prior=diag(1,2);
nburn=as.integer(200);
nbetween=as.integer(200);
nimp=as.integer(5);

# Then we run he function:

imp<-jomo1con(Y,X,beta.start,l1cov.start,l1cov.prior,nburn,nbetween,nimp)

cat("Original value was missing(",imp[1,1],"), imputed value:", imp[301,1])

  # Check help page for function jomo to see how to fit the model and
  # combine estimates with Rubin's rules
```

---

jomo1con.MCMCchain	*JM Imputation of single level data with continuous variables only - A tool to check convergence of the MCMC*

Description

This function is similar to jomo1con, but it returns the values of all the parameters in the model at each step of the MCMC instead of the imputations. It is useful to check the convergence of the MCMC sampler.

Usage

```
jomo1con.MCMCchain(Y, X=NULL, beta.start=NULL, l1cov.start=NULL,
l1cov.prior=NULL, start.imp=NULL, nburn=100, output=1, out.iter=10)
```

Arguments

Y	A data frame, or matrix, with responses of the joint imputation model. Rows correspond to different observations, while columns are different variables. Missing values are coded as NA.
X	A data frame, or matrix, with covariates of the joint imputation model. Rows correspond to different observations, while columns are different variables. Missing values are not allowed in these variables. In case we want an intercept, a column of 1 is needed. The default is a column of 1.
beta.start	Starting value for beta, the vector(s) of fixed effects. Rows index different covariates and columns index different outcomes. The default is a matrix of zeros.
l1cov.start	Starting value for the covariance matrix. Dimension of this square matrix is equal to the number of outcomes in the imputation model. The default is the identity matrix.
l1cov.prior	Scale matrix for the inverse-Wishart prior for the covariance matrix. The default is the identity matrix.
start.imp	Starting value for the imputed dataset.
nburn	Number of iterations. Default is 100.
output	When set to any value different from 1 (default), no output is shown on screen at the end of the process.
out.iter	When set to K, every K iterations a dot is printed on screen. Default is 10.

Value

A list with three elements is returned: the final imputed dataset (finimp) and three 3-dimensional matrices, containing all the values for the fixed effect parameters beta (collectbeta) and the covariance matrix omega (collectomega).

Examples

```
#We define all the inputs:

Y=sldata[,c("measure", "age")]
X=data.frame(rep(1,300),sldata[,c("sex")])
colnames(X)<-c("const", "sex")

beta.start<-matrix(0,2,2)
l1cov.start<-diag(1,2)
l1cov.prior=diag(1,2);
nburn=as.integer(200);
```

```
# Then we run he function:

imp<-jomo1con.MCMCchain(Y,X,beta.start,l1cov.start,l1cov.prior,nburn=nburn)

#We can check the convergence of the first element of beta:

plot(c(1:nburn),imp$collectbeta[1,1,1:nburn],type="l")

#Or similarly we can check the convergence of any element of omega:

plot(c(1:nburn),imp$collectomega[1,2,1:nburn],type="l")
```

---

jomo1mix   *JM Imputation of single level data with mixed variable types*

Description

Impute a single level dataset with mixed data types as outcome. A joint multivariate model for partially observed data is assumed and imputations are generated through the use of a Gibbs sampler where the covariance matrix is updated with a Metropolis-Hastings step. Fully observed categorical variables may be considered as covariates as well, but they have to be included as dummy variables.

Usage

```
jomo1mix(Y.con, Y.cat, Y.numcat, X=NULL, beta.start=NULL, l1cov.start=NULL,
l1cov.prior=NULL, nburn=100, nbetween=100, nimp=5, output=1,out.iter=10)
```

Arguments

Y.con	A data frame, or matrix, with continuous responses of the joint imputation model. Rows correspond to different observations, while columns are different variables. Missing values are coded as NA. If no continuous outcomes are present in the model, jomo1cat should be used instead.
Y.cat	A data frame, or matrix, with categorical (or binary) responses of the joint imputation model. Rows correspond to different observations, while columns are different variables. Missing values are coded as NA.
Y.numcat	A vector with the number of categories in each categorical (or binary) variable.
X	A data frame, or matrix, with covariates of the joint imputation model. Rows correspond to different observations, while columns are different variables. Missing values are not allowed in these variables. In case we want an intercept, a column of 1 is needed. The default is a column of 1.
beta.start	Starting value for beta, the vector(s) of fixed effects. Rows index different covariates and columns index different outcomes. For each n-category variable we define n-1 latent normals. The default is a matrix of zeros.

l1cov.start   Starting value for the covariance matrix. Dimension of this square matrix is
                   equal to the number of outcomes (continuous plus latent normals) in the impu-
                   tation model. The default is the identity matrix.

l1cov.prior   Scale matrix for the inverse-Wishart prior for the covariance matrix. The default
                   is the identity matrix.

nburn   Number of burn in iterations. Default is 100.

nbetween   Number of iterations between two successive imputations. Default is 100.

nimp   Number of Imputations. Default is 5.

output   When set to any value different from 1 (default), no output is shown on screen
                   at the end of the process.

out.iter   When set to K, every K iterations a dot is printed on screen. Default is 10.

Details

Regarding the choice of the priors, a flat prior is considered for beta and for the covariance matrix.
A Metropolis Hastings step is implemented to update the covariance matrix, as described in the
book. Binary or continuous covariates in the imputation model may be considered without any
problem, but when considering a categorical covariate it has to be included with dummy variables
(binary indicators) only.

Value

On screen, the posterior mean of the fixed effects estimates and of the covariance matrix are shown.
The only argument returned is the imputed dataset in long format. Column "Imputation" indexes
the imputations. Imputation number 0 are the original data.

References

Carpenter J.R., Kenward M.G., (2013), Multiple Imputation and its Application. Chapter 5, Wiley,
ISBN: 978-0-470-74052-1.

Examples

```
#Then, we define all the inputs:
# nburn is smaller than needed. This is
#just because of CRAN policies on the examples.

Y.con=sldata[,c("measure","age")]
Y.cat=sldata[,c("social"), drop=FALSE]
Y.numcat=matrix(4,1,1)
X=data.frame(rep(1,300),sldata[,c("sex")])
colnames(X)<-c("const", "sex")
beta.start<-matrix(0,2,5)
l1cov.start<-diag(1,5)
l1cov.prior=diag(1,5);
nburn=as.integer(100);
nbetween=as.integer(100);
nimp=as.integer(5);
```

```
#Then we run the sampler:

imp<-jomo1mix(Y.con,Y.cat,Y.numcat,X,beta.start,l1cov.start,
      l1cov.prior,nburn,nbetween,nimp)

cat("Original value was missing(",imp[1,1],"), imputed value:", imp[301,1])

  # Check help page for function jomo to see how to fit the model and
  # combine estimates with Rubin's rules
```

---

jomo1mix.MCMCchain   *JM Imputation of single level data with mixed variable types*

Description

This function is similar to jomo1mix, but it returns the values of all the parameters in the model at each step of the MCMC instead of the imputations. It is useful to check the convergence of the MCMC sampler.

Usage

```
jomo1mix.MCMCchain(Y.con, Y.cat, Y.numcat, X=NULL, beta.start=NULL,
l1cov.start=NULL, l1cov.prior=NULL, start.imp=NULL, nburn=100,
output=1, out.iter=10)
```

Arguments

Y.con	A data frame, or matrix, with continuous responses of the joint imputation model. Rows correspond to different observations, while columns are different variables. Missing values are coded as NA. If no continuous outcomes are present in the model, jomo1cat should be used instead.
Y.cat	A data frame, or matrix, with categorical (or binary) responses of the joint imputation model. Rows correspond to different observations, while columns are different variables. Missing values are coded as NA.
Y.numcat	A vector with the number of categories in each categorical (or binary) variable.
X	A data frame, or matrix, with covariates of the joint imputation model. Rows correspond to different observations, while columns are different variables. Missing values are not allowed in these variables. In case we want an intercept, a column of 1 is needed. The default is a column of 1.
beta.start	Starting value for beta, the vector(s) of fixed effects. Rows index different covariates and columns index different outcomes. For each n-category variable we define n-1 latent normals. The default is a matrix of zeros.

l1cov.start	Starting value for the covariance matrix. Dimension of this square matrix is equal to the number of outcomes (continuous plus latent normals) in the imputation model. The default is the identity matrix.
l1cov.prior	Scale matrix for the inverse-Wishart prior for the covariance matrix. The default is the identity matrix.
start.imp	Starting value for the imputed dataset. n-level categorical variables are substituted by n-1 latent normals.
nburn	Number of iterations. Default is 100.
output	When set to any value different from 1 (default), no output is shown on screen at the end of the process.
out.iter	When set to K, every K iterations a dot is printed on screen. Default is 10.

Value

A list with four elements is returned: the final imputed dataset (finimp) and three 3-dimensional matrices, containing all the values for beta (collectbeta) and omega (collectomega). Finally, in finimp.latnorm it is stored the final state of the imputed dataset with the latent normals in place of the categorical variables.

Examples

```
#Then, we define all the inputs:
# nburn is smaller than needed. This is
#just because of CRAN policies on the examples.

Y.con=sldata[,c("measure","age")]
Y.cat=sldata[,c("social"), drop=FALSE]
Y.numcat=matrix(4,1,1)
X=data.frame(rep(1,300),sldata[,c("sex")])
colnames(X)<-c("const", "sex")
beta.start<-matrix(0,2,5)
l1cov.start<-diag(1,5)
l1cov.prior=diag(1,5);
nburn=as.integer(100);


#Then we run the sampler:

imp<-jomo1mix.MCMCchain(Y.con,Y.cat,Y.numcat,X,beta.start,l1cov.start,l1cov.prior,nburn=nburn)

#We can check the convergence of the first element of beta:

plot(c(1:nburn),imp$collectbeta[1,1,1:nburn],type="l")

#Or similarly we can check the convergence of any element of omega:

plot(c(1:nburn),imp$collectomega[1,1,1:nburn],type="l")
```

---

jomo1ran   *JM Imputation of clustered data*

Description

A wrapper function linking the six 2-level JM Imputation functions. The matrix of responses Y, must be a data.frame where continuous variables are numeric and binary/categorical variables are factors.

Usage

```
jomo1ran(Y, X=NULL, Z=NULL,clus,
      beta.start=NULL, u.start=NULL, l1cov.start=NULL, l2cov.start=NULL,
      l1cov.prior=NULL, l2cov.prior=NULL, nburn=1000, nbetween=1000, nimp=5,
      a=NULL, a.prior=NULL, meth="common", output=1, out.iter=10)
```

Arguments

Y	A data.frame containing the outcomes of the imputation model. Columns related to continuous variables have to be numeric and columns related to binary/categorical variables have to be factors.
X	A data frame, or matrix, with covariates of the joint imputation model. Rows correspond to different observations, while columns are different variables. Missing values are not allowed in these variables. In case we want an intercept, a column of 1 is needed. The default is a column of 1.
Z	A data frame, or matrix, for covariates associated to random effects in the joint imputation model. Rows correspond to different observations, while columns are different variables. Missing values are not allowed in these variables. In case we want an intercept, a column of 1 is needed. The default is a column of 1.
clus	A data frame, or matrix, containing the cluster indicator for each observation.
beta.start	Starting value for beta, the vector(s) of fixed effects. Rows index different covariates and columns index different outcomes. For each n-category variable we define n-1 latent normals. The default is a matrix of zeros.
u.start	A matrix where different rows are the starting values within each cluster for the random effects estimates u. The default is a matrix of zeros.
l1cov.start	Starting value for the covariance matrix. Dimension of this square matrix is equal to the number of outcomes (continuous plus latent normals) in the imputation model. The default is the identity matrix.
l2cov.start	Starting value for the level 2 covariance matrix. Dimension of this square matrix is equal to the number of outcomes (continuous plus latent normals) in the imputation model times the number of random effects. The default is an identity matrix.

l1cov.prior	Scale matrix for the inverse-Wishart prior for the covariance matrix. The default is the identity matrix.
l2cov.prior	Scale matrix for the inverse-Wishart prior for the level 2 covariance matrix. The default is the identity matrix.
nburn	Number of burn in iterations. Default is 1000.
nbetween	Number of iterations between two successive imputations. Default is 1000.
nimp	Number of Imputations. Default is 5.
a	Starting value for the degrees of freedom of the inverse Wishart distribution of the cluster-specific covariance matrices. Default is 50+D, with D being the dimension of the covariance matrices. This is used only when option meth is set to "random".
a.prior	Hyperparameter (Degrees of freedom) of the chi square prior distribution for the degrees of freedom of the inverse Wishart distribution for the cluster-specific covariance matrices. Default is the starting value for a.
meth	Method used to deal with level 1 covariance matrix. When set to "common", a common matrix across clusters is used (functions jomo1rancon, jomo1rancat and jomo1ranmix). When set to "fixed", fixed study-specific matrices are considered (jomo1ranconhr, jomo1rancathr and jomo1ranmixhr with coption meth="fixed"). Finally, when set to "random", random study-specific matrices are considered (jomo1ranconhr, jomo1rancathr and jomo1ranmixhr with coption meth="random")
output	When set to any value different from 1 (default), no output is shown on screen at the end of the process.
out.iter	When set to K, every K iterations a dot is printed on screen. Default is 10.

Details

This is just a wrapper function to link jomo1rancon, jomo1rancat and jomo1ranmix and the respective "hr" (heterogeneity in covariance matrices) versions. Format of the columns of Y is crucial in order for the function to be using the right sub-function.

Value

On screen, the posterior mean of the fixed effects estimates and of the covariance matrix are shown. The only argument returned is the imputed dataset in long format. Column "Imputation" indexes the imputations. Imputation number 0 are the original data.

References

Carpenter J.R., Kenward M.G., (2013), Multiple Imputation and its Application. Chapter 9, Wiley, ISBN: 978-0-470-74052-1.

Examples

```
# define all the inputs:
```

```
Y<-cldata[,c("measure","age")]
clus<-cldata[,c("city")]
nburn=as.integer(200);
nbetween=as.integer(200);
nimp=as.integer(5);


#And finally we run the imputation function:
imp<-jomo1ran(Y,clus=clus,nburn=nburn,nbetween=nbetween,nimp=nimp)

#we could even run it with fixed or random cluster-specific covariance matrices:

#imp<-jomo1ran(Y,clus=clus,nburn=nburn,nbetween=nbetween,nimp=nimp, meth="fixed")
#or:
#imp<-jomo1ran(Y,clus=clus,nburn=nburn,nbetween=nbetween,nimp=nimp, meth="random")

