Package 'spaMM'

November 9, 2014

Type Package

Title Mixed Models, Particularly Spatial GLMMs

Encoding UTF-8

Version 1.4.1

Date 2014-11-08

Maintainer François Rousset <francois.rousset@univ-montp2.fr>

Imports stats, graphics, Matrix, MASS, lpSolveAPI (>= 5.5.0.14),proxy, geometry, Rcpp (>= 0.11.0), nlme

LinkingTo Rcpp, RcppEigen

Suggests maps, testthat, lme4

Depends R (>= 3.0.0)

NeedsCompilation yes

Description

Implements a collection of functions for inference in mixed models. It was developed in particular for GLMMs with spatial correlations, but also fits models with non-Gaussian random effects (e.g., Beta Binomial, or negative-binomial mixed models). Heteroskedasticity can further be fitted by a linear model. The algorithms are currently various Laplace approximations methods for ML or REML, in particular h-likelihood and penalized-likelihood methods.

License CeCILL-2

URL http://www.r-project.org,http://kimura.univ-montp2.fr/~rousset/spaMM.htm

Author François Rousset [aut, cre, cph],Jean-Baptiste Ferdy [aut, cph]

Repository CRAN

Date/Publication 2014-11-09 00:46:19

R topics documented:

`arabidopsis` *Arabidopsis genetic and climatic data*

Description

For 948 "accessions" from European Arabidopsis thaliana populations, this data set merges the genotypic information at four single nucleotide polmorphisms (SNP) putatively involved in adaptation to climate (Fournier-Level et al, 2011, Table 1), with 13 climatic variables from Hancock et al. (2011).

Usage

```
data(arabidopsis)
```

Format

The data frame includes 948 observations on the following variables:

pos1046738, pos5510910, pos6235221, pos8132698 Genotypes at four SNP loci

LAT latitude

LONG longitude

seasonal, tempWarmest, tempColdest, preciWettest, preciDriest, preciCV, PAR_SPRING, growingL, conseqCold, conseqFrFree, RelHumidSp, dayLSp, aridity Thirteen climatic variables. See Hancock et al. (2011) for details about these variables.

Details

The response is binary so "HLmethod="PQL/L"" seems warranted (see Rousset and Ferdy, 2014).

Source

The data were retrieved from http://bergelson.uchicago.edu/regmap-data/climate-genome-scan on 22 February 2013.

References

Fournier-Level A, Korte A., Cooper M. D., Nordborg M., Schmitt J., Wilczek AM (2011). A map of local adaptation in Arabidopsis thaliana. Science 334: 86-89.

Hancock, A. M., Brachi, B., Faure, N., Horton, M. W., Jarymowycz, L. B., Sperone, F. G., Toomajian, C., Roux, F., and Bergelson, J. 2011. Adaptation to climate across the Arabidopsis thaliana genome, Science 334: 83-86.

Rousset F., Ferdy, J.-B. (2014) Testing environmental and genetic effects in the presence of spatial autocorrelation. Ecography, 37: 781-790. http://dx.doi.org/10.1111/ecog.00566

Examples

```
data(arabidopsis)
## takes about ~45 s. on a laptop.
## Not run:
HLCor(cbind(pos1046738,1-pos1046738)~seasonal+Matern(1|LAT+LONG),
            ranPars=list(rho=0.129,lambda=4.28,nu=0.291),
            family=binomial(),HLmethod="PQL/L",data=arabidopsis)

## End(Not run)

## the above ranPars are deduced from an hours-long analysis:
## Not run:
verylong <- corrHLfit(cbind(pos1046738,1-pos1046738)~seasonal+Matern(1|LAT+LONG),
              family=binomial(),HLmethod="PQL/L",data=arabidopsis)
summary(verylong)
```

```
## End(Not run)
```

`blackcap` *Genetic polymorphism in relation to migration in the blackcap*

Description

This data set is extracted from a study of genetic polymorphisms potentially associated to migration behaviour in the blackcap (Sylvia atricapilla). Across different populations in Europe and Africa, the average migration behaviour was found to correlate with average allele size (dependent on the number of repeats of a small DNA motif) at the locus ADCYAP1, encoding a neuropeptide.

Usage

```
data(blackcap)
```

Format

The data frame includes 14 observations on the following variables:

latitude latitude, indeed.

longitude longitude, indeed.

migStatus migration status as determined by Mueller et al, from 0 (resident populations) to 2.5 (long-distance migratory populations)

means Mean allele sizes in each population

pos Numerical index for the populations

Details

Migration status was coded as : pure resident populations as '0', resident populations with some migratory restlessness as '0.5', partial migratory populations as '1', completely migratory populations migrating short-distances as '1.5', intermediate-distance migratory populations as '2' and distinct long-distance migratory populations as '2.5'.

