Package ‘MCMCglmm’

November 14, 2016

Version 2.24
Date 2016-11-14
Title MCMC Generalised Linear Mixed Models
Depends Matrix, coda, ape
Imports corpcor, tensorA, cubature, methods
Suggests rgl, combinat, mvtnorm, orthopolynom
Author Jarrod Hadfield
Maintainer Jarrod Hadfield <j.hadfield@ed.ac.uk>
Description MCMC Generalised Linear Mixed Models.
License GPL (>= 2)
SystemRequirements C++11
NeedsCompilation yes
Repository CRAN
Date/Publication 2016-11-14 13:22:11

R topics documented:

MCMCglmm-package ......................................................... 2
at.level ................................................................. 3
at.set ................................................................. 4
BTdata ................................................................. 5
BTped ................................................................. 5
buildV ................................................................. 6
commutation ........................................................... 7
dcmvnorm ............................................................. 7
Ddivergence ............................................................ 8
Dexpressions ........................................................... 9
Dtensor ................................................................. 10
evalDtensor .......................................................... 11
gelman.prior ........................................................ 12
inverseA .............................................................. 13
MCMCglmm-package

Multivariate Generalised Linear Mixed Models

Description

MCMCglmm is a package for fitting Generalised Linear Mixed Models using Markov chain Monte Carlo techniques (Hadfield 2009). Most commonly used distributions like the normal and the Poisson are supported together with some useful but less popular ones like the zero-inflated Poisson and the multinomial. Missing values and left, right and interval censoring are accommodated for all traits. The package also supports multi-trait models where the multiple responses can follow different types of distribution. The package allows various residual and random-effect variance structures.
to be specified including heterogeneous variances, unstructured covariance matrices and random regression (e.g. random slope models). Three special types of variance structure that can be specified are those associated with pedigrees (animal models), phylogenies (the comparative method) and measurement error (meta-analysis).

The package makes heavy use of results in Sorensen & Gianola (2002) and Davis (2006) which taken together result in what is hopefully a fast and efficient routine. Most small to medium sized problems should take seconds to a few minutes, but large problems (> 20,000 records) are possible. My interest is in evolutionary biology so there are also several functions for applying Rice’s (2004) tensor analysis to real data and functions for visualising and comparing matrices.

Please read the tutorial vignette("Tutorial", "MCMCglmm") or the course notes vignette("CourseNotes", "MCMCglmm")

Author(s)
Jarrod Hadfield <j.hadfield@ed.ac.uk>

References

---

**at.level**

*Incidence Matrix of Levels within a Factor*

Description

Incidence matrix of levels within a factor

Usage

```r
at.level(x, level)
```

Arguments

- `x` factor
- `level` factor level

Value

incidence matrix for level in x

Author(s)
Jarrod Hadfield <j.hadfield@ed.ac.uk>
See Also

at.set

Examples

```r
fac <- gl(3,10,30, labels=letters[1:3])
x <- rnorm(30)
model.matrix(~at.level(fac,"b")::x)
```

---

**at.set**

*Incidence Matrix of Combined Levels within a Factor*

Description

Incidence Matrix of Combined Levels within a Factor

Usage

```r
at.set(x, level)
```

Arguments

- `x`: factor
- `level`: set of factor levels

Value

incidence matrix for the set `level` in `x`

Author(s)

Jarrod Hadfield <j.hadfield@ed.ac.uk>

See Also

at.level

Examples

```r
fac <- gl(3,10,30, labels=letters[1:3])
x <- rnorm(30)
model.matrix(~at.set(fac,2:3)::x)
```
Blue Tit Data for a Quantitative Genetic Experiment

Description

Blue Tit (Cyanistes caeruleus) Data for a Quantitative Genetic Experiment

Usage

BTdata

Format

a data frame with 828 rows and 7 columns, with variables tarsus length (tarsus) and colour (back) measured on 828 individuals (animal). The mother of each is also recorded (dam) together with the foster nest (fosternest) in which the chicks were reared. The date on which the first egg in each nest hatched (hatchdate) is recorded together with the sex (sex) of the individuals.

References

Hadfield, J.D. et. al. 2007 Journal of Evolutionary Biology 20 549-557

See Also

BTped

Blue Tit Pedigree

Description

Blue Tit (Cyanistes caeruleus) Pedigree

Usage

BTped

Format

a data frame with 1040 rows and 3 columns, with individual identifier (animal) mother identifier (dam) and father identifier (sire). The first 212 rows are the parents of the 828 offspring from 106 full-sibling families. Parents are assumed to be unrelated to each other and have NA's in the dam and sire column.

References

Hadfield, J.D. et. al. 2007 Journal of Evolutionary Biology 20 549-557
buildV

Forms expected (co)variances for GLMMs fitted with MCMCglmm

Description

Forms the expected covariance structure of link-scale observations for GLMMs fitted with MCMCglmm.

Usage

buildV(object, marginal=object$Random$formula, diag=TRUE, it=NULL, posterior="mean", ...)

Arguments

- **object**: an object of class "MCMCglmm"
- **marginal**: formula defining random effects to be marginalised
- **diag**: logical; if TRUE the covariances between observations are not calculated
- **it**: integer; optional, MCMC iteration on which covariance structure should be based
- **posterior**: character; if it is NULL should the covariance structure be based on the marginal posterior means ('mean') of the VCV parameters, or the posterior modes ('mode'), or a random draw from the posterior with replacement ('distribution'). If posterior="all" the posterior distribution of observation variances is returned
- **...**: Further arguments to be passed

Value

If diag=TRUE an n by n covariance matrix. If diag=FALSE and posterior!="all" an 1 by n matrix of variances. If posterior="all" an nit by n matrix of variances (where nit is the number of saved MCMC iterations).

Author(s)

Jarrod Hadfield <j.hadfield@ed.ac.uk>

See Also

MCMCglmm
commutation

Commutation Matrix

Description
Forms an mn x mn commutation matrix which transforms vec(A) into vec(A\'), where A is an m x n matrix.

Usage
commutation(m, n)

Arguments
m integer; number of rows of A
n integer; number of columns of A

Value
Commutation Matrix

Author(s)
Jarrod Hadfield <j.hadfield@ed.ac.uk>

References

Examples
commutation(2,2)

dcmvnorm

Density of a (conditional) multivariate normal variate

Description
Density of a (conditional) multivariate normal variate

Usage
dcmvnorm(x, mean = 0, V = 1, keep=1, cond=(1:length(x))[-keep], log=FALSE)
**Arguments**

- **x**: vector of observations
- **mean**: vector of means
- **V**: covariance matrix
- **keep**: vector of integers: observations for which density is required
- **cond**: vector of integers: observations to condition on
- **log**: if TRUE, density p is given as log(p)

**Value**

numeric

**Author(s)**

Jarrod Hadfield <j.hadfield@ed.ac.uk>

**Examples**

```r
V1 <- cbind(c(1, 0.5), c(0.5, 1))
dcmvnorm(c(0, 2), c(0, 0), V=V1, keep=1, cond=2) # density of x[1]=0 conditional on x[2]=2 given # x ~ MVN(c(0, 0), V1)

dcmvnorm(c(0, 2), c(0, 0), V=V1, keep=1, cond=NULL) # density of x[1]=0 marginal to x[2]
dnorm(0, 0, 1) # same as univariate density

V2 <- diag(2)
dcmvnorm(c(0, 2), c(0, 0), V=V2, keep=1, cond=2) # density of x[1]=0 conditional on x[2]=2 given # x ~ MVN(c(0, 0), V2)
dnorm(0, 0, 1) # same as univariate density because V2 is diagonal
```

**Dddivergence**

**$d$-divergence**

**Description**

Calculates Ovaskainen’s (2008) $d$-divergence between 2 zero-mean multivariate normal distributions.