  # Check help page for function jomo to see how to fit the model and
  # combine estimates with Rubin's rules
```

---

jomo1ran.MCMCchain	*JM Imputation of clustered data - A tool to check convergence of the MCMC*

Description

This function is similar to jomo1ran, but it returns the values of all the parameters in the model at each step of the MCMC instead of the imputations. It is useful to check the convergence of the MCMC sampler.

Usage

```
jomo1ran.MCMCchain(Y, X=NULL, Z=NULL,clus, beta.start=NULL, u.start=NULL,
l1cov.start=NULL,l2cov.start=NULL, l1cov.prior=NULL, l2cov.prior=NULL,
start.imp=NULL, nburn=1000, a=NULL,a.prior=NULL, meth="common", output=1,
out.iter=10)
```

Arguments

Y   A data.frame containing the outcomes of the imputation model. Columns related to continuous variables have to be numeric and columns related to binary/categorical variables have to be factors.

X   A data frame, or matrix, with covariates of the joint imputation model. Rows correspond to different observations, while columns are different variables. Missing values are not allowed in these variables. In case we want an intercept, a column of 1 is needed. The default is a column of 1.

Z	A data frame, or matrix, for covariates associated to random effects in the joint imputation model. Rows correspond to different observations, while columns are different variables. Missing values are not allowed in these variables. In case we want an intercept, a column of 1 is needed. The default is a column of 1.
clus	A data frame, or matrix, containing the cluster indicator for each observation.
beta.start	Starting value for beta, the vector(s) of fixed effects. Rows index different covariates and columns index different outcomes. For each n-category variable we define n-1 latent normals. The default is a matrix of zeros.
u.start	A matrix where different rows are the starting values within each cluster for the random effects estimates u. The default is a matrix of zeros.
l1cov.start	Starting value for the covariance matrix. Dimension of this square matrix is equal to the number of outcomes (continuous plus latent normals) in the imputation model. The default is the identity matrix.
l2cov.start	Starting value for the level 2 covariance matrix. Dimension of this square matrix is equal to the number of outcomes (continuous plus latent normals) in the imputation model times the number of random effects. The default is an identity matrix.
l1cov.prior	Scale matrix for the inverse-Wishart prior for the covariance matrix. The default is the identity matrix.
l2cov.prior	Scale matrix for the inverse-Wishart prior for the level 2 covariance matrix. The default is the identity matrix.
start.imp	Starting value for the imputed dataset. n-level categorical variables are substituted by n-1 latent normals.
nburn	Number of iterations. Default is 1000.
a	Starting value for the degrees of freedom of the inverse Wishart distribution of the cluster-specific covariance matrices. Default is 50+D, with D being the dimension of the covariance matrices. This is used only when option meth is set to "random".
a.prior	Hyperparameter (Degrees of freedom) of the chi square prior distribution for the degrees of freedom of the inverse Wishart distribution for the cluster-specific covariance matrices. Default is the starting value for a.
meth	Method used to deal with level 1 covariance matrix. When set to "common", a common matrix across clusters is used (functions jomo1rancon, jomo1rancat and jomo1ranmix). When set to "fixed", fixed study-specific matrices are considered (jomo1ranconhr, jomo1rancathr and jomo1ranmixhr with coption meth="fixed"). Finally, when set to "random", random study-specific matrices are considered (jomo1ranconhr, jomo1rancathr and jomo1ranmixhr with option meth="random")
output	When set to any value different from 1 (default), no output is shown on screen at the end of the process.
out.iter	When set to K, every K iterations a dot is printed on screen. Default is 10.

Value

A list with six elements is returned: the final imputed dataset (finimp) and four 3-dimensional matrices, containing all the values for beta (collectbeta), the random effects (collectu) and the level 1 (collectomega) and level 2 covariance matrices (collectcovu). Finally, for cases where categorical variabels are present, the final state of the imputed dataset with the latent normals in place of the categorical variables is stored in finimp.latnorm.

Examples

```
  # define all the inputs:

  Y<-cldata[,c("measure","age")]
  clus<-cldata[,c("city")]
nburn=as.integer(200);

#And finally we run the imputation function:
imp<-jomo1ran.MCMCchain(Y,clus=clus,nburn=nburn)
#We can check the convergence of the first element of beta:

plot(c(1:nburn),imp$collectbeta[1,1,1:nburn],type="l")

#Or similarly we can check the convergence of any element of the level 2 covariance matrix:

plot(c(1:nburn),imp$collectcovu[1,2,1:nburn],type="l")
```

---

jomo1rancat	*JM Imputation of clustered data with categorical variables*

Description

Impute a clustered dataset with categorical variables as outcome. A joint multivariate model for partially observed data is assumed and imputations are generated through the use of a Gibbs sampler where the covariance matrix is updated with a Metropolis-Hastings step. Fully observed categorical covariates may be considered as covariates as well, but they have to be included as dummy variables.

Usage

```
jomo1rancat( Y.cat, Y.numcat, X=NULL, Z=NULL, clus, beta.start=NULL,
u.start=NULL, l1cov.start=NULL, l2cov.start=NULL, l1cov.prior=NULL,
l2cov.prior=NULL, nburn=1000, nbetween=1000, nimp=5, output=1, out.iter=10)
```

Arguments

Y.cat	A data frame, or matrix, with categorical (or binary) responses of the joint imputation model. Rows correspond to different observations, while columns are different variables. Missing values are coded as NA.
Y.numcat	A vector with the number of categories in each categorical (or binary) variable.

X	A data frame, or matrix, with covariates of the joint imputation model. Rows correspond to different observations, while columns are different variables. Missing values are not allowed in these variables. In case we want an intercept, a column of 1 is needed. The default is a column of 1.
Z	A data frame, or matrix, for covariates associated to random effects in the joint imputation model. Rows correspond to different observations, while columns are different variables. Missing values are not allowed in these variables. In case we want an intercept, a column of 1 is needed. The default is a column of 1.
clus	A data frame, or matrix, containing the cluster indicator for each observation.
beta.start	Starting value for beta, the vector(s) of fixed effects. Rows index different covariates and columns index different outcomes. For each n-category variable we define n-1 latent normals. The default is a matrix of zeros.
u.start	A matrix where different rows are the starting values within each cluster for the random effects estimates u. The default is a matrix of zeros.
l1cov.start	Starting value for the covariance matrix. Dimension of this square matrix is equal to the number of outcomes (continuous plus latent normals) in the imputation model. The default is the identity matrix.
l2cov.start	Starting value for the level 2 covariance matrix. Dimension of this square matrix is equal to the number of outcomes (continuous plus latent normals) in the imputation model times the number of random effects. The default is an identity matrix.
l1cov.prior	Scale matrix for the inverse-Wishart prior for the covariance matrix. The default is the identity matrix.
l2cov.prior	Scale matrix for the inverse-Wishart prior for the level 2 covariance matrix. The default is the identity matrix.
nburn	Number of burn in iterations. Default is 1000.
nbetween	Number of iterations between two successive imputations. Default is 1000.
nimp	Number of Imputations. Default is 5.
output	When set to any value different from 1 (default), no output is shown on screen at the end of the process.
out.iter	When set to K, every K iterations a dot is printed on screen. Default is 10.

Details

The Gibbs sampler algorithm used is described in detail in Chapter 9 of Carpenter and Kenward (2013). Regarding the choice of the priors, a flat prior is considered for beta and for the covariance matrix. A Metropolis Hastings step is implemented to update the covariance matrix, as described in the book. Binary or continuous covariates in the imputation model may be considered without any problem, but when considering a categorical covariate it has to be included with dummy variables (binary indicators) only.

Value

On screen, the posterior mean of the fixed effects estimates and of the covariance matrix are shown. The only argument returned is the imputed dataset in long format. Column "Imputation" indexes the imputations. Imputation number 0 are the original data.

References

Carpenter J.R., Kenward M.G., (2013), Multiple Imputation and its Application. Chapter 9, Wiley, ISBN: 978-0-470-74052-1.

Examples

```
#we define all the inputs:
# nimp, nburn and nbetween are smaller than they should. This is
#just because of CRAN policies on the examples.

Y.cat=cldata[,c("social"), drop=FALSE]
Y.numcat=matrix(4,1,1)
X=data.frame(rep(1,1000),cldata[,c("sex")])
colnames(X)<-c("const", "sex")
Z<-data.frame(rep(1,1000))
clus<-cldata[,c("city")]
beta.start<-matrix(0,2,3)
u.start<-matrix(0,10,3)
l1cov.start<-diag(1,3)
l2cov.start<-diag(1,3)
l1cov.prior=diag(1,3);
l2cov.prior=diag(1,3);
nburn=as.integer(100);
nbetween=as.integer(100);
nimp=as.integer(4);

#And finally we run the imputation function:

imp<-jomo1rancat(Y.cat, Y.numcat, X,Z,clus,beta.start,u.start,l1cov.start,
                l2cov.start,l1cov.prior,l2cov.prior,nburn,nbetween,nimp)

 cat("Original value was missing (",imp[3,1],"), imputed value:", imp[1003,1])

  # Check help page for function jomo to see how to fit the model and
  # combine estimates with Rubin's rules
```

---

jomo1rancat.MCMCchain *JM Imputation of clustered data with categorical variables - A tool to check convergence of the MCMC*

Description

This function is similar to jomo1rancat, but it returns the values of all the parameters in the model at each step of the MCMC instead of the imputations. It is useful to check the convergence of the MCMC sampler.

Usage

```
jomo1rancat.MCMCchain(Y.cat, Y.numcat, X=NULL, Z=NULL,clus, beta.start=NULL,
u.start=NULL, l1cov.start=NULL, l2cov.start=NULL, l1cov.prior=NULL,
l2cov.prior=NULL, start.imp=NULL,nburn=1000, output=1, out.iter=10)
```

Arguments

Y.cat	A data frame, or matrix, with categorical (or binary) responses of the joint imputation model. Rows correspond to different observations, while columns are different variables. Missing values are coded as NA.
Y.numcat	A vector with the number of categories in each categorical (or binary) variable.
X	A data frame, or matrix, with covariates of the joint imputation model. Rows correspond to different observations, while columns are different variables. Missing values are not allowed in these variables. In case we want an intercept, a column of 1 is needed. The default is a column of 1.
Z	A data frame, or matrix, for covariates associated to random effects in the joint imputation model. Rows correspond to different observations, while columns are different variables. Missing values are not allowed in these variables. In case we want an intercept, a column of 1 is needed. The default is a column of 1.
clus	A data frame, or matrix, containing the cluster indicator for each observation.
beta.start	Starting value for beta, the vector(s) of fixed effects. Rows index different covariates and columns index different outcomes. For each n-category variable we define n-1 latent normals. The default is a matrix of zeros.
u.start	A matrix where different rows are the starting values within each cluster for the random effects estimates u. The default is a matrix of zeros.
l1cov.start	Starting value for the covariance matrix. Dimension of this square matrix is equal to the number of outcomes (continuous plus latent normals) in the imputation model. The default is the identity matrix.
l2cov.start	Starting value for the level 2 covariance matrix. Dimension of this square matrix is equal to the number of outcomes (continuous plus latent normals) in the imputation model times the number of random effects. The default is an identity matrix.
l1cov.prior	Scale matrix for the inverse-Wishart prior for the covariance matrix. The default is the identity matrix.
l2cov.prior	Scale matrix for the inverse-Wishart prior for the level 2 covariance matrix. The default is the identity matrix.
start.imp	Starting value for the imputed dataset. n-level categorical variables are substituted by n-1 latent normals.
nburn	Number of burn in iterations. Default is 1000.
output	When set to any value different from 1 (default), no output is shown on screen at the end of the process.
out.iter	When set to K, every K iterations a dot is printed on screen. Default is 10.

Value

A list with six elements is returned: the final imputed dataset (finimp) and four 3-dimensional matrices, containing all the values for beta (collectbeta), the random effects (collectu) and the level 1 (collectomega) and level 2 covariance matrices (collectcovu). Finally, the final state of the imputed dataset with the latent normals in place of the categorical variables is stored in finimp.latnorm.

Examples

```
# define all the inputs:
# nburn  smaller than needed. This is
#just because of CRAN policies on the examples.

Y.cat=cldata[,c("social"), drop=FALSE]
Y.numcat=matrix(4,1,1)
X=data.frame(rep(1,1000),cldata[,c("sex")])
colnames(X)<-c("const", "sex")
Z<-data.frame(rep(1,1000))
clus<-cldata[,c("city")]
beta.start<-matrix(0,2,3)
u.start<-matrix(0,10,3)
l1cov.start<-diag(1,3)
l2cov.start<-diag(1,3)
l1cov.prior=diag(1,3);
l2cov.prior=diag(1,3);
nburn=as.integer(100);

#And finally we run the imputation function:

imp<-jomo1rancat.MCMCchain(Y.cat, Y.numcat, X,Z,clus,beta.start,u.start,l1cov.start,
l2cov.start,l1cov.prior,l2cov.prior,nburn=nburn)
#We can check the convergence of the first element of beta:

plot(c(1:nburn),imp$collectbeta[1,1,1:nburn],type="l")

#Or similarly we can check the convergence of any element of the level 2 covariance matrix:

plot(c(1:nburn),imp$collectcovu[1,2,1:nburn],type="l")
```

---

jomo1rancathr   *JM Imputation of clustered data with categorical variables with cluster-specific covariance matrices*

Description

Impute a clustered dataset with categorical variables as outcome. A joint multivariate model for partially observed data is assumed and imputations are generated through the use of a Gibbs sampler where a different covariance matrix is sampled within each cluster. Fully observed categorical covariates may be considered as covariates as well, but they have to be included as dummy variables.

Usage