Source

Data from Mueller et al. (2011), including supplementary material retrieved from http://rspb.royalsocietypublishing.org/content/suppl/2011/02/11/rspb.2010.2567.DC1.

References

Mueller, J. C., Pulido, F., and Kempenaers, B. 2011. Identification of a gene associated with avian migratory behaviour, Proc. Roy. Soc. (Lond.) B 278, 2848-2856.

Examples

```
## see 'corrHLfit' and 'fixedLRT' for examples involving these data
```

confint.HLfit *Confidence intervals for fixed-effect parameters.*

Description

This computes confidence intervals for a given fixed effect parameter, based on the p_v-based approximation of the profile likelihood ratio for this parameter. The profiling is other all other fitted parameters: other fixed effects, as well as variances of random effects and spatial correlations if these were fitted.

Usage

```
## S3 method for class 'HLfit'
confint(object, parm, level=0.95, verbose=TRUE,...)
```

Arguments

object	An object of class HLfit, such as return object of HLfit, HLCor or corrHLfit calls;
parm	The name of a parameter to be fitted, or its position in the the object's $fixef vector. Valid names are those of the object's $fixef;
level	The coverage of the interval;
verbose	whether to print the interval or not. As the function returns its more extensive results invisibly, this printing is the only visible output;
...	Additional arguments (maybe not used, but conforming to the generic definition of confint).

Value

A list including the confidence interval for the target parameter, and the fits lowerfit and upperfit giving the profile fits at the confidence bounds. This is returned invisibly.

Examples

```
data(wafers)
wfit <- HLfit(y ~X1+(1|batch),family=Gamma(log),data=wafers,HLmethod="ML")
confint(wfit,"X1")
```

corMatern *Matern Correlation Structure as a corSpatial object*

Description

This implements the Matérn correlation structure for use with lme or glmmPQL. Usage is as for others corSpatial objects such as corGaus or corExp, except that the Matérn family has an additional parameter. This function was defined for comparing results obtained with corrHLfit to those produced by lme and glmmmPQL. There are problems in fitting (G)LMMs in the latter way, so it is not a recommended practice.

Usage

```
corMatern(value = c(1, 0.5), form = ~1, nugget = FALSE, nuScaled = FALSE,
          metric = c("euclidean", "maximum", "manhattan"), fixed = FALSE)
```

Arguments

value	An optional vector of parameter values, with serves as initial values or as fixed values depending on the fixed argument. It has either two or three elements, depending on the nugget argument. If nugget is FALSE, value should have two elements, corresponding to the "range" and the "smoothness" ν of the Matérn correlation structure. If value has zero length, the default is a range of 90% of the minimum distance and a smoothness of 0.5 (exponential correlation). **Warning**: the range parameter used in corSpatial objects is the inverse of the scale parameter used in Matern.corr and thus they have opposite meaning despite both being denoted ρ elsewhere in this package or in nlme literature. If nugget is TRUE, meaning that a nugget effect is present, value can contain two or three elements, the first two as above, the third being the "nugget effect" (one minus the correlation between two observations taken arbitrarily close together). If value has length zero or two, the nugget defaults to 0.1. The range and smoothness must be greater than zero and the nugget must be between zero and one.
form	(Pasted from corSpatial) a one sided formula of the form ~ S1+...+Sp, or ~ S1+...+Sp | g, specifying spatial covariates S1 through Sp and, optionally, a grouping factor g. When a grouping factor is present in form, the correlation structure is assumed to apply only to observations within the same grouping level; observations with different grouping levels are assumed to be uncorrelated. Defaults to ~ 1, which corresponds to using the order of the observations in the data as a covariate, and no groups.
nugget	an optional logical value indicating whether a nugget effect is present. Defaults to FALSE.
nuScaled	If nuScaled is set to TRUE the "range" parameter ρ is divided by $2\sqrt{\nu}$. With this option and for large values of ν, corMatern reproduces the calculation of corGaus. Defaults to FALSE, in which case the function compares to corGaus with range parameter $2(\sqrt{\nu})\rho$ when ν is large.

`metric` (Pasted from corSpatial) an optional character string specifying the distance metric to be used. The currently available options are `"euclidean"` for the root sum-of-squares of distances; `"maximum"` for the maximum difference; and `"manhattan"` for the sum of the absolute differences. Partial matching of arguments is used, so only the first three characters need to be provided. Defaults to `"euclidean"`.