**Usage**

```r
Dddivergence(CA=NULL, CB=NULL, n=10000)
```
**Dexpressions**

**Arguments**
- CA: Matrix A
- CB: Matrix B
- n: number of Monte Carlo samples for approximating the integral

**Value**
- d-divergence

**Author(s)**
- Jarrod Hadfield <j.hadfield@ed.ac.uk>

**References**

**Examples**
```
CA<-rIW(diag(2),10, n=1)
CB<-rIW(diag(2),10, n=1)
Ddivergence(CA, CB)
```

**Value**

<table>
<thead>
<tr>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>List of unevaluated expressions for (mixed) partial derivatives of fitness with respect to linear predictors.</td>
<td>Dexpressions</td>
</tr>
</tbody>
</table>

**Description**
Unevaluated expressions for (mixed) partial derivatives of fitness with respect to linear predictors for survival and fecundity.

**Usage**
Dexpressions

**Value**
```
PW.d0W: Fitness (W) function for the Poisson-Weibull (PW) model.
PW.d1Ws: First Partial derivative of fitness (d1W) with respect to survival (d1s) linear predictor for the Poisson-Weibull (PW) model.
PW.d1Wf: First Partial derivative of fitness (d1W) with respect to fecundity (d1f) linear predictor for the Poisson-Weibull (PW) model.
PW.d3Wd2sd1f: Mixed third partial derivative of fitness (d3W) with 2nd derivative of survival linear predictor (d2s) and first derivative of fecundity linear predictor (d1f) from the Poisson-Weibull (PW) model.
```
**Dtensor**


dtensor

Tensor of (mixed) partial derivatives

**Description**

Forms tensor of (mixed) partial derivatives

**Usage**


dtensor(expr, name=NULL, mu = NULL, m=1, evaluate = TRUE)

**Arguments**


<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>expr</td>
<td>'expression'</td>
</tr>
<tr>
<td>name</td>
<td>character vector, giving the variable names with respect to which derivatives will be computed. If NULL all variables in the expression will be used</td>
</tr>
<tr>
<td>mu</td>
<td>optional: numeric vector, at which the derivatives are evaluated</td>
</tr>
<tr>
<td>m</td>
<td>order of derivative</td>
</tr>
<tr>
<td>evaluate</td>
<td>logical; if TRUE the derivatives are evaluated at mu, if FALSE the derivatives are left unevaluated</td>
</tr>
</tbody>
</table>

**Value**

dtensor (list) of unevaluated expression(s) if evaluate=FALSE or a tensor if evaluate=TRUE

**Author(s)**

Jarrod Hadfield j.hadfield@ed.ac.uk

**References**

evalDtensor

See Also
evalDtensor, Dexpressions, D

Examples

\[ f <- \text{expression}(\beta_1 \times \text{time} \times \beta_2 \times u) \]
\[ \text{Dtensor}(f, \text{eval}=\text{FALSE}) \]

evalDtensor Evaluates a list of (mixed) partial derivatives

Description

Evaluates a list of (mixed) partial derivatives

Usage

evalDtensor(x, mu, m=1)

Arguments

x unevaluated (list) of expression(s)
mu values at which the derivatives are evaluated: names need to match terms in x
m order of derivative

Value
tensor

Author(s)

Jarrod Hadfield <j.hadfield@ed.ac.uk>

See Also

Dtensor, D

Examples

\[ f <- \text{expression}(\beta_1 \times \text{time} \times \beta_2 \times u) \]
\[ \text{Dtensor}(f, \text{eval}=\text{FALSE}, m=2) \]
\[ \text{evalDtensor}(\text{Dt}, \text{mu}=\text{data.frame(} \beta_1=0.5, \beta_2=1, \text{time}=3, u=2.3)) \]
\[ \text{Dtensor}(f, \text{mu}=\text{c}(1,3,1,2.3), m=2) \]
Prior Covariance Matrix for Fixed Effects.

Description
Prior Covariance Matrix for Fixed Effects.

Usage
```
gelman.prior(formula, data, scale=1, intercept=scale, singular.ok=FALSE)
```

Arguments
- `formula`: formula for the fixed effects.
- `data`: data.frame.
- `intercept`: prior standard deviation for the intercept
- `scale`: prior standard deviation for regression parameters
- `singular.ok`: logical; if FALSE linear dependencies in the fixed effects are removed. if TRUE they are left in an estimated, although all information comes form the prior

Details
Gelman et al. (2008) suggest that the input variables of a categorical regression are standardised and that the associated regression parameters are assumed independent in the prior. Gelman et al. (2008) recommend a scaled t-distribution with a single degree of freedom (scaled Cauchy) and a scale of 10 for the intercept and 2.5 for the regression parameters. If the degree of freedom is infinity (i.e. a normal distribution) then a prior covariance matrix \( bDv \) can be defined for the regression parameters without input standardisation that corresponds to a diagonal prior \( D \) for the regression parameters had the inputs been standardised. The diagonal elements of \( D \) are set to \( scale^2 \) except the first which is set to \( intercept^2 \). With logistic regression \( D = \pi^2/3 + \sigma^2 \) gives a prior that is approximately flat on the probability scale, where \( \sigma^2 \) is the total variance due to the random effects. For probit regression it is \( D = 1 + \sigma^2 \).

Value
- prior covariance matrix

Author(s)
- Jarrod Hadfield <j.hadfield@ed.ac.uk>

References
Examples

\begin{verbatim}
dat<-data.frame(y=c(0,0,1,1), x=gl(2,2))  # data with complete separation

# probit regression #

prior1<-list(B=list(mu=c(0,0), V=gelman.prior(~x, data=dat, scale=sqrt(1+1))),
              R=list(V=1,fix=1))

m1<-MCMCglmm(y~x, prior=prior1, data=dat, family="ordinal", verbose=FALSE)

c2<-1
p1<-pnorm(m1$Sol[,1]/sqrt(1+c2))  # marginal probability when x=1

# logistic regression #

prior2<-list(B=list(mu=c(0,0), V=gelman.prior(~x, data=dat, scale=sqrt(pi^2/3+1))),
              R=list(V=1,fix=1))

m2<-MCMCglmm(y~x, prior=prior2, data=dat, family="categorical", verbose=FALSE)

c2 <- (16 * sqrt(3)/(15 * pi))^2
p2<-plogis(m2$Sol[,1]/sqrt(1+c2))  # marginal probability when x=1

plot(mcmc.list(p1,p2))
\end{verbatim}

Description

Henderson (1976) and Meuwissen and Luo (1992) algorithm for inverting relatedness matrices, and
Hadfield and Nakagawa (2010) algorithm for inverting phylogenetic covariance matrices.