```
jomo1rancathr( Y.cat, Y.numcat, X=NULL, Z=NULL, clus, beta.start=NULL,
u.start=NULL, l1cov.start=NULL, l2cov.start=NULL, l1cov.prior=NULL,
l2cov.prior=NULL, nburn=1000, nbetween=1000, nimp=5, a=NULL,
a.prior=NULL, meth="random", output=1, out.iter=10)
```

Arguments

Y.cat	A data frame, or matrix, with categorical (or binary) responses of the joint imputation model. Rows correspond to different observations, while columns are different variables. Missing values are coded as NA.
Y.numcat	A vector with the number of categories in each categorical (or binary) variable.
X	A data frame, or matrix, with covariates of the joint imputation model. Rows correspond to different observations, while columns are different variables. Missing values are not allowed in these variables. In case we want an intercept, a column of 1 is needed. The default is a column of 1.
Z	A data frame, or matrix, for covariates associated to random effects in the joint imputation model. Rows correspond to different observations, while columns are different variables. Missing values are not allowed in these variables. In case we want an intercept, a column of 1 is needed. The default is a column of 1.
clus	A data frame, or matrix, containing the cluster indicator for each observation.
beta.start	Starting value for beta, the vector(s) of fixed effects. Rows index different covariates and columns index different outcomes. For each n-category variable we define n-1 latent normals. The default is a matrix of zeros.
u.start	A matrix where different rows are the starting values within each cluster for the random effects estimates u. The default is a matrix of zeros.
l1cov.start	Starting value for the covariance matrices, stacked one above the other. Dimension of each square matrix is equal to the number of outcomes (continuous plus latent normals) in the imputation model. The default is the identity matrix for each cluster.
l2cov.start	Starting value for the level 2 covariance matrix. Dimension of this square matrix is equal to the number of outcomes (continuous plus latent normals) in the imputation model times the number of random effects. The default is an identity matrix.
l1cov.prior	Scale matrix for the inverse-Wishart prior for the covariance matrices. The default is the identity matrix.
l2cov.prior	Scale matrix for the inverse-Wishart prior for the level 2 covariance matrix. The default is the identity matrix.
nburn	Number of burn in iterations. Default is 1000.
nbetween	Number of iterations between two successive imputations. Default is 1000.
nimp	Number of Imputations. Default is 5.
a	Starting value for the degrees of freedom of the inverse Wishart distribution of the cluster-specific covariance matrices. Default is 50+D, with D being the dimension of the covariance matrices.

a.prior	Hyperparameter (Degrees of freedom) of the chi square prior distribution for the degrees of freedom of the inverse Wishart distribution for the cluster-specific covariance matrices. Default is D, with D being the dimension of the covariance matrices.
meth	When set to "fixed", a flat prior is put on the study-specific covariance matrices and each matrix is updated separately with a different MH-step. When set to "random", we are assuming that all the covariance matrices are draws from an inverse-Wishart distribution, whose parameter values are updated with 2 steps similar to the ones presented in the case of continuous data only for function jomo1ranconhr.
output	When set to any value different from 1 (default), no output is shown on screen at the end of the process.
out.iter	When set to K, every K iterations a dot is printed on screen. Default is 10.

Details

The Gibbs sampler algorithm used is obtained is a mixture of the ones described in chapter 5 and 9 of Carpenter and Kenward (2013). We update the covariance matrices element-wise with a Metropolis-Hastings step. When meth="fixed", we use a flat prior for rhe matrices, while with meth="random" we use an inverse-Wishar tprior and we assume that all the covariance matrices are drawn from an inverse Wishart distribution. We update values of a and A, degrees of freedom and scale matrix of the inverse Wishart distribution from which all the covariance matrices are sampled, from the proper conditional distributions. A flat prior is considered for beta. Binary or continuous covariates in the imputation model may be considered without any problem, but when considering a categorical covariate it has to be included with dummy variables (binary indicators) only.

Value

On screen, the posterior mean of the fixed effects estimates and of the covariance matrix are shown. The only argument returned is the imputed dataset in long format. Column "Imputation" indexes the imputations. Imputation number 0 are the original data.

References

Carpenter J.R., Kenward M.G., (2013), Multiple Imputation and its Application. Chapter 9, Wiley, ISBN: 978-0-470-74052-1.

Yucel R.M., (2011), Random-covariances and mixed-effects models for imputing multivariate multilevel continuous data, Statistical Modelling, 11 (4), 351-370, DOI: 10.1177/1471082X100110040.

Examples

```
# we define the inputs
# nimp, nburn and nbetween are smaller than they should. This is
#just because of CRAN policies on the examples.

Y.cat=cldata[,c("social"), drop=FALSE]
Y.numcat=matrix(4,1,1)
X=data.frame(rep(1,1000),cldata[,c("sex")])
```

```
colnames(X)<-c("const", "sex")
Z<-data.frame(rep(1,1000))
clus<-cldata[,c("city")]
beta.start<-matrix(0,2,3)
u.start<-matrix(0,10,3)
l1cov.start<-matrix(diag(1,3),30,3,2)
l2cov.start<-diag(1,3)
l1cov.prior=diag(1,3);
l2cov.prior=diag(1,3);
a=5
nburn=as.integer(100);
nbetween=as.integer(100);
nimp=as.integer(4);

#Finally we run either the model with fixed or random cluster-specific cov. matrices:

imp<-jomo1rancathr(Y.cat, Y.numcat, X,Z,clus,beta.start,u.start,l1cov.start,
      l2cov.start,l1cov.prior,l2cov.prior,nburn,nbetween,nimp, a, meth="fixed")

cat("Original value was missing (",imp[3,1],"), imputed value:", imp[1003,1])

  # Check help page for function jomo to see how to fit the model and
  # combine estimates with Rubin's rules
```

---

jomo1rancathr.MCMCchain

JM Imputation of clustered data with categorical variables with cluster-specific covariance matrices - A tool to check convergence of the MCMC

Description

This function is similar to jomo1rancathr, but it returns the values of all the parameters in the model at each step of the MCMC instead of the imputations. It is useful to check the convergence of the MCMC sampler.

Usage

```
jomo1rancathr.MCMCchain(Y.cat, Y.numcat, X=NULL, Z=NULL, clus, beta.start=NULL,
u.start=NULL, l1cov.start=NULL, l2cov.start=NULL, l1cov.prior=NULL,
l2cov.prior=NULL, start.imp=NULL, nburn=1000, a=NULL, a.prior=NULL, meth="random",
output=1, out.iter=10)
```

Arguments

Y.cat	A data frame, or matrix, with categorical (or binary) responses of the joint imputation model. Rows correspond to different observations, while columns are different variables. Missing values are coded as NA.
Y.numcat	A vector with the number of categories in each categorical (or binary) variable.
X	A data frame, or matrix, with covariates of the joint imputation model. Rows correspond to different observations, while columns are different variables. Missing values are not allowed in these variables. In case we want an intercept, a column of 1 is needed. The default is a column of 1.
Z	A data frame, or matrix, for covariates associated to random effects in the joint imputation model. Rows correspond to different observations, while columns are different variables. Missing values are not allowed in these variables. In case we want an intercept, a column of 1 is needed. The default is a column of 1.
clus	A data frame, or matrix, containing the cluster indicator for each observation.
beta.start	Starting value for beta, the vector(s) of fixed effects. Rows index different covariates and columns index different outcomes. For each n-category variable we define n-1 latent normals. The default is a matrix of zeros.
u.start	A matrix where different rows are the starting values within each cluster for the random effects estimates u. The default is a matrix of zeros.
l1cov.start	Starting value for the covariance matrices, stacked one above the other. Dimension of each square matrix is equal to the number of outcomes (continuous plus latent normals) in the imputation model. The default is the identity matrix for each cluster.
l2cov.start	Starting value for the level 2 covariance matrix. Dimension of this square matrix is equal to the number of outcomes (continuous plus latent normals) in the imputation model times the number of random effects. The default is an identity matrix.
l1cov.prior	Scale matrix for the inverse-Wishart prior for the covariance matrices. The default is the identity matrix.
l2cov.prior	Scale matrix for the inverse-Wishart prior for the level 2 covariance matrix. The default is the identity matrix.
start.imp	Starting value for the imputed dataset. n-level categorical variables are substituted by n-1 latent normals.
nburn	Number of burn in iterations. Default is 1000.
a	Starting value for the degrees of freedom of the inverse Wishart distribution of the cluster-specific covariance matrices. Default is 50+D, with D being the dimension of the covariance matrices.
a.prior	Hyperparameter (Degrees of freedom) of the chi square prior distribution for the degrees of freedom of the inverse Wishart distribution for the cluster-specific covariance matrices. Default is D, with D being the dimension of the covariance matrices.

meth	When set to "fixed", a flat prior is put on the study-specific covariance matrices and each matrix is updated separately with a different MH-step. When set to "random", we are assuming that all the covariance matrices are draws from an inverse-Wishart distribution, whose parameter values are updated with 2 steps similar to the ones presented in the case of continuous data only for function jomo1ranconhr.
output	When set to any value different from 1 (default), no output is shown on screen at the end of the process.
out.iter	When set to K, every K iterations a dot is printed on screen. Default is 10.

Value

A list with six elements is returned: the final imputed dataset (finimp) and four 3-dimensional matrices, containing all the values for beta (collectbeta), the random effects (collectu) and the level 1 (collectomega) and level 2 covariance matrices (collectcovu). Finally, the final state of the imputed dataset with the latent normals in place of the categorical variables is stored in finimp.latnorm.

Examples

```
#we define the inputs
#  nburn is smaller than needed. This is
#just because of CRAN policies on the examples.

Y.cat=cldata[,c("social"), drop=FALSE]
Y.numcat=matrix(4,1,1)
X=data.frame(rep(1,1000),cldata[,c("sex")])
colnames(X)<-c("const", "sex")
Z<-data.frame(rep(1,1000))
clus<-cldata[,c("city")]
beta.start<-matrix(0,2,3)
u.start<-matrix(0,10,3)
l1cov.start<-matrix(diag(1,3),30,3,2)
l2cov.start<-diag(1,3)
l1cov.prior=diag(1,3);
l2cov.prior=diag(1,3);
a=5
nburn=as.integer(100);

#Finally we run either the model with fixed or random cluster-specific covariance matrices:

imp<-jomo1rancathr.MCMCchain(Y.cat, Y.numcat, X,Z,clus,beta.start,
        u.start,l1cov.start, l2cov.start,l1cov.prior,l2cov.prior,nburn=nburn, a=a, meth="fixed")

#We can check the convergence of the first element of beta:

plot(c(1:nburn),imp$collectbeta[1,1,1:nburn],type="l")

#Or similarly we can check the convergence of any element of th elevel 2 covariance matrix:

plot(c(1:nburn),imp$collectcovu[1,2,1:nburn],type="l")
```

| jomo1rancon | *JM Imputation of clustered data with continuous variables only* |

Description

Impute a clustered dataset with continuous outcomes only. A joint multivariate model for partially observed data is assumed and imputations are generated through the use of a Gibbs sampler. Categorical covariates may be considered, but they have to be included with dummy variables.

Usage

```
jomo1rancon(Y, X=NULL, Z=NULL, clus, beta.start=NULL,u.start=NULL,
l1cov.start=NULL,l2cov.start=NULL, l1cov.prior=NULL, l2cov.prior=NULL,
nburn=1000, nbetween=1000, nimp=5, output=1, out.iter=10)
```

Arguments

Y	A data frame, or matrix, with responses of the joint imputation model. Rows correspond to different observations, while columns are different variables. Missing values are coded as NA.
X	A data frame, or matrix, with covariates of the joint imputation model. Rows correspond to different observations, while columns are different variables. Missing values are not allowed in these variables. In case we want an intercept, a column of 1 is needed. The default is a column of 1.
Z	A data frame, or matrix, for covariates associated to random effects in the joint imputation model. Rows correspond to different observations, while columns are different variables. Missing values are not allowed in these variables. In case we want an intercept, a column of 1 is needed. The default is a column of 1.
clus	A data frame, or matrix, containing the cluster indicator for each observation.
beta.start	Starting value for beta, the vector(s) of fixed effects. Rows index different covariates and columns index different outcomes. The default is a matrix of zeros.
u.start	A matrix where different rows are the starting values within each cluster for the random effects estimates u. The default is a matrix of zeros.
l1cov.start	Starting value for the covariance matrix. Dimension of this square matrix is equal to the number of outcomes in the imputation model. The default is the identity matrix.
l2cov.start	Starting value for the level 2 covariance matrix. Dimension of this square matrix is equal to the number of outcomes in the imputation model times the number of random effects. The default is an identity matrix.
l1cov.prior	Scale matrix for the inverse-Wishart prior for the covariance matrix. The default is the identity matrix.
l2cov.prior	Scale matrix for the inverse-Wishart prior for the level 2 covariance matrix. The default is the identity matrix.

nburn	Number of burn in iterations. Default is 1000.
nbetween	Number of iterations between two successive imputations. Default is 1000.
nimp	Number of Imputations. Default is 5.
output	When set to any value different from 1 (default), no output is shown on screen at the end of the process.
out.iter	When set to K, every K iterations a dot is printed on screen. Default is 10.

Details

The Gibbs sampler algorithm used is a simplification of the one described in detail in Chapter 9 of Carpenter and Kenward (2013), where we exclude the presence of level 2 variables. Regarding the choice of the priors, a flat prior is considered for beta, while an inverse-Wishart prior is given to the covariance matrices, with p-1 degrees of freedom, aka the minimum possible, to guarantee the greatest uncertainty. Binary or continuous covariates in the imputation model may be considered without any problem, but when considering a categorical covariate it has to be included with dummy variables (binary indicators) only.

Value

On screen, the posterior mean of the fixed effects estimates and of the covariance matrix are shown. The only argument returned is the imputed dataset in long format. Column "Imputation" indexes the imputations. Imputation number 0 are the original data.

References

Carpenter J.R., Kenward M.G., (2013), Multiple Imputation and its Application. Chapter 9, Wiley, ISBN: 978-0-470-74052-1.

Examples

```
# we define all the inputs:
Y<-cldata[,c("measure","age")]
clus<-cldata[,c("city")]
X=data.frame(rep(1,1000),cldata[,c("sex")])
colnames(X)<-c("const", "sex")
Z<-data.frame(rep(1,1000))
beta.start<-matrix(0,2,2)
u.start<-matrix(0,10,2)
l1cov.start<-diag(1,2)
l2cov.start<-diag(1,2)
l1cov.prior=diag(1,2);
nburn=as.integer(200);
nbetween=as.integer(200);
nimp=as.integer(5);
l2cov.prior=diag(1,5);

#And finally we run the imputation function:
imp<-jomo1rancon(Y,X,Z,clus,beta.start,u.start,l1cov.start, l2cov.start,l1cov.prior,
```

```
              l2cov.prior,nburn,nbetween,nimp)

cat("Original value was missing(",imp[4,1],"), imputed value:", imp[1004,1])

   # Check help page for function jomo to see how to fit the model and
   # combine estimates with Rubin's rules
```

---

jomo1rancon.MCMCchain   *JM Imputation of clustered data with continuous variables only - A tool to check convergence of the MCMC*

Description

This function is similar to jomo1rancon, but it returns the values of all the parameters in the model at each step of the MCMC instead of the imputations. It is useful to check the convergence of the MCMC sampler.

Usage