`fixed` an optional logical value indicating whether the coefficients should be allowed to vary in the optimization, or kept fixed at their initial value. Defaults to `FALSE`, in which case the coefficients are allowed to vary.

Details

This function is a constructor for the `corMatern` class, representing a Matérn spatial correlation structure. See `Matern.corr` for details on the Matérn family.

Value

an object of class `corMatern`, also inheriting from class `corSpatial`, representing a Matérn spatial correlation structure.

References

Mixed-Effects Models in S and S-PLUS, José C. Pinheiro and Douglas M. Bates, Statistics and Computing Series, Springer-Verlag, New York, NY, 2000.

See Also

`glmmPQL`, `lme`

Examples

```
## LMM

data(blackcap)
blackcapD <-cbind(blackcap,dummy=1) ## obscure, isn't it?
require(nlme)
## With method= 'ML' in lme, The correlated random effect is described
##  as a correlated residual error and no extra residual variance is fitted:
lme(fixed = migStatus ~ means, data = blackcapD, random = ~ 1 | dummy,
    correlation = corMatern(form = ~ longitude+latitude  | dummy),
    method = "ML")


## Binomial GLMM
## Not run:
## ~54s. on a laptop
data(Loaloa)
LoaloaD <-cbind(Loaloa,dummy=1)
require(MASS)
glmmPQL(fixed =cbind(npos,ntot-npos)~elev1+elev2+elev3+elev4+maxNDVI1+seNDVI,
        data = LoaloaD, random = ~ 1 | dummy,family=binomial,
```

```
        correlation = corMatern(form = ~ longitude+latitude | dummy))

## End(Not run)
```

`corrHLfit` *Fits a mixed model, typically a spatial GLMM.*

Description

`corrHLfit` performs the joint estimation of correlation parameters, fixed effect and dispersion parameters.

Usage

```
corrHLfit(formula,data,init.corrHLfit=list(),init.HLfit=list(),ranFix=list(),
        lower=list(),upper=list(),trace=list(file=NULL,append=T),objective="p_bv",
        rho.mapping,control.corrHLfit=list(),processed=NULL,family=gaussian(),...)
```

Arguments

`formula`	Either a linear model `formula` (as handled by various fitting functions) or a `predictor`, i.e. a formula with attributes (see `Predictor` and examples below). See Details in `spaMM` for allowed terms in the formula.
`data`	A data frame containing the variables in the response and the model formula.
`init.corrHLfit`	An optional list of initial values for correlation and/or dispersion parameters, e.g. `list(rho=1,nu=1,lambda=1,phi=1)` where `rho` and `nu` are parameters of the Matérn family, and `lambda` and `phi` are dispersion parameters (see Details in `spaMM` for the meaning of these parameters). All are optional, but giving values for a dispersion parameter changes the ways it is estimated (see Details).
`init.HLfit`	See identically named `HLfit` argument.
`ranFix`	A list similar to `initpar`, but specifying fixed values of the parameters not estimated.
`lower`	An optional list of values of parameters specified through `initpar`, used as lower values in calls to `optim`. See Details for default values.
`upper`	Same as `lower`, but upper values.
`trace`	Not for normal use. If trace is of the form `trace=list(file=<filename>,append=F)`, some trace information is written in the file 'filename'. This file is written over by each new call of `corrHLfit` unless `append=T`. Information is written for each `HLCor` call. It includes the `APHLs`, and all given fixed random effect parameters in the HLCor call. Information about the variables is printed at the end of the file, but may be slightly inaccurate (some amount of guesswork is expected from users venturing into `trace`). The arguments of all HLCor calls are saved in RData files (if this is felt inappropriate, then using `spaMM.options(TRACE.UNLINK=TRUE)` will keep only the latest HLCor call).

objective	The objective function maximized for estimation of parameters by optim. Either "p_bv" for restricted likelihood or "p_v" for marginal likelihood.
rho.mapping	A set of indices controlling which elements of the rho scale vector scales which dimension(s) of the space in which (spatial) correlation matrices of random effects are computed. See same argument in make.scaled.dist for details and examples.
control.corrHLfit	Only for development purposes, not documented.
processed	For programming purposes, not documented.
family	Either a family or a multi value.
...	Optional arguments passed to HLCor, HLfit or designL.from.Corr, for example the distMatrix argument of HLCor. Arguments that do not fit within these functions are detected and a warning is issued.

Details

Under the Matérn correlation model, corrHLfit typically performs a optimization over the ρ and ν parameters, with maximum possible values as set by spaMM.options.