Usage

inverseA(pedigree=NULL, nodes="ALL", scale=TRUE, reduced=FALSE)

Arguments

pedigree ordered pedigree with 3 columns: id, dam and sire, or a phylo object.
nodes "ALL" calculates the inverse for all individuals/nodes. For phylogenies "TIPS" calculates the inverse for the species tips only, and for pedigrees a vector of id's can be passed which inverts the relatedness matrix for that subset.

scale logical: should a phylogeny (needs to be ultrametric) be scaled to unit length (distance from root to tip)?

reduced logical: should childless nodes be dropped from the inverse and the pedigree/phylogeny representation be reduced?

Value

- Ainv inverse as sparseMatrix
- inbreeding inbreeding coefficients/branch lengths
- pedigree pedigree/pedigree representation of phylogeny

Author(s)

Jarrod Hadfield <j.hadfield@ed.ac.uk>

References


Examples

data(bird.families)
Ainv<-inverseA(bird.families)

knorm (Mixed) Central Moments of a Multivariate Normal Distribution

Description

Forms a tensor of (mixed) central moments of a multivariate normal distribution

Usage

knorm(V, k)

Arguments

- V (co)variance matrix
- k kth central moment, must be even
KPPM

Value
tensor

Author(s)
Jarrod Hadfield <j.hadfield@ed.ac.uk>

References
Schott, J.R.(2003) Journal of Multivariate Analysis 87 (1) 177-190

See Also
dnorm

Examples
V<-diag(2)
knorm(V,2)
knorm(V,4)

KPPM
Kronecker Product Permutation Matrix

Description
Forms an mk x mk Kronecker Product Permutation Matrix

Usage
KPPM(m, k)

Arguments
m integer
k integer

Value
Matrix

Author(s)
Jarrod Hadfield <j.hadfield@ed.ac.uk>

References
Schott, J.R.(2003) Journal of Multivariate Analysis 87 (1) 177-190
krzanowski.test

**Example**

```
KPPP(2,3)
```

---

**krzanowski.test**  
*Krzanzowski’s Comparison of Subspaces*

---

**Description**

Calculates statistics of Krzanowski’s comparison of subspaces.

**Usage**

```
krzanowski.test(CA, CB, vecsA, vecsB, corr = FALSE, ...)
```

**Arguments**

- **CA**: Matrix A
- **CB**: Matrix B
- **vecsA**: Vector of integers indexing the eigenvectors determining the subspace of A
- **vecsB**: Vector of integers indexing the eigenvectors determining the subspace of B
- **corr**: logical; if TRUE the variances of A and B are standardised
- **...**: further arguments to be passed

**Value**

- **sumofS**: metric for overall similarity with 0 indicting no similarity and a value of length(vecsA) for identical subspaces
- **angles**: angle in degrees between each best matched pair of vectors
- **bisector**: vector that lies between each best matched pair of vectors

**Author(s)**

Jarrod Hadfield <j.hadfield@ed.ac.uk>

**References**


**Examples**

```
CA <- rIW(diag(5), 10, n=1)
CB <- rIW(diag(5), 10, n=1)
krzanowski.test(CA, CB, vecsA=1:2, vecsB=1:2)
krzanowski.test(CA, CA, vecsA=1:2, vecsB=1:2)
```
kunif

Central Moments of a Uniform Distribution

Description
Returns the central moments of a uniform distribution

Usage
kunif(min, max, k)

Arguments
min, max lower and upper limits of the distribution. Must be finite.
k k central moment, must be even

Value
kth central moment

Author(s)
Jarrod Hadfield <j.hadfield@ed.ac.uk>

See Also
dunif

Examples
kunif(-1,1,4)
y<-runif(1000,-1,1)
mean((y-mean(y))^4)

list2bdiag
Forms the direct sum from a list of matrices

Description
Forms a block-diagonal matrix from a list of matrices

Usage
list2bdiag(x)
Arguments

\(x\)

list of square matrices

Value

matrix

Author(s)

Jarrod Hadfield <j.hadfield@ed.ac.uk>

Examples

\[
M <- \text{list(rIW(diag(3), 10), rIW(diag(2), 10))}
\text{list2bdiag}(M)
\]

**MCMCglmm**  
*Multivariate Generalised Linear Mixed Models*

Description

Markov chain Monte Carlo Sampler for Multivariate Generalised Linear Mixed Models with special emphasis on correlated random effects arising from pedigrees and phylogenies (Hadfield 2010). Please read the course notes: vignette("CourseNotes", "MCMCglmm") or the overview vignette("Overview", "MCMCglmm")

Usage

\[
\text{MCMCglmm(fixed, random=\text{NULL}, rcov=}\sim\text{units, family}="\text{gaussian}", \text{mev=}\text{NULL,}
\text{data, start=}\text{NULL, prior=}\text{NULL, tune=}\text{NULL, pedigree=}\text{NULL, nodes}="\text{ALL},
\text{scale=}\text{TRUE, nitt=}13000, \text{thin=}10, \text{burnin=}3000, \text{pr=}\text{FALSE,}
\text{pl=}\text{FALSE, verbose=}\text{TRUE, DIC=}\text{TRUE, singular.ok=}\text{FALSE, saveX=}\text{TRUE,}
\text{saveZ=}\text{TRUE, saveXL=}\text{TRUE, slice=}\text{FALSE, ginverse=}\text{NULL})
\]

Arguments

**fixed**

Formula for the fixed effects, multiple responses are passed as a matrix using \text{cbind}

**random**

Formula for the random effects. Multiple random terms can be passed using the \text{+} operator, and in the most general case each random term has the form \text{variance.function(formula):linking.function(random.terms)}. Currently, the only variance.functions available are \text{idv, idh, us, cor[]} and \text{ante[]}. \text{idv} fits a constant variance across all components in formula. Both \text{idh} and \text{us} fit different variances across each component in formula, but us will also fit the covariances. \text{cor} fixes the variances along the diagonal to one and \text{corgh} fixes the variances along the diagonal to those specified in the prior. \text{corh} allows correlation submatrices. \text{ante[]} fits ante-dependence structures of different order (e.g. ante1, ante2), and the number can be prefixed by a \text{c} to hold all
regression coefficients of the same order equal. The number can also be suffixed by a v to hold all innovation variances equal (e.g. antec2v has 3 parameters). The formula can contain both factors and numeric terms (i.e. random regression) although it should be noted that the intercept term is suppressed. The (co)variances are the (co)variances of the random terms effects. Currently, the only linking functions available are mm and str. mm fits a multimembership model where multiple random terms are separated by the + operator. str allows covariances to exist between multiple random terms that are also separated by the + operator. In both cases the levels of all multiple random terms have to be the same. For simpler models the variance.function(formula) and linking.function(random.terms) can be omitted and the model syntax has the simpler form ~random1+random2+. ... There are two reserved variables: units which index rows of the response variable and trait which index columns of the response variable.