```
jomo1rancon.MCMCchain(Y, X=NULL, Z=NULL, clus, beta.start=NULL,
u.start=NULL, l1cov.start=NULL, l2cov.start=NULL, l1cov.prior=NULL,
l2cov.prior=NULL, start.imp=NULL, nburn=1000, output=1, out.iter=10)
```

Arguments

Y	A data frame, or matrix, with responses of the joint imputation model. Rows correspond to different observations, while columns are different variables. Missing values are coded as NA.
X	A data frame, or matrix, with covariates of the joint imputation model. Rows correspond to different observations, while columns are different variables. Missing values are not allowed in these variables. In case we want an intercept, a column of 1 is needed. The default is a column of 1.
Z	A data frame, or matrix, for covariates associated to random effects in the joint imputation model. Rows correspond to different observations, while columns are different variables. Missing values are not allowed in these variables. In case we want an intercept, a column of 1 is needed. The default is a column of 1.
clus	A data frame, or matrix, containing the cluster indicator for each observation.
beta.start	Starting value for beta, the vector(s) of fixed effects. Rows index different covariates and columns index different outcomes. The default is a matrix of zeros.
u.start	A matrix where different rows are the starting values within each cluster for the random effects estimates u. The default is a matrix of zeros.
l1cov.start	Starting value for the covariance matrix. Dimension of this square matrix is equal to the number of outcomes in the imputation model. The default is the identity matrix.

l2cov.start	Starting value for the level 2 covariance matrix. Dimension of this square matrix is equal to the number of outcomes in the imputation model times the number of random effects. The default is an identity matrix.
l1cov.prior	Scale matrix for the inverse-Wishart prior for the covariance matrix. The default is the identity matrix.
l2cov.prior	Scale matrix for the inverse-Wishart prior for the level 2 covariance matrix. The default is the identity matrix.
start.imp	Starting value for the imputed dataset.
nburn	Number of iterations. Default is 1000.
output	When set to any value different from 1 (default), no output is shown on screen at the end of the process.
out.iter	When set to K, every K iterations a dot is printed on screen. Default is 10.

Value

A list with five elements is returned: the final imputed dataset (finimp) and four 3-dimensional matrices, containing all the values for beta (collectbeta), the random effects (collectu) and the level 1 (collectomega) and level 2 covariance matrices (collectcovu).

Examples

```
# define all the inputs:

Y<-cldata[,c("measure","age")]
clus<-cldata[,c("city")]
X=data.frame(rep(1,1000),cldata[,c("sex")])
colnames(X)<-c("const", "sex")
Z<-data.frame(rep(1,1000))
beta.start<-matrix(0,2,2)
u.start<-matrix(0,10,2)
l1cov.start<-diag(1,2)
l2cov.start<-diag(1,2)
l1cov.prior=diag(1,2);
nburn=as.integer(200);

l2cov.prior=diag(1,5);

#And finally we run the imputation function:
imp<-jomo1rancon.MCMCchain(Y,X,Z,clus,beta.start,u.start,l1cov.start,
          l2cov.start,l1cov.prior,l2cov.prior,nburn=nburn)

#We can check the convergence of the first element of beta:

plot(c(1:nburn),imp$collectbeta[1,1,1:nburn],type="l")

#Or similarly we can check the convergence of any element of the level 2 covariance matrix:

plot(c(1:nburn),imp$collectcovu[1,1,1:nburn],type="l")
```

jomo1ranconhr	*JM Imputation of clustered data with continuous variables only with cluster-specific covariance matrices*

Description

Impute a clustered dataset with continuous outcomes only. A joint multivariate model for partially observed data is assumed and imputations are generated through the use of a Gibbs sampler. A different covariance matrix is estimated within each cluster. Categorical covariates may be considered, but they have to be included with dummy variables.

Usage

```
jomo1ranconhr(Y, X=NULL, Z=NULL, clus, beta.start=NULL, u.start=NULL,
l1cov.start=NULL, l2cov.start=NULL, l1cov.prior=NULL, l2cov.prior=NULL,
nburn=1000, nbetween=1000, nimp=5, a=(ncol(Y)+50),a.prior=NULL,
meth="random", output=1, out.iter=10)
```

Arguments

Y	A data frame, or matrix, with responses of the joint imputation model. Rows correspond to different observations, while columns are different variables. Missing values are coded as NA.
X	A data frame, or matrix, with covariates of the joint imputation model. Rows correspond to different observations, while columns are different variables. Missing values are not allowed in these variables. In case we want an intercept, a column of 1 is needed. The default is a column of 1.
Z	A data frame, or matrix, for covariates associated to random effects in the joint imputation model. Rows correspond to different observations, while columns are different variables. Missing values are not allowed in these variables. In case we want an intercept, a column of 1 is needed. The default is a column of 1.
clus	A data frame, or matrix, containing the cluster indicator for each observation.
beta.start	Starting value for beta, the vector(s) of fixed effects. Rows index different covariates and columns index different outcomes. The default is a matrix of zeros.
u.start	A matrix where different rows are the starting values within each cluster for the random effects estimates u. The default is a matrix of zeros.
l1cov.start	Starting value for the covariance matrices, stacked one above the other. Dimension of each square matrix is equal to the number of outcomes in the imputation model. The default is the identity matrix for each cluster.
l2cov.start	Starting value for the level 2 covariance matrix. Dimension of this square matrix is equal to the number of outcomes in the imputation model times the number of random effects. The default is an identity matrix.

l1cov.prior	Scale matrix for the inverse-Wishart prior for the covariance matrices. The default is the identity matrix.
l2cov.prior	Scale matrix for the inverse-Wishart prior for the level 2 covariance matrix. The default is the identity matrix.
nburn	Number of burn in iterations. Default is 1000.
nbetween	Number of iterations between two successive imputations. Default is 1000.
nimp	Number of Imputations. Default is 5.
a	Starting value for the degrees of freedom of the inverse Wishart distribution of the cluster-specific covariance matrices. Default is 50+D, with D being the dimension of the covariance matrices.
a.prior	Hyperparameter (Degrees of freedom) of the chi square prior distribution for the degrees of freedom of the inverse Wishart distribution for the cluster-specific covariance matrices. Default is D, with D being the dimension of the covariance matrices.
meth	This can be set to "Fixed" or "Random". In the first case the function will consider fixed study-specific covariance matrices, in the second, random study-specific distributed according to an inverse-Wishart distribution.
output	When set to any value different from 1 (default), no output is shown on screen at the end of the process.
out.iter	When set to K, every K iterations a dot is printed on screen. Default is 10.

Details

The Gibbs sampler algorithm used is similar to the one described in detail in Chapter 9 of Carpenter and Kenward (2013), where we exclude the presence of level 2 variables and we estimate separetely different covariance matrices within each study. When option meth="random" is specified, all the covariance matrices ae assumed to be random draws from the same underlying inverse Wishart distributions. Details of this algorithm may be found in (Yucel, 2011). Regarding the choice of the priors, a flat prior is considered for beta, while an inverse-Wishart prior is given to the covariance matrices, with p-1 degrees of freedom, aka the minimum possible, to guarantee the greatest uncertainty. Binary or continuous covariates in the imputation model may be considered without any problem, but when considering a categorical covariate it has to be included with dummy variables (binary indicators) only.

Value

On screen, the posterior mean of the fixed effects estimates and of the covariance matrix are shown. The only argument returned is the imputed dataset in long format. Column "Imputation" indexes the imputations. Imputation number 0 are the original data.

References

Carpenter J.R., Kenward M.G., (2013), Multiple Imputation and its Application. Chapter 9, Wiley, ISBN: 978-0-470-74052-1.

Yucel R.M., (2011), Random-covariances and mixed-effects models for imputing multivariate multilevel continuous data, Statistical Modelling, 11 (4), 351-370, DOI: 10.1177/1471082X100110040.

Examples

```
# we define the inputs
# nimp, nburn and nbetween are smaller than they should. This is
#just because of CRAN policies on the examples.

Y<-cldata[,c("measure","age")]
clus<-cldata[,c("city")]
X=data.frame(rep(1,1000),cldata[,c("sex")])
colnames(X)<-c("const", "sex")
Z<-data.frame(rep(1,1000))
beta.start<-matrix(0,2,2)
u.start<-matrix(0,10,2)
l1cov.start<-matrix(diag(1,2),20,2,2)
l2cov.start<-diag(1,2)
l1cov.prior=diag(1,2);
nburn=as.integer(50);
nbetween=as.integer(20);
nimp=as.integer(5);
l2cov.prior=diag(1,5);
a=3

# Finally we run either the model with fixed or random cluster-specific covariance matrices:

imp<-jomo1ranconhr(Y,X,Z,clus,beta.start,u.start,l1cov.start, l2cov.start,
        l1cov.prior,l2cov.prior,nburn,nbetween,nimp,meth="fixed")

cat("Original value was missing(",imp[4,1],"), imputed value:", imp[1004,1])

#or:

#imp<-jomo1ranconhr(Y,X,Z,clus,beta.start,u.start,l1cov.start, l2cov.start,
#        l1cov.prior,l2cov.prior,nburn,nbetween,nimp,a,meth="random")

  # Check help page for function jomo to see how to fit the model and
  # combine estimates with Rubin's rules
```

---

jomo1ranconhr.MCMCchain

> *JM Imputation of clustered data with continuous variables only with cluster-specific covariance matrices - A tool to check convergence of the MCMC*

Description

This function is similar to jomo1ranconhr, but it returns the values of all the parameters in the model at each step of the MCMC instead of the imputations. It is useful to check the convergence of the MCMC sampler.

Usage

```
jomo1ranconhr.MCMCchain(Y, X=NULL, Z=NULL, clus,
beta.start=NULL, u.start=NULL, l1cov.start=NULL,
l2cov.start=NULL, l1cov.prior=NULL, l2cov.prior=NULL,start.imp=NULL,
nburn=1000, a=(ncol(Y)+50),a.prior=NULL, meth="random", output=1, out.iter=10)
```

Arguments

Y	A data frame, or matrix, with responses of the joint imputation model. Rows correspond to different observations, while columns are different variables. Missing values are coded as NA.
X	A data frame, or matrix, with covariates of the joint imputation model. Rows correspond to different observations, while columns are different variables. Missing values are not allowed in these variables. In case we want an intercept, a column of 1 is needed. The default is a column of 1.
Z	A data frame, or matrix, for covariates associated to random effects in the joint imputation model. Rows correspond to different observations, while columns are different variables. Missing values are not allowed in these variables. In case we want an intercept, a column of 1 is needed. The default is a column of 1.
clus	A data frame, or matrix, containing the cluster indicator for each observation.
beta.start	Starting value for beta, the vector(s) of fixed effects. Rows index different covariates and columns index different outcomes. The default is a matrix of zeros.
u.start	A matrix where different rows are the starting values within each cluster for the random effects estimates u. The default is a matrix of zeros.
l1cov.start	in column. Dimension of each square matrix is equal to the number of outcomes in the imputation model. The default is the identity matrix for each cluster.
l2cov.start	Starting value for the level 2 covariance matrix. Dimension of this square matrix is equal to the number of outcomes in the imputation model times the number of random effects. The default is an identity matrix.
l1cov.prior	Scale matrix for the inverse-Wishart prior for the covariance matrices. The default is the identity matrix.
l2cov.prior	Scale matrix for the inverse-Wishart prior for the level 2 covariance matrix. The default is the identity matrix.
start.imp	Starting value for the imputed dataset.
nburn	Number of iterations. Default is 1000.
a	Starting value for the degrees of freedom of the inverse Wishart distribution of the cluster-specific covariance matrices. Default is 50+D, with D being the dimension of the covariance matrices.

a.prior	Hyperparameter (Degrees of freedom) of the chi square prior distribution for the degrees of freedom of the inverse Wishart distribution for the cluster-specific covariance matrices. Default is D, with D being the dimension of the covariance matrices.
meth	This can be set to "Fixed" or "Random". In the first case the function will consider fixed study-specific covariance matrices, in the second, random study-specific distributed according to an inverse-Wishart distribution.
output	When set to any value different from 1 (default), no output is shown on screen at the end of the process.
out.iter	When set to K, every K iterations a dot is printed on screen. Default is 10.

Value

A list with five elements is returned: the final imputed dataset (finimp) and four 3-dimensional matrices, containing all the values for beta (collectbeta), the random effects (collectu) and the level 1 (collectomega) and level 2 covariance matrices (collectcovu).

Examples

```
# we define the inputs
# nburn is smaller than needed. This is
#just because of CRAN policies on the examples.