By default corrHLfit will estimate correlation parameters by maximizing the objective value returned by HLCor calls wherein the dispersion parameters are estimated jointly with fixed effects for given correlation parameters. If dispersion parameters are specified in init.corrHLfit, they will also be estimated by maximizing the objective value, and HLCor calls will not estimate them jointly with fixed effects. This means that in general the fixed effect estimates may vary depending on init.corrHLfit when any form of REML correction is applied.

Correctly using corrHLfit for likelihood ratio tests of fixed effects may then by tricky. It is safe to perform full ML fits of all parameters (using objective="p_v",HLmethod="ML") for such tests (see Examples). The higher level function fixedLRT is a safe interface for likelihood ratio tests using some form of REML estimation in corrHLfit.

attr(<fitted object>,"optimInfo")$lower and ...$upper gives the lower and upper bounds for optimization of correlation parameters. These are the default values if the user did not provide explicit values. For the adjacency model, the default values are the inverse of the maximum and minimum eigenvalues of the adjMatrix. For the Matérn model, these values are not so easily summarized. They are intended to cover the range of values for which there is statistical information to distinguish among them.

Value

The return value of an HLCor call, with additional attributes. The HLCor call is evaluated at the estimated correlation parameter values. These values are included in the return object as its $corrPars member. The attributes added by corrHLfit include the original call of the function (corrHLfitcall), and various informations about the optimization call within corrHLfit.

See Also

See more examples on data set Loaloa. See fixedLRT for likelihood ratio tests.

Examples

```
#### example with an adjacency matrix (autoregressive model)
data(scotlip)
corrHLfit(cases~I(prop.ag/10) +adjacency(1|gridcode)+offset(log(scotlip$expec)),
               data=scotlip,family=poisson(),
               adjMatrix=Nmatrix)


#### Examples with Matérn correlations
## A likelihood ratio test based on the ML fits of a full and of a null model.

data(blackcap)
fullfit <- corrHLfit(migStatus ~ means+ Matern(1|latitude+longitude),data=blackcap,
                     HLmethod="ML") ## takes ~ 6s
summary(fullfit)
nullfit <- corrHLfit(migStatus ~ 1 + Matern(1|latitude+longitude),data=blackcap,
                     HLmethod="ML") ## takes ~ 3s
summary(nullfit)
## p-value:
1-pchisq(2*(logLik(fullfit)-logLik(nullfit)),df=1)


## see data set Loaloa for additional examples
```

`designL.from.Corr` *Computation of "square root" of correlation matrix*

Description

This function is not usually directly called by users, but arguments may be passed to it through higher-level calls (see Examples). For given correlation matrix **C**, it computes a "design matrix" **L** such that **C** = **L** * t(**L**). `chol` (Cholesky factorization) is a fast method for this computation, but it is not robust numerically and may even return an error, in which cases other methods (`eigen` or `svd`) are used. Note that these methods differ by much more than simply numerical accuracy, and in particular that this will affect the simulation of samples for the parametric bootstrap implemented in `fixedLRT`.

Usage

```
designL.from.Corr(m,try.chol=TRUE,try.eigen=FALSE,threshold=1e-06,debug=FALSE,SVDfix=1/10)
```

Arguments

`m`	The matrix which 'root' is to be computed.
`try.chol`	If `try.chol=TRUE`, the Cholesky factorization will be tried.
`try.eigen`	The default behavior is to try `chol`, and use `svd` if chol fails. If `try.eigen=TRUE`, the `eigen` factorization will be tried before `svd`. `eigen` is a compromise between speed and accuracy, but in our experience it may *hang* so by default it is not tried.

threshold	A correction threshold for low eigenvalues is the case and eigensystem or singular-value decomposition are used.
debug	Not documented, only for development purposes.
SVDfix	A solution to failures of svd: see Details.

Details

Singular value decomposition (SVD) of a matrix $\mathbf{M}$ using svd may encounter problems resulting in "error code 1 from Lapack routine 'dgesdd"' (cf. unhelpful discussions on R forums). This can be circumvented by computing the SVD of $(1-x)\mathbf{I}+x\mathbf{M}$ and deducing the singular values of $\mathbf{M}$ in a trivial way. The x value is here provided by the SVDfix argument.

svd errors have occurred for correlation matrices that were close to the identity matrix except for a few large non-diagonal elements. Such matrices may occur in particular for low ν, high ρ, and if some samples are spatially close. Then, an alternative fix to the svd problem may be to restrict the ν and/or ρ ranges, using the lower and upper arguments of corrHLfit, although one should make sure that this has no bearing on the inferences.