rcov

formula for residual covariance structure. This has to be set up so that each data point is associated with a unique residual. For example a multi-response model might have the R-structure defined by ~us(trait):units

family

optional character vector of trait distributions. Currently, "gaussian", "poisson", "categorical", "multinomial", "ordinal", "threshold", "exponential", "geometric", "cengaussian", "cenpoisson", "cenexponential", "zipoisson", "ztpoisson", "hupoisson", "zibinomial" and "threshold" are supported, where the prefix "cen" means censored, the prefix "zi" means zero inflated, the prefix "za" means zero altered, the prefix "zt" means zero truncated and the prefix "hu" means hurdle. If NULL, data needs to contain a family column.

mev

optional vector of measurement error variances for each data point for random effect meta-analysis.

data

data.frame

start

optional list having 4 possible elements: R (R-structure) G (G-structure) and liab (latent variables or liabilities) should contain the starting values where G itself is also a list with as many elements as random effect components. The fourth element QUASI should be logical: if TRUE starting latent variables are obtained heuristically, if FALSE then they are sampled from a Z-distribution

prior

optional list of prior specifications having 3 possible elements: R (R-structure) G (G-structure) and B (fixed effects). B is a list containing the expected value (mu) and a (co)variance matrix (V) representing the strength of belief: the defaults are B$mu=0 and B$V=1*1e+10, where where I is an identity matrix of appropriate dimension. The priors for the variance structures (R and G) are lists with the expected (co)variances (V) and degree of belief parameter (nu) for the inverse-Wishart, and also the mean vector (alpha.mu) and covariance matrix (alpha.V) for the redundant working parameters. The defaults are nu=0, V=I, alpha.mu=0, and alpha.V=0. When alpha.V is non-zero, parameter expanded algorithms are used.

tune

optional (co)variance matrix defining the proposal distribution for the latent variables. If NULL an adaptive algorithm is used which ceases to adapt once the burn-in phase has finished.
pedigree ordered pedigree with 3 columns id, dam and sire or a phylo object. This argument is retained for back compatibility - see ginv argument for a more general formulation.

nodes pedigree/phylogeny nodes to be estimated. The default, "ALL" estimates effects for all individuals in a pedigree or nodes in a phylogeny (including ancestral nodes). For phylogenies "TIPS" estimates effects for the tips only, and for pedigrees a vector of ids can be passed to nodes specifying the subset of individuals for which animal effects are estimated. Note that all analyses are equivalent if omitted nodes have missing data but by absorbing these nodes the chain max mix better. However, the algorithm may be less numerically stable and may iterate slower, especially for large phylogenies.

scale logical: should the phylogeny (needs to be ultrametric) be scaled to unit length (distance from root to tip)?

nitt number of MCMC iterations

thin thinning interval

burnin burnin

pr logical: should the posterior distribution of random effects be saved?

pl logical: should the posterior distribution of latent variables be saved?

verbose logical: if TRUE MH diagnostics are printed to screen

DIC logical: if TRUE deviance and deviance information criterion are calculated

singular.ok logical: if FALSE linear dependencies in the fixed effects are removed. if TRUE they are left in an estimated, although all information comes form the prior

saveX logical: save fixed effect design matrix

saveZ logical: save random effect design matrix

saveXL logical: save structural parameter design matrix

slice logical: should slice sampling be used? Only applicable for binary traits with independent residuals

ginv a list of sparse inverse matrices (A−1) that are proportional to the covariance structure of the random effects. The names of the matrices should correspond to columns in data that are associated with the random term. All levels of the random term should appear as rownames for the matrices.

Value

Sol Posterior Distribution of MME solutions, including fixed effects

VCV Posterior Distribution of (co)variance matrices

CP Posterior Distribution of cut-points from an ordinal model

Liab Posterior Distribution of latent variables

Fixed list: fixed formula and number of fixed effects

Random list: random formula, dimensions of each covariance matrix, number of levels per covariance matrix, and term in random formula to which each covariance belongs
MCMCglmm

Residual list: residual formula, dimensions of each covariance matrix, number of levels per covariance matrix, and term in residual formula to which each covariance belongs

Deviance deviance -2*\log(p(y|...))

DIC deviance information criterion

X sparse fixed effect design matrix

Z sparse random effect design matrix

XL sparse structural parameter design matrix

error.term residual term for each datum

family distribution of each datum

Tune (co)variance matrix of the proposal distribution for the latent variables

Author(s)

Jarrod Hadfield <j.hadfield@ed.ac.uk>

References


See Also

mcmc

Examples

# Example 1: univariate Gaussian model with standard random effect
data(PlodiaPO)
model1<-MCMCglmm(PO~1, random=~FSfamily, data=PlodiaPO, verbose=FALSE)
summary(model1)

# Example 2: univariate Gaussian model with phylogenetically correlated # random effect
data(bird.families)

phylo.effect<-rbv(bird.families, 1, nodes="TIPS")
phenotype<-phylo.effect+rnorm(dim(phylo.effect)[1], 0, 1)

# simulate phylogenetic and residual effects with unit variance
test.data<-data.frame(phenotype=phenotype, taxon=row.names(phenotype))
mult.memb

Design Matrices for Multiple Membership Models

Description

Forms design matrices for multiple membership models

Usage

mult.memb(formula)

Arguments

formula

Details

Currently `mult.memb` can only usefully be used inside an `idv` variance function. The formula usually contains several factors that have the same factor levels.

Value

design matrix

Author(s)

Jarrod Hadfield <j.hadfield@ed.ac.uk>

Examples

```r
fac1 <- factor(sample(letters[1:3], 5, TRUE), levels = letters[1:3])
fac2 <- factor(sample(letters[1:3], 5, TRUE), levels = letters[1:3])
cbind(fac1, fac2)
mult.memb(~fac1 + fac2)
```
**Description**

Forms design matrix for path analyses that involve paths within residual blocks.

**Usage**

```r
path(cause=NULL, effect=NULL, k)
```

**Arguments**

- `cause` integer; index of predictor ‘trait’ within residual block
- `effect` integer; index of response ‘trait’ within residual block
- `k` integer; dimension of residual block

**Value**

design matrix

**Note**

For more general path analytic models see `sir` which allows paths to exist between responses that are not in the same residual block. However, `sir` does not handle non-Gaussian or missing responses. Note that path models involving non-Gaussian data are defined on the link scale which may not always be appropriate.