Y<-cldata[,c("measure","age")]
clus<-cldata[,c("city")]
X=data.frame(rep(1,1000),cldata[,c("sex")])
colnames(X)<-c("const", "sex")
Z<-data.frame(rep(1,1000))
beta.start<-matrix(0,2,2)
u.start<-matrix(0,10,2)
l1cov.start<-matrix(diag(1,2),20,2,2)
l2cov.start<-diag(1,2)
l1cov.prior=diag(1,2);
nburn=as.integer(200);
nbetween=as.integer(200);
nimp=as.integer(5);
l2cov.prior=diag(1,5);
a=3

# Finally we run either the model with fixed or random cluster-specific cov. matrices:

imp<-jomo1ranconhr.MCMCchain(Y,X,Z,clus,beta.start,u.start,l1cov.start, l2cov.start,
          l1cov.prior,l2cov.prior,nburn=nburn,meth="random")

#We can check the convergence of the first element of beta:

plot(c(1:nburn),imp$collectbeta[1,1,1:nburn],type="l")
```

```
#Or similarly we can check the convergence of any element of the level 2 cov. matrix:

plot(c(1:nburn),imp$collectcovu[1,2,1:nburn],type="l")
```

---

jomo1ranmix   *JM Imputation of clustered data with mixed variable types*

Description

Impute a clustered dataset with mixed data types as outcome. A joint multivariate model for partially observed data is assumed and imputations are generated through the use of a Gibbs sampler where the covariance matrix is updated with a Metropolis-Hastings step. Fully observed categorical covariates may be considered as covariates as well, but they have to be included as dummy variables.

Usage

```
jomo1ranmix(Y.con, Y.cat, Y.numcat, X=NULL, Z=NULL, clus,
beta.start=NULL, u.start=NULL, l1cov.start=NULL, l2cov.start=NULL,
l1cov.prior=NULL, l2cov.prior=NULL, nburn=1000, nbetween=1000, nimp=5,
output=1, out.iter=10)
```

Arguments

Y.con	A data frame, or matrix, with continuous responses of the joint imputation model. Rows correspond to different observations, while columns are different variables. Missing values are coded as NA.
Y.cat	A data frame, or matrix, with categorical (or binary) responses of the joint imputation model. Rows correspond to different observations, while columns are different variables. Categories must be integer numbers from 1 to N. Missing values are coded as NA.
Y.numcat	A vector with the number of categories in each categorical (or binary) variable.
X	A data frame, or matrix, with covariates of the joint imputation model. Rows correspond to different observations, while columns are different variables. Missing values are not allowed in these variables. In case we want an intercept, a column of 1 is needed. The default is a column of 1.
Z	A data frame, or matrix, for covariates associated to random effects in the joint imputation model. Rows correspond to different observations, while columns are different variables. Missing values are not allowed in these variables. In case we want an intercept, a column of 1 is needed. The default is a column of 1.
clus	A data frame, or matrix, containing the cluster indicator for each observation.
beta.start	Starting value for beta, the vector(s) of fixed effects. Rows index different covariates and columns index different outcomes. For each n-category variable we define n-1 latent normals. The default is a matrix of zeros.

u.start	A matrix where different rows are the starting values within each cluster for the random effects estimates u. The default is a matrix of zeros.
l1cov.start	Starting value for the covariance matrix. Dimension of this square matrix is equal to the number of outcomes (continuous plus latent normals) in the imputation model. The default is the identity matrix.
l2cov.start	Starting value for the level 2 covariance matrix. Dimension of this square matrix is equal to the number of outcomes (continuous plus latent normals) in the imputation model times the number of random effects. The default is an identity matrix.
l1cov.prior	Scale matrix for the inverse-Wishart prior for the covariance matrix. The default is the identity matrix.
l2cov.prior	Scale matrix for the inverse-Wishart prior for the level 2 covariance matrix. The default is the identity matrix.
nburn	Number of burn in iterations. Default is 1000.
nbetween	Number of iterations between two successive imputations. Default is 1000.
nimp	Number of Imputations. Default is 5.
output	When set to any value different from 1 (default), no output is shown on screen at the end of the process.
out.iter	When set to K, every K iterations a dot is printed on screen. Default is 10.

Details

TThe Gibbs sampler algorithm used is described in detail in Chapter 9 of Carpenter and Kenward (2013). Regarding the choice of the priors, a flat prior is considered for beta and for the covariance matrix. A Metropolis Hastings step is implemented to update the covariance matrix, as described in the book. Binary or continuous covariates in the imputation model may be considered without any problem, but when considering a categorical covariate it has to be included with dummy variables (binary indicators) only.

Value

On screen, the posterior mean of the fixed effects estimates and of the covariance matrix are shown. The only argument returned is the imputed dataset in long format. Column "Imputation" indexes the imputations. Imputation number 0 are the original data.

References

Carpenter J.R., Kenward M.G., (2013), Multiple Imputation and its Application. Chapter 9, Wiley, ISBN: 978-0-470-74052-1.

Examples

```
# we define the inputs:
# nimp, nburn and nbetween are smaller than they should. This is
#just because of CRAN policies on the examples.
```

```
Y.con=cldata[,c("measure","age")]
Y.cat=cldata[,c("social"), drop=FALSE]
Y.numcat=matrix(4,1,1)
X=data.frame(rep(1,1000),cldata[,c("sex")])
colnames(X)<-c("const", "sex")
Z<-data.frame(rep(1,1000))
clus<-cldata[,c("city")]
beta.start<-matrix(0,2,5)
u.start<-matrix(0,10,5)
l1cov.start<-diag(1,5)
l2cov.start<-diag(1,5)
l1cov.prior=diag(1,5);
l2cov.prior=diag(1,5);
nburn=as.integer(50);
nbetween=as.integer(50);
nimp=as.integer(5);

#Then we can run the sampler:

imp<-jomo1ranmix(Y.con, Y.cat, Y.numcat, X,Z,clus,beta.start,u.start,l1cov.start,
          l2cov.start,l1cov.prior,l2cov.prior,nburn,nbetween,nimp)

cat("Original value was missing (",imp[4,1],"), imputed value:", imp[1004,1])

  # Check help page for function jomo to see how to fit the model and
  # combine estimates with Rubin's rules
```

---

jomo1ranmix.MCMCchain   *JM Imputation of clustered data with mixed variable types - A tool to*
                          check convergence of the MCMC

Description

This function is similar to jomo1ranmix, but it returns the values of all the parameters in the model
at each step of the MCMC instead of the imputations. It is useful to check the convergence of the
MCMC sampler.

Usage

```
jomo1ranmix.MCMCchain(Y.con, Y.cat, Y.numcat, X=NULL, Z=NULL, clus,
beta.start=NULL, u.start=NULL, l1cov.start=NULL, l2cov.start=NULL,
l1cov.prior=NULL, l2cov.prior=NULL, start.imp=NULL, nburn=1000,
output=1, out.iter=10)
```

Arguments

Y.con	A data frame, or matrix, with continuous responses of the joint imputation model. Rows correspond to different observations, while columns are different variables. If no continuous outcomes are present in the model, jomo1rancat must be used instead.
Y.cat	A data frame, or matrix, with categorical (or binary) responses of the joint imputation model. Rows correspond to different observations, while columns are different variables. Categories must be integer numbers from 1 to N. Missing values are coded as NA.
Y.numcat	A vector with the number of categories in each categorical (or binary) variable.
X	A data frame, or matrix, with covariates of the joint imputation model. Rows correspond to different observations, while columns are different variables. Missing values are not allowed in these variables. In case we want an intercept, a column of 1 is needed. The default is a column of 1.
Z	A data frame, or matrix, for covariates associated to random effects in the joint imputation model. Rows correspond to different observations, while columns are different variables. Missing values are not allowed in these variables. In case we want an intercept, a column of 1 is needed. The default is a column of 1.
clus	A data frame, or matrix, containing the cluster indicator for each observation.
beta.start	Starting value for beta, the vector(s) of fixed effects. Rows index different covariates and columns index different outcomes. For each n-category variable we define n-1 latent normals. The default is a matrix of zeros.
u.start	A matrix where different rows are the starting values within each cluster for the random effects estimates u. The default is a matrix of zeros.
l1cov.start	Starting value for the covariance matrix. Dimension of this square matrix is equal to the number of outcomes (continuous plus latent normals) in the imputation model. The default is the identity matrix.
l2cov.start	Starting value for the level 2 covariance matrix. Dimension of this square matrix is equal to the number of outcomes (continuous plus latent normals) in the imputation model times the number of random effects. The default is an identity matrix.
l1cov.prior	Scale matrix for the inverse-Wishart prior for the covariance matrix. The default is the identity matrix.
l2cov.prior	Scale matrix for the inverse-Wishart prior for the level 2 covariance matrix. The default is the identity matrix.
start.imp	Starting value for the imputed dataset. n-level categorical variables are substituted by n-1 latent normals.
nburn	Number of burn in iterations. Default is 1000.
output	When set to any value different from 1 (default), no output is shown on screen at the end of the process.
out.iter	When set to K, every K iterations a dot is printed on screen. Default is 10.

Value

A list with six elements is returned: the final imputed dataset (finimp) and four 3-dimensional matrices, containing all the values for beta (collectbeta), the random effects (collectu) and the level 1 (collectomega) and level 2 covariance matrices (collectcovu). Finally, the final state of the imputed dataset with the latent normals in place of the categorical variables is stored in finimp.latnorm.

Examples

```
#we define the inputs:
#  nburn is smaller than necessary. This is
#just because of CRAN policies on the examples.

Y.con=cldata[,c("measure","age")]
Y.cat=cldata[,c("social"), drop=FALSE]
Y.numcat=matrix(4,1,1)
X=data.frame(rep(1,1000),cldata[,c("sex")])
colnames(X)<-c("const", "sex")
Z<-data.frame(rep(1,1000))
clus<-cldata[,c("city")]
beta.start<-matrix(0,2,5)
u.start<-matrix(0,10,5)
l1cov.start<-diag(1,5)
l2cov.start<-diag(1,5)
l1cov.prior=diag(1,5);
l2cov.prior=diag(1,5);
nburn=as.integer(100);

#Then we can run the sampler:

imp<-jomo1ranmix.MCMCchain(Y.con, Y.cat, Y.numcat, X,Z,clus,beta.start,u.start,
            l1cov.start, l2cov.start,l1cov.prior,l2cov.prior,nburn=nburn)

#We can check the convergence of the first element of beta:

plot(c(1:nburn),imp$collectbeta[1,1,1:nburn],type="l")

#Or similarly we can check the convergence of any element of the level 2 covariance matrix:

plot(c(1:nburn),imp$collectcovu[1,2,1:nburn],type="l")
```

---

jomo1ranmixhr   *JM Imputation of clustered data with mixed variable types with*
                        cluster-specific covariance matrices

Description

Impute a clustered dataset with mixed data types as outcome. A joint multivariate model for partially observed data is assumed and imputations are generated through the use of a Gibbs sampler where

a different covariance matrix is sampled within each cluster. Fully observed categorical covariates may be considered as covariates as well, but they have to be included as dummy variables.

Usage

```
jomo1ranmixhr(Y.con, Y.cat, Y.numcat, X=NULL, Z=NULL, clus,
beta.start=NULL, u.start=NULL, l1cov.start=NULL,l2cov.start=NULL,
l1cov.prior=NULL, l2cov.prior=NULL, nburn=1000, nbetween=1000,nimp=5,
a=NULL,a.prior=NULL,meth="random", output=1, out.iter=10)
```

Arguments

Y.con	A data frame, or matrix, with continuous responses of the joint imputation model. Rows correspond to different observations, while columns are different variables. Missing values are coded as NA. If no continuous outcomes are present in the model, jomo1rancathr must be used instead.
Y.cat	A data frame, or matrix, with categorical (or binary) responses of the joint imputation model. Rows correspond to different observations, while columns are different variables. Missing values are coded as NA.
Y.numcat	A vector with the number of categories in each categorical (or binary) variable.
X	A data frame, or matrix, with covariates of the joint imputation model. Rows correspond to different observations, while columns are different variables. Missing values are not allowed in these variables. In case we want an intercept, a column of 1 is needed. The default is a column of 1.
Z	A data frame, or matrix, for covariates associated to random effects in the joint imputation model. Rows correspond to different observations, while columns are different variables. Missing values are not allowed in these variables. In case we want an intercept, a column of 1 is needed. The default is a column of 1.
clus	A data frame, or matrix, containing the cluster indicator for each observation.
beta.start	Starting value for beta, the vector(s) of fixed effects. Rows index different covariates and columns index different outcomes. For each n-category variable we define n-1 latent normals. The default is a matrix of zeros.
u.start	A matrix where different rows are the starting values within each cluster for the random effects estimates u. The default is a matrix of zeros.
l1cov.start	Starting value for the covariance matrices, stacked one above the other. Dimension of each square matrix is equal to the number of outcomes (continuous plus latent normals) in the imputation model. The default is the identity matrix for each cluster.
l2cov.start	Starting value for the level 2 covariance matrix. Dimension of this square matrix is equal to the number of outcomes (continuous plus latent normals) in the imputation model times the number of random effects. The default is an identity matrix.
l1cov.prior	Scale matrix for the inverse-Wishart prior for the covariance matrices. The default is the identity matrix.

l2cov.prior	Scale matrix for the inverse-Wishart prior for the level 2 covariance matrix. The default is the identity matrix.
nburn	Number of burn in iterations. Default is 1000.
nbetween	Number of iterations between two successive imputations. Default is 1000.
nimp	Number of Imputations. Default is 5.
a	Starting value for the degrees of freedom of the inverse Wishart distribution of the cluster-specific covariance matrices. Default is 50+D, with D being the dimension of the covariance matrices.
a.prior	Hyperparameter (Degrees of freedom) of the chi square prior distribution for the degrees of freedom of the inverse Wishart distribution for the cluster-specific covariance matrices. Default is D, with D being the dimension of the covariance matrices.
meth	When set to "fixed", a flat prior is put on the study-specific covariance matrices and each matrix is updated separately with a different MH-step. When set to "random", we are assuming that all the covariance matrices are draws from an inverse-Wishart distribution, whose parameter values are updated with 2 steps similar to the ones presented in the case of continuous data only for function jomo1ranconhr.
output	When set to any value different from 1 (default), no output is shown on screen at the end of the process.
out.iter	When set to K, every K iterations a dot is printed on screen. Default is 10.

Details

The Gibbs sampler algorithm used is obtained is a mixture of the ones described in chapter 5 and 9 of Carpenter and Kenward (2013). We update the covariance matrices element-wise with a Metropolis-Hastings step. When meth="fixed", we use a flat prior for rhe matrices, while with meth="random" we use an inverse-Wishar tprior and we assume that all the covariance matrices are drawn from an inverse Wishart distribution. We update values of a and A, degrees of freedom and scale matrix of the inverse Wishart distribution from which all the covariance matrices are sampled, from the proper conditional distributions. A flat prior is considered for beta. Binary or continuous covariates in the imputation model may be considered without any problem, but when considering a categorical covariate it has to be included with dummy variables (binary indicators) only.

Value

On screen, the posterior mean of the fixed effects estimates and of the covariance matrix are shown. The only argument returned is the imputed dataset in long format. Column "Imputation" indexes the imputations. Imputation number 0 are the original data.

References

Carpenter J.R., Kenward M.G., (2013), Multiple Imputation and its Application. Chapter 9, Wiley, ISBN: 978-0-470-74052-1.

Yucel R.M., (2011), Random-covariances and mixed-effects models for imputing multivariate multilevel continuous data, Statistical Modelling, 11 (4), 351-370, DOI: 10.1177/1471082X100110040.

Examples

```
#we define all the inputs:
# nimp, nburn and nbetween are smaller than they should. This is
#just because of CRAN policies on the examples.