**Author(s)**

Jarrod Hadfield <j.hadfield@ed.ac.uk>

**See Also**

- `sir`

**Examples**

```r
path(1, 2, 2)
```
PlodiaPO

*Phenoloxidase measures on caterpillars of the Indian meal moth.*

**Description**

Phenoloxidase measures on caterpillars of the Indian meal moth (*Plodia interpunctella*).

**Usage**

PlodiaPO

**Format**

a data frame with 511 rows and 3 columns, with variables indicating full-sib family (F5family), phenoloxidase measures (PO), and plate (plate). PO has undergone a Box-Cox power transformation of 0.141

**Source**

Tidbury H & Boots M (2007) University of Sheffield

**See Also**

PlodiaR, PlodiaRB

PlodiaR

*Resistance of Indian meal moth caterpillars to the granulosis virus PiGV.*

**Description**

Resistance of Indian meal moth (*Plodia interpunctella*) caterpillars to the granulosis virus PiGV.

**Usage**

PlodiaR

**Format**

a data frame with 50 rows and 5 columns, with variables indicating full-sib family (F5family), date of egg laying (date_laid) and assaying (date_Ass), and the number of individuals from the family that were experimentally infected with the virus Infected and the number of those that pupated Pupated. These full-sib family identifiers also relate to the full-sib family identifiers in PlodiaPO

**Source**

Tidbury H & Boots M (2007) University of Sheffield
PlodiaRB

See Also

PlodiaRB, PlodiaPO

PlodiaRB

Resistance (as a binary trait) of Indian meal moth caterpillars to the granulosis virus PiGV.

Description

Resistance (as a binary trait) of Indian meal moth (Plodia interpunctella) caterpillars to the granulosis virus PiGV.

Usage

PlodiaRB

Format

a data frame with 784 rows and 4 columns, with variables indicating full-sib family (FSfamily), date of egg laying (date_laid) and assaying (date_ass), and a binary variable indicating whether an individual was resistant (Pupated) to an experimental infection of the virus. These data are identical to those in the data.frame PlodiaR except each family-level binomial variable has been expanded into a binary variable for each individual.

Source

Tidbury H & Boots M (2007) University of Sheffield

See Also

PlodiaR, PlodiaPO

plot.MCMCglmm

Plots MCMC chains from MCMCglmm using plot.mcmc

Description

plot method for class "MCMCglmm".

Usage

## S3 method for class 'MCMCglmm'
plot(x, random=FALSE, ...)

plotsubspace

Arguments

- **x**: an object of class "MCMCglmm"
- **random**: logical; should saved random effects be plotted
- **...**: Further arguments to be passed

Author(s)

Jarrod Hadfield <j.hadfield@ed.ac.uk>

See Also

plot.mcmc, MCMCglmm

Description

Represents covariance matrices as 3-d ellipsoids using the rgl package. Covariance matrices of dimension greater than 3 are plotted on the subspace defined by the first three eigenvectors.

Usage

```r
plotsubspace(CA, CB=NULL, corr = FALSE, shadeCA = TRUE,
             shadeCB = TRUE, axes.lab = FALSE, ...)
```

Arguments

- **CA**: Matrix
- **CB**: Optional second matrix
- **corr**: If TRUE the covariance matrices are transformed into correlation matrices
- **shadeCA**: If TRUE the ellipsoid is solid, if FALSE the ellipsoid is wireframe
- **shadeCB**: If TRUE the ellipsoid is solid, if FALSE the ellipsoid is wireframe
- **axes.lab**: If TRUE the axes are labelled with the eigenvectors
- **...**: further arguments to be passed

Details

The matrix CA is always red, and the matrix CB if given is always blue. The subspace is defined by the first three eigenvectors of CA, and the percentage of variance for each matrix along these three dimensions is given in the plot title.

Author(s)

Jarrod Hadfield <j.hadfield@ed.ac.uk> with code taken from the rgl package
### posterior.ante

#### Description

Posterior distribution of ante-dependence parameters

#### Usage

```r
posterior.ante(x, k=1)
```

#### Arguments

- `x` mcmc object of (co)variances stacked column-wise
- `k` order of the ante-dependence structure

#### Value

posterior ante-dependence parameters (innovation variances followed by regression coefficients)

#### Author(s)

Jarrod Hadfield <j.hadfield@ed.ac.uk>

#### See Also

- `posterior.cor`, `posterior.evals`, `posterior.inverse`

#### Examples

```r
v <- rIW(diag(2), 10, n=1000)
plot(posterior.ante(mcmc(v), 1))
```
**posterior.cor**  
*Transforms posterior distribution of covariances into correlations*

**Description**
Transforms posterior distribution of covariances into correlations

**Usage**

```r
posterior.cor(x)
```

**Arguments**

- `x` mcmc object of (co)variances stacked column-wise

**Value**
posterior correlation matrices

**Author(s)**
Jarrod Hadfield <j.hadfield@ed.ac.uk>

**See Also**

- `posterior.evals`, `posterior.inverse`, `posterior.ante`

**Examples**

```r
v <- rIW(diag(2), 3, n=1000)  
hist(posterior.cor(mcmc(v))[,2])
```

---

**posterior.evals**  
*Posterior distribution of eigenvalues*

**Description**
Posterior distribution of eigenvalues

**Usage**

```r
posterior.evals(x)
```

**Arguments**

- `x` mcmc object of (co)variances stacked column-wise
Value
posterior eigenvalues

Author(s)
Jarrod Hadfield <j.hadfield@ed.ac.uk>

See Also
posterior.cor, posterior.inverse, posterior.ante

Examples
v <- rIW(diag(2), 3, n=1000)
hist(posterior.evals(mcmc(v))[,2])

##

posterior.inverse  Posterior distribution of matrix inverse

Description
Posterior distribution of matrix inverse

Usage
posterior.inverse(x)

Arguments
x mcmc object of (co)variances stacked column-wise

Value
posterior of inverse (co)variance matrices

Author(s)
Jarrod Hadfield <j.hadfield@ed.ac.uk>

See Also
posterior.cor, posterior.evals, posterior.ante

Examples
v <- rIW(diag(2), 3, n=1000)
plot(posterior.inverse(mcmc(v)))
posterior.mode

Estimates the marginal parameter modes using kernel density estimation

Description

Estimates the marginal parameter modes using kernel density estimation

Usage

posterior.mode(x, adjust=0.1, ...)