Y.con=cldata[,c("measure","age")]
Y.cat=cldata[,c("social"), drop=FALSE]
Y.numcat=matrix(4,1,1)
X=data.frame(rep(1,1000),cldata[,c("sex")])
colnames(X)<-c("const", "sex")
Z<-data.frame(rep(1,1000))
clus<-cldata[,c("city")]
beta.start<-matrix(0,2,5)
u.start<-matrix(0,10,5)
l1cov.start<-matrix(diag(1,5),50,5,2)
l2cov.start<-diag(1,5)
l1cov.prior=diag(1,5);
l2cov.prior=diag(1,5);
nburn=as.integer(50);
nbetween=as.integer(50);
nimp=as.integer(5);
a=6

# And we are finally able to run the imputation:

imp<-jomo1ranmixhr(Y.con, Y.cat, Y.numcat, X,Z,clus,beta.start,u.start,l1cov.start,
        l2cov.start,l1cov.prior,l2cov.prior,nburn,nbetween,nimp, a, meth="random")

cat("Original value was missing (",imp[4,1],"), imputed value:", imp[1004,1])

  # Check help page for function jomo to see how to fit the model and
  # combine estimates with Rubin's rules
```

---

jomo1ranmixhr.MCMCchain

> *JM Imputation of clustered data with mixed variable types with cluster-specific covariance matrices - A tool to check convergence of the MCMC*

Description

This function is similar to jomo1ranmixhr, but it returns the values of all the parameters in the model at each step of the MCMC instead of the imputations. It is useful to check the convergence of the MCMC sampler.

Usage

```
jomo1ranmixhr.MCMCchain(Y.con, Y.cat, Y.numcat, X=NULL, Z=NULL, clus,
beta.start=NULL, u.start=NULL, l1cov.start=NULL, l2cov.start=NULL,
l1cov.prior=NULL, l2cov.prior=NULL, start.imp=NULL,
nburn=1000, a=NULL,a.prior=NULL,meth="random", output=1, out.iter=10)
```

Arguments

Y.con	A data frame, or matrix, with continuous responses of the joint imputation model. Rows correspond to different observations, while columns are different variables. Missing values are coded as NA. If no continuous outcomes are present in the model, jomo1rancathr must be used instead.
Y.cat	A data frame, or matrix, with categorical (or binary) responses of the joint imputation model. Rows correspond to different observations, while columns are different variables. Missing values are coded as NA.
Y.numcat	A vector with the number of categories in each categorical (or binary) variable.
X	A data frame, or matrix, with covariates of the joint imputation model. Rows correspond to different observations, while columns are different variables. Missing values are not allowed in these variables. In case we want an intercept, a column of 1 is needed. The default is a column of 1.
Z	A data frame, or matrix, for covariates associated to random effects in the joint imputation model. Rows correspond to different observations, while columns are different variables. Missing values are not allowed in these variables. In case we want an intercept, a column of 1 is needed. The default is a column of 1.
clus	A data frame, or matrix, containing the cluster indicator for each observation.
beta.start	Starting value for beta, the vector(s) of fixed effects. Rows index different covariates and columns index different outcomes. For each n-category variable we define n-1 latent normals. The default is a matrix of zeros.
u.start	A matrix where different rows are the starting values within each cluster for the random effects estimates u. The default is a matrix of zeros.
l1cov.start	Starting value for the covariance matrices, stacked one above the other. Dimension of each square matrix is equal to the number of outcomes (continuous plus latent normals) in the imputation model. The default is the identity matrix for each cluster.
l2cov.start	Starting value for the level 2 covariance matrix. Dimension of this square matrix is equal to the number of outcomes (continuous plus latent normals) in the imputation model times the number of random effects. The default is an identity matrix.
l1cov.prior	Scale matrix for the inverse-Wishart prior for the covariance matrices. The default is the identity matrix.
l2cov.prior	Scale matrix for the inverse-Wishart prior for the level 2 covariance matrix. The default is the identity matrix.

start.imp	Starting value for the imputed dataset. n-level categorical variables are substituted by n-1 latent normals.
nburn	Number of iterations. Default is 1000.
a	Starting value for the degrees of freedom of the inverse Wishart distribution of the cluster-specific covariance matrices. Default is 50+D, with D being the dimension of the covariance matrices.
a.prior	Hyperparameter (Degrees of freedom) of the chi square prior distribution for the degrees of freedom of the inverse Wishart distribution for the cluster-specific covariance matrices. Default is D, with D being the dimension of the covariance matrices.
meth	When set to "fixed", a flat prior is used for the study-specific covariance matrices and each matrix is updated separately with a different MH-step. When set to "random", we are assuming that all the covariance matrices are draws from an inverse-Wishart distribution, whose parameter values are updated with 2 steps similar to the ones presented in the case of continuous data only for function jomo1ranconhr.
output	When set to any value different from 1 (default), no output is shown on screen at the end of the process.
out.iter	When set to K, every K iterations a dot is printed on screen. Default is 10.

Value

A list with six elements is returned: the final imputed dataset (finimp) and four 3-dimensional matrices, containing all the values for beta (collectbeta), the random effects (collectu) and the level 1 (collectomega) and level 2 covariance matrices (collectcovu). Finally, the final state of the imputed dataset with the latent normals in place of the categorical variables is stored in finimp.latnorm.

Examples

```
# we define all the inputs:
# nburn is smaller than needed. This is
#just because of CRAN policies on the examples.

Y.con=cldata[,c("measure","age")]
Y.cat=cldata[,c("social"), drop=FALSE]
Y.numcat=matrix(4,1,1)
X=data.frame(rep(1,1000),cldata[,c("sex")])
colnames(X)<-c("const", "sex")
Z<-data.frame(rep(1,1000))
clus<-cldata[,c("city")]
beta.start<-matrix(0,2,5)
u.start<-matrix(0,10,5)
l1cov.start<-matrix(diag(1,5),50,5,2)
l2cov.start<-diag(1,5)
l1cov.prior=diag(1,5);
l2cov.prior=diag(1,5);
nburn=as.integer(80);
```

```
a=6

# And we are finally able to run the imputation:

imp<-jomo1ranmixhr.MCMCchain(Y.con, Y.cat, Y.numcat, X,Z,clus,beta.start,u.start,
    l1cov.start, l2cov.start,l1cov.prior,l2cov.prior,nburn=nburn, a=a)

#We can check the convergence of the first element of beta:

plot(c(1:nburn),imp$collectbeta[1,1,1:nburn],type="l")

#Or similarly we can check the convergence of any element of the level 2 covariance matrix:

plot(c(1:nburn),imp$collectcovu[1,2,1:nburn],type="l")
```

---

jomo2   *JM Imputation of 2-level data*

Description

A wrapper function linking the 2-level JM Imputation functions. The matrices of responses Y and Y2, must be data.frames where continuous variables are numeric and binary/categorical variables are factors.

Usage

```
jomo2(Y, Y2, X=NULL, X2=NULL, Z=NULL,clus, beta.start=NULL, l2.beta.start=NULL,
u.start=NULL, l1cov.start=NULL, l2cov.start=NULL, l1cov.prior=NULL,
l2cov.prior=NULL, nburn=1000, nbetween=1000, nimp=5, a=NULL, a.prior=NULL,
meth="common", output=1, out.iter=10)
```

Arguments

Y   A data.frame with the level-1 outcomes of the imputation model, where columns related to continuous variables are numeric and columns related to binary/categorical variables are factors.

Y2   A data.frame containing the level-2 outcomes of the imputation model, i.e. the partially observed level-2 variables. Columns related to continuous variables have to be numeric and columns related to binary/categorical variables have to be factors.

X   A data frame, or matrix, with covariates of the joint imputation model. Rows correspond to different observations, while columns are different variables. Missing values are not allowed in these variables. In case we want an intercept, a column of 1 is needed. The default is a column of 1.

X2	A data frame, or matrix, with level-2 covariates of the joint imputation model. Rows correspond to different level-1 observations, while columns are different variables. Missing values are not allowed in these variables. In case we want an intercept, a column of 1 is needed. The default is a column of 1.
Z	A data frame, or matrix, for covariates associated to random effects in the joint imputation model. Rows correspond to different observations, while columns are different variables. Missing values are not allowed in these variables. In case we want an intercept, a column of 1 is needed. The default is a column of 1.
clus	A data frame, or matrix, containing the cluster indicator for each observation.
beta.start	Starting value for beta, the vector(s) of level-1 fixed effects. Rows index different covariates and columns index different outcomes. For each n-category variable we have a fixed effect parameter for each of the n-1 latent normals. The default is a matrix of zeros.
l2.beta.start	Starting value for beta2, the vector(s) of level-2 fixed effects. Rows index different covariates and columns index different level-2 outcomes. For each n-category variable we have a fixed effect parameter for each of the n-1 latent normals. The default is a matrix of zeros.
u.start	A matrix where different rows are the starting values within each cluster for the random effects estimates u. The default is a matrix of zeros.
l1cov.start	Starting value for the covariance matrix. Dimension of this square matrix is equal to the number of outcomes (continuous plus latent normals) in the imputation model. The default is the identity matrix.
l2cov.start	Starting value for the level 2 covariance matrix. Dimension of this square matrix is equal to the number of outcomes (continuous plus latent normals) in the imputation model times the number of random effects plus the number of level-2 outcomes. The default is an identity matrix.
l1cov.prior	Scale matrix for the inverse-Wishart prior for the covariance matrix. The default is the identity matrix.
l2cov.prior	Scale matrix for the inverse-Wishart prior for the level 2 covariance matrix. The default is the identity matrix.
nburn	Number of burn in iterations. Default is 1000.
nbetween	Number of iterations between two successive imputations. Default is 1000.
nimp	Number of Imputations. Default is 5.
a	Starting value for the degrees of freedom of the inverse Wishart distribution of the cluster-specific covariance matrices. Default is 50+D, with D being the dimension of the covariance matrices. This is used only when option meth is set to "random".
a.prior	Hyperparameter (Degrees of freedom) of the chi square prior distribution for the degrees of freedom of the inverse Wishart distribution for the cluster-specific covariance matrices. Default is D, with D being the dimension of the covariance matrices.
meth	Method used to deal with level 1 covariance matrix. When set to "common", a common matrix across clusters is used (function jomo2com). When set to

"fixed", fixed study-specific matrices are considered (jomo2hr with option meth="fixed").
Finally, when set to "random", random study-specific matrices are considered
(jomo2hr with option meth="random")

output   When set to any value different from 1 (default), no output is shown on screen
                at the end of the process.

out.iter   When set to K, every K iterations a dot is printed on screen. Default is 10.

Details

This is just a wrapper function to link jomo1rancon, jomo1rancat and jomo1ranmix and the respec-
tive "hr" (heterogeneity in covariance matrices) versions. Format of the columns of Y is crucial in
order for the function to be using the right sub-function.

Value

On screen, the posterior mean of the fixed effects estimates and of the covariance matrix are shown.
The only argument returned is the imputed dataset in long format. Column "Imputation" indexes
the imputations. Imputation number 0 are the original data.

References

Carpenter J.R., Kenward M.G., (2013), Multiple Imputation and its Application. Chapter 9, Wiley,
ISBN: 978-0-470-74052-1.

Examples

```
Y<-tldata[,c("measure.a"), drop=FALSE]
Y2<-tldata[,c("big.city"), drop=FALSE]
clus<-tldata[,c("city")]
nburn=10
nbetween=10
nimp=2

#now we run the imputation function. Note that we would typically use an higher
#number of nburn iterations in real applications (at least 1000)

imp<-jomo2(Y=Y, Y2=Y2, clus=clus,nburn=nburn, nbetween=nbetween, nimp=nimp)

  # Check help page for function jomo to see how to fit the model and
  # combine estimates with Rubin's rules
```

| jomo2.MCMCchain | *JM Imputation of 2-level data - A tool to check convergence of the MCMC* |

Description

This function is similar to jomo2, but it returns the values of all the parameters in the model at each step of the MCMC instead of the imputations. It is useful to check the convergence of the MCMC sampler.

Usage

```
jomo2.MCMCchain(Y, Y2, X=NULL, X2=NULL, Z=NULL, clus, beta.start=NULL,
l2.beta.start=NULL, u.start=NULL, l1cov.start=NULL,l2cov.start=NULL,
l1cov.prior=NULL, l2cov.prior=NULL, start.imp=NULL, l2.start.imp=NULL,
nburn=1000, a=NULL, a.prior=NULL, meth="common", output=1, out.iter=10)
```

Arguments

Y	A data.frame with level-1 outcomes of the imputation model, where columns related to continuous variables are numeric and columns related to binary/categorical variables are factors.
Y2	A data.frame containing the level-2 outcomes of the imputation model. Columns related to continuous variables have to be numeric and columns related to binary/categorical variables have to be factors.
X	A data frame, or matrix, with covariates of the joint imputation model. Rows correspond to different observations, while columns are different variables. Missing values are not allowed in these variables. In case we want an intercept, a column of 1 is needed. The default is a column of 1.
X2	A data frame, or matrix, with level-2 covariates of the joint imputation model. Rows correspond to different level-1 observations, while columns are different variables. Missing values are not allowed in these variables. In case we want an intercept, a column of 1 is needed. The default is a column of 1.
Z	A data frame, or matrix, for covariates associated to random effects in the joint imputation model. Rows correspond to different observations, while columns are different variables. Missing values are not allowed in these variables. In case we want an intercept, a column of 1 is needed. The default is a column of 1.
clus	A data frame, or matrix, containing the cluster indicator for each observation.
beta.start	Starting value for beta, the vector(s) of level-1 fixed effects. Rows index different covariates and columns index different outcomes. For each n-category variable we have a fixed effect parameter for each of the n-1 latent normals. The default is a matrix of zeros.

l2.beta.start	Starting value for beta2, the vector(s) of level-2 fixed effects. Rows index different covariates and columns index different level-2 outcomes. For each n-category variable we have a fixed effect parameter for each of the n-1 latent normals. The default is a matrix of zeros.
u.start	A matrix where different rows are the starting values within each cluster for the random effects estimates u. The default is a matrix of zeros.
l1cov.start	Starting value for the covariance matrix. Dimension of this square matrix is equal to the number of outcomes (continuous plus latent normals) in the imputation model. The default is the identity matrix.
l2cov.start	Starting value for the level 2 covariance matrix. Dimension of this square matrix is equal to the number of outcomes (continuous plus latent normals) in the imputation model times the number of random effects plus the number of level-2 outcomes. The default is an identity matrix.
l1cov.prior	Scale matrix for the inverse-Wishart prior for the covariance matrix. The default is the identity matrix.
l2cov.prior	Scale matrix for the inverse-Wishart prior for the level 2 covariance matrix. The default is the identity matrix.
start.imp	Starting value for the imputed dataset. n-level categorical variables are substituted by n-1 latent normals.
l2.start.imp	Starting value for the level-2 imputed variables. n-level categorical variables are substituted by n-1 latent normals.
nburn	Number of iterations. Default is 1000.
a	Starting value for the degrees of freedom of the inverse Wishart distribution of the cluster-specific covariance matrices. Default is 50+D, with D being the dimension of the covariance matrices. This is used only when option meth is set to "random".
a.prior	Hyperparameter (Degrees of freedom) of the chi square prior distribution for the degrees of freedom of the inverse Wishart distribution for the cluster-specific covariance matrices. Default is D, with D being the dimension of the covariance matrices.
meth	Method used to deal with level 1 covariance matrix. When set to "common", a common matrix across clusters is used (function jomo2com). When set to "fixed", fixed study-specific matrices are considered (jomo2hr with option meth="fixed"). Finally, when set to "random", random study-specific matrices are considered (jomo2hr with option meth="random")
output	When set to any value different from 1 (default), no output is shown on screen at the end of the process.
out.iter	When set to K, every K iterations a dot is printed on screen. Default is 10.

Value

A list is returned; this contains the final imputed dataset (finimp) and several 3-dimensional matrices, containing all the values drawn for each parameter at each iteration: these are, potentially, fixed effect parameters beta (collectbeta), random effects (collectu), level 1 (collectomega) and level 2 covariance matrices (collectcovu) and level-2 fixed effect parameters. If there are some categorical

outcomes, a further output is included in the list, finimp.latnorm, containing the final state of the imputed dataset with the latent normal variables.

Examples

```
Y<-tldata[,c("measure.a"), drop=FALSE]
Y2<-tldata[,c("big.city"), drop=FALSE]
clus<-tldata[,c("city")]
nburn=20

#now we run the imputation function. Note that we would typically use an higher
#number of nburn iterations in real applications (at least 100)

imp<-jomo2.MCMCchain(Y=Y, Y2=Y2, clus=clus,nburn=nburn)

#We can check the convergence of the first element of beta:

plot(c(1:nburn),imp$collectbeta[1,1,1:nburn],type="l")

#Or similarly we can check the convergence of any element of the level 2 covariance matrix:

plot(c(1:nburn),imp$collectcovu[1,2,1:nburn],type="l")
```

---

jomo2com	*JM Imputation of 2-level data assuming a common level-1 covariance matrix across level-2 units.*

Description

Impute a 2-level dataset with mixed data types as outcome. A joint multivariate model for partially observed data is assumed and imputations are generated through the use of a Gibbs sampler where the covariance matrix is updated with a Metropolis-Hastings step. Fully observed categorical covariates may be considered as covariates as well, but they have to be included as dummy variables.

Usage

```
jomo2com(Y.con=NULL, Y.cat=NULL, Y.numcat=NULL, Y2.con=NULL, Y2.cat=NULL,
Y2.numcat=NULL,X=NULL, X2=NULL, Z=NULL, clus, beta.start=NULL, l2.beta.start=NULL,
u.start=NULL, l1cov.start=NULL, l2cov.start=NULL, l1cov.prior=NULL,
l2cov.prior=NULL, nburn=1000, nbetween=1000, nimp=5, output=1, out.iter=10)
```

Arguments

Y.con   A data frame, or matrix, with level-1 continuous responses of the joint imputation model. Rows correspond to different observations, while columns are different variables.

Y.cat	A data frame, or matrix, with categorical (or binary) responses of the joint imputation model. Rows correspond to different observations, while columns are different variables. Missing values are coded as NA.
Y.numcat	A vector with the number of categories in each categorical (or binary) variable.
Y2.con	A data frame, or matrix, with level-2 continuous responses of the joint imputation model. Rows correspond to different observations, while columns are different variables.
Y2.cat	A data frame, or matrix, with level-2 categorical (or binary) responses of the joint imputation model. Rows correspond to different observations, while columns are different variables. Missing values are coded as NA.
Y2.numcat	A vector with the number of categories in each level-2 categorical (or binary) variable.
X	A data frame, or matrix, with covariates of the joint imputation model. Rows correspond to different observations, while columns are different variables. Missing values are not allowed in these variables. In case we want an intercept, a column of 1 is needed. The default is a column of 1.
X2	A data frame, or matrix, with level-2 covariates of the joint imputation model. Rows correspond to different observations, while columns are different variables. Missing values are not allowed in these variables. In case we want an intercept, a column of 1 is needed. The default is a column of 1.
Z	A data frame, or matrix, for covariates associated to random effects in the joint imputation model. Rows correspond to different observations, while columns are different variables. Missing values are not allowed in these variables. In case we want an intercept, a column of 1 is needed. The default is a column of 1.
clus	A data frame, or matrix, containing the cluster indicator for each observation.
beta.start	Starting value for beta, the vector(s) of level-1 fixed effects. Rows index different covariates and columns index different outcomes. For each n-category variable we have a fixed effect parameter for each of the n-1 latent normals. The default is a matrix of zeros.
l2.beta.start	Starting value for beta2, the vector(s) of level-2 fixed effects. Rows index different covariates and columns index different level-2 outcomes. For each n-category variable we have a fixed effect parameter for each of the n-1 latent normals. The default is a matrix of zeros.
u.start	A matrix where different rows are the starting values within each cluster for the random effects estimates u. The default is a matrix of zeros.
l1cov.start	Starting value for the covariance matrix. Dimension of this square matrix is equal to the number of outcomes (continuous plus latent normals) in the imputation model. The default is the identity matrix.
l2cov.start	Starting value for the level 2 covariance matrix. Dimension of this square matrix is equal to the number of outcomes (continuous plus latent normals) in the imputation model times the number of random effects plus the number of level-2 outcomes. The default is an identity matrix.
l1cov.prior	Scale matrix for the inverse-Wishart prior for the covariance matrix. The default is the identity matrix.

l2cov.prior	Scale matrix for the inverse-Wishart prior for the level 2 covariance matrix. The default is the identity matrix.
nburn	Number of burn in iterations. Default is 1000.
nbetween	Number of iterations between two successive imputations. Default is 1000.
nimp	Number of Imputations. Default is 5.
output	When set to any value different from 1 (default), no output is shown on screen at the end of the process.
out.iter	When set to K, every K iterations a dot is printed on screen. Default is 10.

Details

TThe Gibbs sampler algorithm used is described in detail in Chapter 9 of Carpenter and Kenward (2013). Regarding the choice of the priors, a flat prior is considered for beta and for the covariance matrix. A Metropolis Hastings step is implemented to update the covariance matrix, as described in the book. Binary or continuous covariates in the imputation model may be considered without any problem, but when considering a categorical covariate it has to be included with dummy variables (binary indicators) only.

Value

On screen, the posterior mean of the fixed effects estimates and of the covariance matrix are shown. The only argument returned is the imputed dataset in long format. Column "Imputation" indexes the imputations. Imputation number 0 are the original data.

References

Carpenter J.R., Kenward M.G., (2013), Multiple Imputation and its Application. Chapter 9, Wiley, ISBN: 978-0-470-74052-1.

Examples

```
Y<-tldata[,c("measure.a"), drop=FALSE]
Y2<-tldata[,c("big.city"), drop=FALSE]
clus<-tldata[,c("city")]

#now we run the imputation function. Note that we would typically use an higher
#number of nburn iterations in real applications (at least 1000)

imp<-jomo2com(Y.con=Y, Y2.cat=Y2, Y2.numcat=2, clus=clus,nburn=10, nbetween=10, nimp=2)
  # Check help page for function jomo to see how to fit the model and
  # combine estimates with Rubin's rules
```

---

jomo2com.MCMCchain   *JM Imputation of 2-level data assuming a common level-1 covariance matrix across level-2 units - A tool to check convergence of the MCMC*

Description

This function is similar to jomo2com, but it returns the values of all the parameters in the model at each step of the MCMC instead of the imputations. It is useful to check the convergence of the MCMC sampler.

Usage

```
jomo2com.MCMCchain(Y.con=NULL, Y.cat=NULL, Y.numcat=NULL, Y2.con=NULL,
Y2.cat=NULL, Y2.numcat=NULL, X=NULL, X2=NULL, Z=NULL, clus, beta.start=NULL,
l2.beta.start=NULL, u.start=NULL, l1cov.start=NULL, l2cov.start=NULL,
l1cov.prior=NULL, l2cov.prior=NULL, start.imp=NULL, l2.start.imp=NULL, nburn=1000,
output=1, out.iter=10)
```

Arguments

Y.con	A data frame, or matrix, with level-1 continuous responses of the joint imputation model. Rows correspond to different observations, while columns are different variables.
Y.cat	A data frame, or matrix, with categorical (or binary) responses of the joint imputation model. Rows correspond to different observations, while columns are different variables. Missing values are coded as NA.
Y.numcat	A vector with the number of categories in each categorical (or binary) variable.
Y2.con	A data frame, or matrix, with level-2 continuous responses of the joint imputation model. Rows correspond to different observations, while columns are different variables.
Y2.cat	A data frame, or matrix, with level-2 categorical (or binary) responses of the joint imputation model. Rows correspond to different observations, while columns are different variables. Missing values are coded as NA.
Y2.numcat	A vector with the number of categories in each level-2 categorical (or binary) variable.
X	A data frame, or matrix, with covariates of the joint imputation model. Rows correspond to different observations, while columns are different variables. Missing values are not allowed in these variables. In case we want an intercept, a column of 1 is needed. The default is a column of 1.
X2	A data frame, or matrix, with level-2 covariates of the joint imputation model. Rows correspond to different observations, while columns are different variables. Missing values are not allowed in these variables. In case we want an intercept, a column of 1 is needed. The default is a column of 1.

Z	A data frame, or matrix, for covariates associated to random effects in the joint imputation model. Rows correspond to different observations, while columns are different variables. Missing values are not allowed in these variables. In case we want an intercept, a column of 1 is needed. The default is a column of 1.
clus	A data frame, or matrix, containing the cluster indicator for each observation.
beta.start	Starting value for beta, the vector(s) of level-1 fixed effects. Rows index different covariates and columns index different outcomes. For each n-category variable we have a fixed effect parameter for each of the n-1 latent normals. The default is a matrix of zeros.
l2.beta.start	Starting value for beta2, the vector(s) of level-2 fixed effects. Rows index different covariates and columns index different level-2 outcomes. For each n-category variable we have a fixed effect parameter for each of the n-1 latent normals. The default is a matrix of zeros.
u.start	A matrix where different rows are the starting values within each cluster for the random effects estimates u. The default is a matrix of zeros.
l1cov.start	Starting value for the covariance matrix. Dimension of this square matrix is equal to the number of outcomes (continuous plus latent normals) in the imputation model. The default is the identity matrix.
l2cov.start	Starting value for the level 2 covariance matrix. Dimension of this square matrix is equal to the number of outcomes (continuous plus latent normals) in the imputation model times the number of random effects plus the number of level-2 outcomes. The default is an identity matrix.
l1cov.prior	Scale matrix for the inverse-Wishart prior for the covariance matrix. The default is the identity matrix.
l2cov.prior	Scale matrix for the inverse-Wishart prior for the level 2 covariance matrix. The default is the identity matrix.
start.imp	Starting value for the imputed dataset. n-level categorical variables are substituted by n-1 latent normals.
l2.start.imp	Starting value for the level-2 imputed variables. n-level categorical variables are substituted by n-1 latent normals.
nburn	Number of burn in iterations. Default is 1000.
output	When set to any value different from 1 (default), no output is shown on screen at the end of the process.
out.iter	When set to K, every K iterations a dot is printed on screen. Default is 10.

Value

A list is returned; this contains the final imputed dataset (finimp) and several 3-dimensional matrices, containing all the values drawn for each parameter at each iteration: these are, potentially, fixed effect parameters beta (collectbeta), random effects (collectu), level 1 (collectomega) and level 2 covariance matrices (collectcovu) and level-2 fixed effect parameters. If there are some categorical outcomes, a further output is included in the list, finimp.latnorm, containing the final state of the imputed dataset with the latent normal variables.

Examples

```
Y<-tldata[,c("measure.a"), drop=FALSE]
Y2<-tldata[,c("big.city"), drop=FALSE]
clus<-tldata[,c("city")]
nburn=20

#now we run the imputation function. Note that we would typically use an higher
#number of nburn iterations in real applications (at least 100)

imp<-jomo2com.MCMCchain(Y.con=Y, Y2.cat=Y2, Y2.numcat=2, clus=clus,nburn=nburn)

#We can check the convergence of the first element of beta:

plot(c(1:nburn),imp$collectbeta[1,1,1:nburn],type="l")

#Or similarly we can check the convergence of any element of the level 2 covariance matrix:

plot(c(1:nburn),imp$collectcovu[1,2,1:nburn],type="l")
```

---

jomo2hr	*JM Imputation of 2-level data assuming cluster-specific level-1 covariance matrices across level-2 unit*

Description

Impute a 2-level dataset with mixed data types as outcome. A joint multivariate normal model for partially observed data, with (either fixed or random) study-specific covariance matrices is assumed and imputations are generated through the use of a Gibbs sampler where a different covariance matrix is sampled within each cluster. Fully observed categorical covariates may be considered as covariates as well, but they have to be included as dummy variables.

Usage