Arguments

x : mcmc object

adjust : numeric, passed to density to adjust the bandwidth of the kernel density

... other arguments to be passed

Value

modes of the kernel density estimates

Author(s)

Jarrod Hadfield <j.hadfield@ed.ac.uk>

See Also

density

Examples

v<-rIW(as.matrix(1),10, n=1000)
hist(v)
abline(v=posterior.mode(mcmc(v)), col="red")
predict.MCMCglmm

Predict method for GLMMs fitted with MCMCglmm

Description

Predicted values for GLMMs fitted with MCMCglmm

Usage

```r
## S3 method for class 'MCMCglmm'
predict(object, newdata=NULL, marginal=object$Random$formula,
  type="response", interval="none", level=0.95, it=NULL,
  posterior="all", verbose=FALSE, ...)
```

Arguments

- `object`: an object of class "MCMCglmm"
- `newdata`: An optional data frame in which to look for variables with which to predict
- `marginal`: formula defining random effects to be marginalised
- `type`: character; either "terms" (link scale) or "response" (data scale)
- `interval`: character; either "none", "confidence" or "prediction"
- `level`: A numeric scalar in the interval (0,1) giving the target probability content of the intervals.
- `it`: integer; optional, MCMC iteration on which predictions should be based
- `posterior`: character; if it is NULL should marginal posterior predictions be calculated ("all"), or should they be made conditional on the marginal posterior means ("mean") of the parameters, the posterior modes ("mode"), or a random draw from the posterior ("distribution").
- `verbose`: logical; if TRUE, warnings are issued with newdata when the original model has fixed effects that do not appear in newdata and/or newdata has random effects not present in the original model.
- `...`: Further arguments to be passed

Value

Expectation and credible interval

Author(s)

Jarrod Hadfield <j.hadfield@ed.ac.uk>

See Also

MCMCglmm
prunePed  

Pedigree pruning

Description

Creates a subset of a pedigree by retaining the ancestors of a specified subset of individuals.

Usage

prunePed(pedigree, keep, make.base=FALSE)

Arguments

- **pedigree**: pedigree with id in column 1 dam in column 2 and sire in column 3.
- **keep**: individuals in pedigree for which the ancestors should be retained.
- **make.base**: logical: should ancestors that do not provide additional information be discarded?

Value

- subsetted pedigree

Note

If the individuals in `keep` are the only phenotyped individuals for some analysis then some non-phenotyped individuals can often be discarded if they are not responsible for pedigree links between phenotyped individuals. In the simplest case (`make.base=FALSE`) all ancestors of phenotyped individuals will be retained, although further pruning may be possible using `make.base=TRUE`. In this case all pedigree links that do not connect phenotyped individuals are discarded resulting in some individuals becoming part of the base population. In terms of variance component and fixed effect estimation pruning the pedigree should have no impact on the target posterior distribution, although convergence and mixing may be better because there is less missing data.

Author(s)

Jarrod Hadfield <j.hadfield@ed.ac.uk> + Michael Morrissey
Ptensor

Tensor of Sample (Mixed) Central Moments

Description
Forms a tensor of sample (mixed) central moments

Usage
Ptensor(x, k)

Arguments
x matrix; traits in columns samples in rows
k kth central moment

Value
tensor

Author(s)
Jarrod Hadfield <j.hadfield@ed.ac.uk>

Examples
n<-1000
y<-matrix(rnorm(n), n/2, 2)
Ptensor(y, 2)
cov(y)*((n-1)/n)

rbv
Random Generation of MVN Breeding Values and Phylogenetic Effects

Description
Random Generation of MVN Breeding Values and Phylogenetic Effects

Usage
rbv(pedigree, G, nodes="ALL", scale=TRUE, ggroups=NULL, gmeans=NULL)
residuals.MCMCglmm

Arguments

pedigree ordered pedigree with 3 columns id, dam and sire or a phylo object.

G (co)variance matrix

nodes effects for pedigree/phylogeny nodes to be returned. The default, nodes="ALL" returns effects for all individuals in a pedigree or nodes in a phylogeny (including ancestral nodes). For phylogenies nodes="TIPS" returns effects for the tips only, and for pedigrees a vector of ids can be passed to nodes specifying the subset of individuals for which animal effects are returned.

scale logical: should a phylogeny (needs to be ultrametric) be scaled to unit length (distance from root to tip)?

ggroups optional; vector of genetic groups

gmeans matrix of mean breeding value for genetic groups (rows) by traits (columns)

Value

matrix of breeding values/phylogenetic effects

Author(s)

Jarrod Hadfield <j.hadfield@ed.ac.uk>

Examples

data(bird.families)
bv<-rbv(bird.families, diag(2))

residuals.MCMCglmm Residuals form a GLMM fitted with MCMCglmm

Description

residuals method for class "MCMCglmm".

Usage

## S3 method for class 'MCMCglmm'
residuals(object, type = c("deviance", "pearson", "working", "response", "partial"), ...

Arguments

object an object of class "MCMCglmm"

type the type of residuals which should be returned. The alternatives are: "deviance" (default), "pearson", "working", "response", and "partial".

... Further arguments to be passed
Value

vector of residuals

Author(s)

Jarrod Hadfield <j.hadfield@ed.ac.uk>

See Also

residuals, MCMCglmm

---

**rIW**

*Random Generation from the Conditional Inverse Wishart Distribution*

**Description**

Samples from the inverse Wishart distribution, with the possibility of conditioning on a diagonal submatrix

**Usage**

```
rIW(V, nu, fix=NULL, n=1, CM=NULL)
```

**Arguments**

- `V` Expected (co)variance matrix as `nu` tends to infinity
- `nu` degrees of freedom
- `fix` optional integer indexing the partition to be conditioned on
- `n` integer: number of samples to be drawn
- `CM` matrix: optional matrix to condition on. If not given, and `fix=NULL`, `V_22` is conditioned on

**Details**

If \( W^{-1} \) is a draw from the inverse Wishart, `fix` indexes the diagonal element of \( W^{-1} \) which partitions \( W^{-1} \) into 4 submatrices. `fix` indexes the upper left corner of the lower diagonal matrix and it is this matrix that is conditioned on.

For example partitioning \( W^{-1} \) such that

\[
W^{-1} = \begin{bmatrix}
W^{-1}_{11} & W^{-1}_{12} \\
W^{-1}_{21} & W^{-1}_{22}
\end{bmatrix}
\]

`fix` indexes the upper left corner of \( W^{-1}_{22} \). If \( CM=NULL \) then \( W^{-1}_{22} \) is fixed at \( CM \), otherwise \( W^{-1}_{22} \) is fixed at \( V_{22} \). For example, if \( \dim(V)=4 \) and \( fix=2 \) then \( W^{-1}_{11} \) is a 1X1 matrix and \( W^{-1}_{22} \) is a 3X3 matrix.
Value

if \( n = 1 \) a matrix equal in dimension to \( V \), if \( n > 1 \) a matrix of dimension \( n \times \text{length}(V) \)

Note

In versions of MCMCglmm >1.10 the arguments to \( \text{riw} \) have changed so that they are more intuitive in the context of MCMCglmm. Following the notation of Wikipedia (http://en.wikipedia.org/wiki/Inverse-Wishart_distribution) the inverse scale matrix \( \Psi = (V \ast \nu) \). In earlier versions of MCMCglmm (<1.11) \( \Psi = V^{-1} \). Although the old parameterisation is consistent with the \( \text{riwish} \) function in MCMCpack and the \( \text{rwishart} \) function in bayesm it is inconsistent with the prior definition for MCMCglmm. The following pieces of code are sampling from the same distributions:

\[
\begin{align*}
\text{riwish}(\nu, \nu \ast V) & \quad \text{from MCMCpack} \\
\text{rwishart}(\nu, \text{solve}(\nu \ast V)) & \quad \text{from bayesm} \\
\text{riw}(\nu, \text{solve}(\nu \ast V)) & \quad \text{from MCMCglmm} < 1.11 \\
\text{riw}(V, \nu) & \quad \text{from MCMCglmm} \geq 1.11
\end{align*}
\]