```
jomo2hr(Y.con=NULL, Y.cat=NULL, Y.numcat=NULL, Y2.con=NULL,
Y2.cat=NULL, Y2.numcat=NULL,X=NULL, X2=NULL, Z=NULL, clus, beta.start=NULL,
l2.beta.start=NULL, u.start=NULL, l1cov.start=NULL, l2cov.start=NULL,
l1cov.prior=NULL, l2cov.prior=NULL, nburn=1000, nbetween=1000, nimp=5,
a=NULL, a.prior=NULL, meth="random", output=1, out.iter=10)
```

Arguments

Y.con
: A data frame, or matrix, with level-1 continuous responses of the joint imputation model. Rows correspond to different observations, while columns are different variables.

Y.cat
: A data frame, or matrix, with categorical (or binary) responses of the joint imputation model. Rows correspond to different observations, while columns are different variables. Missing values are coded as NA.

Y.numcat	A vector with the number of categories in each categorical (or binary) variable.
Y2.con	A data frame, or matrix, with level-2 continuous responses of the joint imputation model. Rows correspond to different observations, while columns are different variables.
Y2.cat	A data frame, or matrix, with level-2 categorical (or binary) responses of the joint imputation model. Rows correspond to different observations, while columns are different variables. Missing values are coded as NA.
Y2.numcat	A vector with the number of categories in each level-2 categorical (or binary) variable.
X	A data frame, or matrix, with covariates of the joint imputation model. Rows correspond to different observations, while columns are different variables. Missing values are not allowed in these variables. In case we want an intercept, a column of 1 is needed. The default is a column of 1.
X2	A data frame, or matrix, with level-2 covariates of the joint imputation model. Rows correspond to different observations, while columns are different variables. Missing values are not allowed in these variables. In case we want an intercept, a column of 1 is needed. The default is a column of 1.
Z	A data frame, or matrix, for covariates associated to random effects in the joint imputation model. Rows correspond to different observations, while columns are different variables. Missing values are not allowed in these variables. In case we want an intercept, a column of 1 is needed. The default is a column of 1.
clus	A data frame, or matrix, containing the cluster indicator for each observation.
beta.start	Starting value for beta, the vector(s) of level-1 fixed effects. Rows index different covariates and columns index different outcomes. For each n-category variable we have a fixed effect parameter for each of the n-1 latent normals. The default is a matrix of zeros.
l2.beta.start	Starting value for beta2, the vector(s) of level-2 fixed effects. Rows index different covariates and columns index different level-2 outcomes. For each n-category variable we have a fixed effect parameter for each of the n-1 latent normals. The default is a matrix of zeros.
u.start	A matrix where different rows are the starting values within each cluster for the random effects estimates u. The default is a matrix of zeros.
l1cov.start	Starting value for the covariance matrices, stacked one above the other. Dimension of each square matrix is equal to the number of outcomes (continuous plus latent normals) in the imputation model. The default is the identity matrix for each cluster.
l2cov.start	Starting value for the level 2 covariance matrix. Dimension of this square matrix is equal to the number of outcomes (continuous plus latent normals) in the imputation model times the number of random effects plus the number of level-2 outcomes. The default is an identity matrix.
l1cov.prior	Scale matrix for the inverse-Wishart prior for the covariance matrices. The default is the identity matrix.
l2cov.prior	Scale matrix for the inverse-Wishart prior for the level 2 covariance matrix. The default is the identity matrix.

nburn	Number of burn in iterations. Default is 1000.
nbetween	Number of iterations between two successive imputations. Default is 1000.
nimp	Number of Imputations. Default is 5.
a	Starting value for the degrees of freedom of the inverse Wishart distribution of the cluster-specific covariance matrices. Default is 50+D, with D being the dimension of the covariance matrices.
a.prior	Hyperparameter (Degrees of freedom) of the chi square prior distribution for the degrees of freedom of the inverse Wishart distribution for the cluster-specific covariance matrices. Default is D, with D being the dimension of the covariance matrices..
meth	When set to "fixed", a flat prior is put on the cluster-specific covariance matrices and each matrix is updated separately with a different MH-step. When set to "random", we are assuming that all the cluster-specific level-1 covariance matrices are draws from an inverse-Wishart distribution, whose parameter values are updated with 2 steps similar to the ones presented in the case of clustered data for function jomo1ranconhr.
output	When set to any value different from 1 (default), no output is shown on screen at the end of the process.
out.iter	When set to K, every K iterations a dot is printed on screen. Default is 10.

Details

The Gibbs sampler algorithm used is obtained is a mixture of the ones described in chapter 5 and 9 of Carpenter and Kenward (2013). We update the covariance matrices element-wise with a Metropolis-Hastings step. When meth="fixed", we use a flat prior for rhe matrices, while with meth="random" we use an inverse-Wishar tprior and we assume that all the covariance matrices are drawn from an inverse Wishart distribution. We update values of a and A, degrees of freedom and scale matrix of the inverse Wishart distribution from which all the covariance matrices are sampled, from the proper conditional distributions. A flat prior is considered for beta. Binary or continuous covariates in the imputation model may be considered without any problem, but when considering a categorical covariate it has to be included with dummy variables (binary indicators) only.

Value

On screen, the posterior mean of the fixed effects estimates and of the covariance matrix are shown. The only argument returned is the imputed dataset in long format. Column "Imputation" indexes the imputations. Imputation number 0 are the original data.

References

Carpenter J.R., Kenward M.G., (2013), Multiple Imputation and its Application. Chapter 9, Wiley, ISBN: 978-0-470-74052-1.

Yucel R.M., (2011), Random-covariances and mixed-effects models for imputing multivariate multilevel continuous data, Statistical Modelling, 11 (4), 351-370, DOI: 10.1177/1471082X100110040.

Examples

```
Y<-tldata[,c("measure.a"), drop=FALSE]
Y2<-tldata[,c("big.city"), drop=FALSE]
clus<-tldata[,c("city")]

#now we run the imputation function. Note that we would typically use an higher
#number of nburn iterations in real applications (at least 1000)

imp<-jomo2hr(Y.con=Y, Y2.cat=Y2, Y2.numcat=2, clus=clus,nburn=10, nbetween=10, nimp=2)

  # Check help page for function jomo to see how to fit the model and
  # combine estimates with Rubin's rules
```

---

jomo2hr.MCMCchain | *JM Imputation of 2-level data assuming cluster-specific level-1 covariance matrices across level-2 units- A tool to check convergence of the MCMC*

Description

This function is similar to jomo2hr, but it returns the values of all the parameters in the model at each step of the MCMC instead of the imputations. It is useful to check the convergence of the MCMC sampler.

Usage

```
jomo2hr.MCMCchain(Y.con=NULL, Y.cat=NULL, Y.numcat=NULL, Y2.con=NULL,
Y2.cat=NULL, Y2.numcat=NULL, X=NULL, X2=NULL, Z=NULL, clus, beta.start=NULL,
l2.beta.start=NULL, u.start=NULL, l1cov.start=NULL, l2cov.start=NULL,
l1cov.prior=NULL, l2cov.prior=NULL, start.imp=NULL, l2.start.imp=NULL,
nburn=1000, a=NULL,a.prior=NULL,meth="random", output=1, out.iter=10)
```

Arguments

Y.con	A data frame, or matrix, with level-1 continuous responses of the joint imputation model. Rows correspond to different observations, while columns are different variables.
Y.cat	A data frame, or matrix, with categorical (or binary) responses of the joint imputation model. Rows correspond to different observations, while columns are different variables. Missing values are coded as NA.
Y.numcat	A vector with the number of categories in each categorical (or binary) variable.
Y2.con	A data frame, or matrix, with level-2 continuous responses of the joint imputation model. Rows correspond to different observations, while columns are different variables.

`Y2.cat`	A data frame, or matrix, with level-2 categorical (or binary) responses of the joint imputation model. Rows correspond to different observations, while columns are different variables. Missing values are coded as NA.
`Y2.numcat`	A vector with the number of categories in each level-2 categorical (or binary) variable.
`X`	A data frame, or matrix, with covariates of the joint imputation model. Rows correspond to different observations, while columns are different variables. Missing values are not allowed in these variables. In case we want an intercept, a column of 1 is needed. The default is a column of 1.
`X2`	A data frame, or matrix, with level-2 covariates of the joint imputation model. Rows correspond to different observations, while columns are different variables. Missing values are not allowed in these variables. In case we want an intercept, a column of 1 is needed. The default is a column of 1.
`Z`	A data frame, or matrix, for covariates associated to random effects in the joint imputation model. Rows correspond to different observations, while columns are different variables. Missing values are not allowed in these variables. In case we want an intercept, a column of 1 is needed. The default is a column of 1.
`clus`	A data frame, or matrix, containing the cluster indicator for each observation.
`beta.start`	Starting value for beta, the vector(s) of level-1 fixed effects. Rows index different covariates and columns index different outcomes. For each n-category variable we have a fixed effect parameter for each of the n-1 latent normals. The default is a matrix of zeros.
`l2.beta.start`	Starting value for beta2, the vector(s) of level-2 fixed effects. Rows index different covariates and columns index different level-2 outcomes. For each n-category variable we have a fixed effect parameter for each of the n-1 latent normals. The default is a matrix of zeros.
`u.start`	A matrix where different rows are the starting values within each cluster for the random effects estimates u. The default is a matrix of zeros.
`l1cov.start`	Starting value for the covariance matrices, stacked one above the other. Dimension of each square matrix is equal to the number of outcomes (continuous plus latent normals) in the imputation model. The default is the identity matrix for each cluster.
`l2cov.start`	Starting value for the level 2 covariance matrix. Dimension of this square matrix is equal to the number of outcomes (continuous plus latent normals) in the imputation model times the number of random effects plus the number of level-2 outcomes. The default is an identity matrix.
`l1cov.prior`	Scale matrix for the inverse-Wishart prior for the covariance matrices. The default is the identity matrix.
`l2cov.prior`	Scale matrix for the inverse-Wishart prior for the level 2 covariance matrix. The default is the identity matrix.
`start.imp`	Starting value for the imputed dataset. n-level categorical variables are substituted by n-1 latent normals.
`l2.start.imp`	Starting value for the level-2 imputed variables. n-level categorical variables are substituted by n-1 latent normals.

nburn	Number of iterations. Default is 1000.
a	Starting value for the degrees of freedom of the inverse Wishart distribution of the cluster-specific covariance matrices. Default is 50+D, with D being the dimension of the covariance matrices.
a.prior	Hyperparameter (Degrees of freedom) of the chi square prior distribution for the degrees of freedom of the inverse Wishart distribution for the cluster-specific covariance matrices. Default is D, with D being the dimension of the covariance matrices.
meth	When set to "fixed", a flat prior is put on the cluster-specific covariance matrices and each matrix is updated separately with a different MH-step. When set to "random", we are assuming that all the cluster-specific level-1 covariance matrices are draws from an inverse-Wishart distribution, whose parameter values are updated with 2 steps similar to the ones presented in the case of clustered data for function jomo1ranconhr.
output	When set to any value different from 1 (default), no output is shown on screen at the end of the process.
out.iter	When set to K, every K iterations a dot is printed on screen. Default is 10.

Value

A list is returned; this contains the final imputed dataset (finimp) and several 3-dimensional matrices, containing all the values drawn for each parameter at each iteration: these are, potentially, fixed effect parameters beta (collectbeta), random effects (collectu), level 1 (collectomega) and level 2 covariance matrices (collectcovu) and level-2 fixed effect parameters. If there are some categorical outcomes, a further output is included in the list, finimp.latnorm, containing the final state of the imputed dataset with the latent normal variables.

Examples

```
Y<-tldata[,c("measure.a"), drop=FALSE]
Y2<-tldata[,c("big.city"), drop=FALSE]
clus<-tldata[,c("city")]
nburn=20

#now we run the imputation function. Note that we would typically use an higher
#number of nburn iterations in real applications (at least 100)

imp<-jomo2hr.MCMCchain(Y.con=Y, Y2.cat=Y2, Y2.numcat=2, clus=clus,nburn=nburn)

#We can check the convergence of the first element of beta:

plot(c(1:nburn),imp$collectbeta[1,1,1:nburn],type="l")

#Or similarly we can check the convergence of any element of the level 2 covariance matrix:

plot(c(1:nburn),imp$collectcovu[1,2,1:nburn],type="l")
```

JSPmiss   *Exam results for six inner London Education Authorities*

Description

A partially observed version of the jspmix1 dataset in package R2MLwiN. This is an educational
dataset of pupils' test scores, a subset of the Junior School Project (Mortimore et al, 1988).

Usage

```
data(cldata)
```

Format

A data frame with 4059 observations on the following 6 variables.

school A school identifier.

id A student ID.

fluent Fluency in English indicator, where 0 = beginner, 1 = intermediate, 2 = fully fluent; mea-
    sured in Year 1.

sex Sex of pupil; numeric with levels 0 (boy), 1 (girl).

cons A column of 1s. Useful to add an intercept to th eimputation model.

ravens Test score, out of 40; measured in Year 1.

english Pupils' English test score, out of 100; measured in Year 3.

behaviour Pupils' behaviour score, where lowerquarter = pupil rated in bottom 25%, and upper
    otherwise; measured in Year 3.

Details

These fully observed verison of the data is available with package R2MLwiN.

Source

Browne, W. J. (2012) MCMC Estimation in MLwiN Version 2.26. University of Bristol: Centre for
Multilevel Modelling.

Mortimore, P., Sammons, P., Stoll, L., Lewis, D., Ecob, R. (1988) School Matters. Wells: Open
Books.

Rasbash, J., Charlton, C., Browne, W.J., Healy, M. and Cameron, B. (2009) MLwiN Version 2.1.
Centre for Multilevel Modelling, University of Bristol.

---

sldata *A simulated single level dataset*

Description

A simulated dataset to test single level functions, i.e. jomo1con, jomo1cat and jomo1mix.

Usage

```
data(sldata)
```

Format

A data frame with 300 observations on the following 4 variables.

age A numeric variable with age. Fully observed.

measure A numeric variable with some measure of interest (unspecified). This is partially observed.

sex A binary variable for gender indicator. Fully observed.

social A 4-category variable with a social status indicator. This is partially observed.

Details

These are not real data, they are simulated to illustrate the use of the main functions of the package.

---

surdata *A simulated dataset with survival data*

Description

A simulated dataset to test functions for imputation compatible with cox model.

Usage

```
data(cldata)
```

Format

A data frame with 500 observations on the following 5 variables.

measure A numeric variable with some measure of interest (unspecified). This is partially observed.

sex A binary variable with gender indicator. Partially observed.

id The id for individuals within each city.

time Time to event (death or censoring).

status Binary variables, which takes value 0 for censored observations and 1 for deaths/events.

Details

These are not real data, they are simulated to illustrate the use of the main functions of the package.

---

tldata   *A simulated 2-level dataset*

Description

A simulated dataset to test 2-level functions, i.e. jomo2com and jomo2hr.

Usage

```
data(tldata)
```

Format

A data frame with 1000 observations on the following 6 variables.

measure.a A numeric variable with some measure of interest (unspecified). This is partially observed.

measure.b A numeric variable with some measure of interest (unspecified). This is fully observed.

measure.a2 A numeric variable with some level-2 measure of interest (unspecified). This is partially observed.

previous.events A binary variable indicating if a patient has previous history of (unspecified) events. Patially observed.

group A 3-category variable indicating to which group each patient belongs. This is partially observed.

big.city A binary variable indicating if each city has more than 100000 inhabitants. Patially observed.

region A 3-category variable indicating to which region each city belongs. This is fully observed.

city The cluster indicator vector. 200 cities are indexed 0 to 199.

id The id for each individual within each city.

Details

These are not real data, they are simulated to illustrate the use of the main functions of the package.

Index