Author(s)

Jarrod Hadfield <j.hadfield@ed.ac.uk>

References

Korsgaard, I.R. et. al. 1999 Genetics Selection Evolution 31 (2) 177:181

See Also

\( \text{rwishart, rwish} \)

Examples

\[
\begin{align*}
\nu & \leftarrow 10 \\
V & \leftarrow \text{diag}(4) \\
\text{riw}(V, \nu, \text{fix}=2)
\end{align*}
\]

\[
\begin{align*}
\text{rtcmvnorn} & \quad \text{Random Generation from a Truncated Conditional Normal Distribution}
\end{align*}
\]

Description

Samples from the Truncated Conditional Normal Distribution

Usage

\[
\text{rtcmvnorn}(n = 1, \text{mean} = 0, V = 1, x = 0, \text{keep}=1, \text{lower} = -\text{Inf}, \text{upper} = \text{Inf})
\]
rtnorm

Random Generation from a Truncated Normal Distribution

Arguments

n integer: number of samples to be drawn
mean vector of means
V covariance matrix
x vector of observations to condition on
keep element of x to be sampled
lower left truncation point
upper right truncation point

Value
vector

Author(s)
Jarrod Hadfield <j.hadfield@ed.ac.uk>

Examples

```r
par(mfrow=c(2,1))
V1<-cbind(c(1,0.5), c(0.5,1))
x1<-rtcvmnorm(10000, c(0,0), V=V1, c(0,2), keep=1, lower=-1, upper=1)
x2<-rtnorm(10000, 0, 1, lower=-1, upper=1)
plot(density(x1), main="Correlated conditioning observation")
lines(density(x2), col="red")
# densities of conditional (black) and unconditional (red) distribution
# when the two variables are correlated (r=0.5)

V2<-diag(2)
x3<-rtcvmnorm(10000, c(0,0), V=V2, c(0,2), keep=1, lower=-1, upper=1)
x4<-rtnorm(10000, 0, 1, lower=-1, upper=1)
plot(density(x3), main="Uncorrelated conditioning observation")
lines(density(x4), col="red")
# densities of conditional (black) and unconditional (red) distribution
# when the two variables are uncorrelated (r=0)
```

Description

Samples from the Truncated Normal Distribution

Usage

```r
rtnorm(n = 1, mean = 0, sd = 1, lower = -Inf, upper = Inf)
```
Arguments

- `n` integer: number of samples to be drawn
- `mean` vector of means
- `sd` vector of standard deviations
- `lower` left truncation point
- `upper` right truncation point

Value

vector

Author(s)

Jarrod Hadfield <j.hadfield@ed.ac.uk>

References


See Also

rtnorm

Examples

```r
hist(rtnorm(100, lower=-1, upper=1))
```

Description

Simulated response vectors for GLMMs fitted with MCMCglmm

Usage

```r
## S3 method for class 'MCMCglmm'
simulate(object, nsim = 1, seed = NULL, newdata=NULL, marginal = object$Random$formula, type = "response", it=NULL, posterior = "all", verbose=FALSE, ...)
```
sir

Arguments

object
an object of class "MCMCglmm"

nsim
number of response vectors to simulate. Defaults to 1.

seed
Either NULL or an integer that will be used in a call to set.seed before sim-
ulating the response vectors. The default, NULL will not change the random
generator state.

newdata
An optional data frame for which to simulate new observations

marginal
formula defining random effects to be marginalised

type
character; either "terms" (link scale) or "response" (data scale)

it
integer; optional, MCMC iteration on which predictions should be based

posterior
character; if it is NULL should the response vector be simulated using the marginal
posterior means ("mean") of the parameters, or the posterior modes ("mode"),
random draws from the posterior with replacement ("distribution") or without
replacement ("all")

verbose
logical; if TRUE, warnings are issued with newdata when the original model has
fixed effects that do not appear in newdata and/or newdata has random effects
not present in the original model.

...
Further arguments to be passed

Value
A matrix (with nsim columns) of simulated response vectors

Author(s)
Jarrod Hadfield <j.hadfield@ed.ac.uk>

See Also
MCMCglmm

Design Matrix for Simultaneous and Recursive Relationships between Responses

Description
Forms design matrix for simultaneous and recursive relationships between responses

Usage
sir(formula1=NULL, formula2=NULL)
sm2asreml

Converting sparseMatrix to asreml's giv format

- **Arguments**
  - `formula1` formula
  - `formula2` formula

- **Value**
  - design matrix

- **Author(s)**
  - Jarrod Hadfield <j.hadfield@ed.ac.uk>

- **Examples**
  - `fac1 <- factor(sample(letters[1:3], 5, TRUE), levels = letters[1:3])`
  - `fac2 <- factor(sample(letters[1:3], 5, TRUE), levels = letters[1:3])`
  - `cbind(fac1, fac2)`
  - `sir(~fac1, ~fac2)`

---

**Description**

Converting sparseMatrix to asreml's giv format: row-ordered, upper triangle sparse matrix.

**Usage**

```
sm2asreml(A=NULL, rownames=NULL)
```

**Arguments**

- `A` sparseMatrix
- `rownames` rownames of A

**Value**

- data.frame: if A was formed from a pedigree equivalent to giv format returned by `asreml.Ainverse`

**Author(s)**

 Jarrod Hadfield <j.hadfield@ed.ac.uk>

**See Also**

- `inverseA`
**spl**

*Orthogonal Spline Design Matrix*

**Description**

Orthogonal Spline Design Matrix

**Usage**

```
spl(x, k=10, knots=NULL, type="LRTP")
```

**Arguments**

- `x` a numeric covariate
- `k` integer, defines knot points at the 1:k/(k+1) quantiles of `x`
- `knots` vector of knot points
- `type` type of spline - currently only low-rank thin-plate ("LRTP") are implemented

**Value**

Design matrix post-multiplied by the inverse square root of the penalty matrix

**Author(s)**

Jarrod Hadfield <j.hadfield@ed.ac.uk>

**Examples**

```R
# Not run:
x<-rnorm(100)
y<-x^2+cos(x)-x+0.2*x^3+rnorm(100)
plot(y~x)
lines((x^2+cos(x)-x+0.2*x^3)[order(x)]~sort(x))

dat<-data.frame(y=y, x=x)

m1<-MCMCglmm(y~x, random=~idv(spl(x)), data=dat, pr=TRUE, verbose=FALSE) # penalised smoother
m2<-MCMCglmm(y~x+spl(x), data=dat, verbose=FALSE)                         # non-penalised

pred1<-cbind(m1$m1X,m1$m2X)*colMeans(m1$m1Sol))@x
pred2<-cbind(m2$m2X)*colMeans(m2$m2Sol))@x

lines(pred1[order(x)]~sort(x), col="red")
```
lines(pred2[order(x)]~sort(x), col="green")

m1$DIC-mean(m1$Deviance) # effective number of parameters < 13
m2$DIC-mean(m2$Deviance) # effective number of parameters ~ 13

## End(Not run)

### SShorns

**Horn type and genders of Soay Sheep**

**Description**

Horn type and genders of Soay Sheep *Ovis aires*

**Usage**

```r
btdata
```

**Format**

a data frame with 666 rows and 3 columns, with individual identifier (id), horn type (horn) and gender (sex).

**References**

Clutton-Brock T., Pemberton, J. Eds. 2004 Soay Sheep: Dynamics & Selection in an Island Population

### summary.MCMCglmm

**Summarising GLMM Fits from MCMCglmm**

**Description**

summary method for class "MCMCglmm". The returned object is suitable for printing with the print.summary.MCMCglmm method.

**Usage**

```r
## S3 method for class 'MCMCglmm'
summary(object, random=FALSE, ...)
```

**Arguments**

- **object**: an object of class "MCMCglmm"
- **random**: logical: should the random effects be summarised
- **...**: Further arguments to be passed
Value

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DIC</td>
<td>Deviance Information Criterion</td>
</tr>
<tr>
<td>fixed.formula</td>
<td>model formula for the fixed terms</td>
</tr>
<tr>
<td>random.formula</td>
<td>model formula for the random terms</td>
</tr>
<tr>
<td>residual.formula</td>
<td>model formula for the residual terms</td>
</tr>
<tr>
<td>solutions</td>
<td>posterior mean, 95% HPD interval, MCMC p-values and effective sample size of fixed (and random) effects</td>
</tr>
<tr>
<td>Gcovariances</td>
<td>posterior mean, 95% HPD interval and effective sample size of random effect (co)variance components</td>
</tr>
<tr>
<td>Gterms</td>
<td>indexes random effect (co)variances by the component terms defined in the random formula</td>
</tr>
<tr>
<td>Rcovariances</td>
<td>posterior mean, 95% HPD interval and effective sample size of residual (co)variance components</td>
</tr>
<tr>
<td>Rterms</td>
<td>indexes residuals (co)variances by the component terms defined in the rcov formula</td>
</tr>
<tr>
<td>csats</td>
<td>chain length, burn-in and thinning interval</td>
</tr>
<tr>
<td>cutpoints</td>
<td>posterior mean, 95% HPD interval and effective sample size of cut-points from an ordinal model</td>
</tr>
</tbody>
</table>

Author(s)

Jarrod Hadfield <j.hadfield@ed.ac.uk>

See Also

mcmcglmm

Tri2M

Lower/Upper Triangle Elements of a Matrix

Description

Lower/Upper triangle elements of a matrix or forms a matrix from a vector of lower/upper triangle elements

Usage

Tri2M(x, lower.tri = TRUE, reverse = TRUE, diag = TRUE)
Arguments

- **x**: Matrix or vector
- **lower.tri**: If `x` is a matrix then the lower triangle (TRUE) or upper triangle (FALSE) elements (including diagonal elements) are returned. If `x` is a vector a matrix is formed under the assumption that `x` are the lower triangle (TRUE) or upper triangle (FALSE) elements.
- **reverse**: logical: if TRUE a symmetric matrix is formed, if FALSE the remaining triangle is left as zeros.
- **diag**: logical: if TRUE diagonal elements are included.

Value

numeric or matrix

Author(s)

Jarrod Hadfield <j.hadfield@ed.ac.uk>

Examples

```r
M <- rW(diag(3), 10)
x <- Tri2M(M)
x
Tri2M(x, reverse = TRUE)
Tri2M(x, reverse = FALSE)
```
Index

*Topic array
  commutation, 7
  Dtensor, 10
  evalDtensor, 11
  inverseA, 13
  KPPM, 15
  Ptensor, 33

*Topic datasets
  BTdata, 5
  BTped, 5
  PlodiaPO, 24
  PlodiaR, 24
  PlodiaRB, 25
  SShorns, 42

*Topic distribution
  dcmvnorm, 7
  gelmanNprior, 12
  knorm, 14
  kunif, 17
  posterior.ante, 27
  posterior.cor, 28
  posterior.evals, 28
  posterior.inverse, 29
  posterior.mode, 30
  rbv, 33
  rIW, 35
  rtcmvnorm, 36
  rtnorm, 37

*Topic hplot
  plot.MCMCglmm, 25
  plotSubspace, 26

*Topic manip
  at.level, 3
  at.set, 4
  list2bdiaq, 17
  mult.memb, 22
  path, 23
  prunePed, 32
  sir, 39

  sm2asreml, 40
  spl, 41
  Tri2M, 43

*Topic math
  Dtensor, 10
  evalDtensor, 11

*Topic models
  buildV, 6
  MCMCglmm, 18
  predict.MCMCglmm, 31
  residuals.MCMCglmm, 34
  simulate.MCMCglmm, 38
  summary.MCMCglmm, 42

*Topic multivariate
  Ddivergence, 8
  krzanowski.test, 16

*Topic symbolmath
  Dexpressions, 9

  at.level, 3, 4
  at.set, 4, 4

  BTdata, 5
  BTped, 5, 6
  buildV, 6

  commutation, 7

  D, 11
  data.frame, 12
  dcmvnorm, 7
  Ddivergence, 8
  density, 30
  Dexpressions, 9, 11
  dnorm, 15
  Dtensor, 10, 10, 11
  dunif, 17

  evalDtensor, 11, 11

  formula, 12, 18, 19
gelman.prior, 12
inverseA, 13
knorm, 14
KPPM, 15
krzanowski.test, 16
kunif, 17
list2bdia, 17
mcmc, 21
MCMCglmm, 6, 18, 26, 31, 35, 36, 39, 43
MCMCglmm-package, 2
mult.memb, 22
path, 23
PlodiaPO, 24, 25
PlodiaR, 24, 24, 25
PlodiaRB, 24, 25, 25
plot.MCMCglmm, 25
plotsubspace, 26
posterior.ante, 27, 28, 29
posterior.cor, 27, 28, 29
posterior.evals, 27, 28, 28, 29
posterior.inverse, 27–29, 29
posterior.mode, 30
predict.MCMCglmm, 31
print.MCMCglmm (summary.MCMCglmm), 42
print.summary.MCMCglmm
  (summary.MCMCglmm), 42
prunePed, 32
Ptensor, 33
rbv, 33
residuals, 35
residuals.MCMCglmm, 34
rgl, 27
rIW, 35
riwish, 36
rtcmvnorm, 36
rtnorm, 37, 38
rwish, 36
rwishart, 36
simulate.MCMCglmm, 38
sir, 23, 39
sm2asreml, 40
spl, 41
SSHorns, 42