Value

The "square root of the input matrix". Its rows and columns are labelled according to the columns of the original matrix.

Examples

```
## Not run:
## try.chol argument passed to designL.from.Corr
## through the '...' argument of higher-level functions
## such as HLCor, corrHLfit, fixedLRT:
data(scotlip)
HLCor(cases~I(prop.ag/10) +adjacency(1|gridcode)+offset(log(scotlip$expec)),
      ranPars=list(rho=0.174),adjMatrix=Nmatrix,family=poisson(),
      data=scotlip,try.chol=FALSE)

## End(Not run)
```

---

extractors *Functions to extract various components of a fit*

Description

logLik extracts the log-likelihood (exact or approximated). fitted extracts fitted values (see fitted.values). fixef extracts the fixed effects coefficients, β. ranef extracts the predicted random effects, u. vcov returns the variance-covariance matrix of the fixed-effects coefficients. predictionCoeffs precomputes coefficients for prediction (see predict for an example)

Usage

```
## S3 method for class 'HLfit'
logLik(object,REML,...)
## S3 method for class 'HLfit'
fitted(object,...)
## S3 method for class 'HLfit'
fixef(object,...)

## S3 method for class 'HLfit'
ranef(object,...)
## S3 method for class 'HLfit'
vcov(object,...)
predictionCoeffs(object)
```

Arguments

object	The return object of an HLfit or similar function.
REML	If TRUE, the function returns the log restricted likelihood (exact or approximated).
...	Other arguments that may be needed by some method.

Value

All return values are numeric (for logLik) or vectors (most cases) or matrices (for vcov). ranef returns a vector with attributes, which inherits from class ranef which has its own (undocumented) print method.

Examples

```
data(wafers)
m1 <- HLfit(y ~X1+X2+(1|batch),
          resid.formula = ~ 1 ,data=wafers,HLmethod="ML")
fixef(m1)
vcov(m1)
ranef(m1)
## see 'predict' for a example with predictionCoeffs
```

---

fixedLRT *Likelihood ratio test of fixed effects.*

Description

fixedLRT performs a likelihood ratio (LR) test between two models, the "full" and the "null" models, currently differing only in their fixed effects. Parametric bootstrap p-values can be computed, either using the raw bootstrap distribution of the likelihood ratio, or a a bootstrap estimate of the Bartlett correction of the LR statistic. This function differ from LRT in its arguments (model fits for LRT, but all arguments required to fit the models for fixedLRT), and in the format of its return value.

Usage

```
fixedLRT(null.formula,formula,data,HLmethod,REMLformula=NULL,boot.repl=0,
         control=list(),control.boot=list(),...)
```

Arguments

null.formula	Either a formula (as in glm) or a predictor (see Predictor) for the null model.
formula	Either a formula or a predictor for the full model.
data	A data frame containing the variables in the model.
HLmethod	A method to fit the full and null models. See the identically-named HLfit argument for background information about such methods. The two most meaningful values of HLmethod in fixedLRT calls are: 'ML' for an LRT based on ML fits (generally recommended); and 'PQL/L' for an LRT based on PQL/L fits (recommended for spatial binary data). Also feasible, but more tricky, and not really recommended (see Rousset and Ferdy, 2014), is 'REML'. This will perform an LRT based on two REML fits of the data, *both* of which use the same conditional (or "restricted") likelihood of residuals for estimating dispersion parameters λ and ϕ (see REMLformula argument). Further, REML will not be effective on a given dispersion parameter if a non-trivial init.corrHLfit value is provided for this parameter.
REMLformula	a formula specifying the fixed effects which design matrix is used in the REML correction for the estimation of dispersion parameters, if these are estimated by REML. This formula is by default that for the *full* model.
boot.repl	the number of bootstrap replicates.
control	A set of control parameters for the fits of the data, mostly for development purposes. However, if an initial value is provided for a dispersion parameter, a better one may be sought if further control=list(prefits=TRUE) (the effect appears small, however).
control.boot	Same as control, but for the fits of the bootstrap replicates. Again, the option control.boot=list(prefits=TRUE) may yield a small improvement in the fits, at the expense of more computation time.
...	Further arguments passed to or from other methods; in particular, additional arguments passed to corrHLfit, including mandatory ones such as data and those ultimately passed to designL.from.Corr. With respect to the latter, note that try.chol affects the simulation of samples for the parametric bootstrap, and although ultimate differences in performance may be small, try.chol=FALSE may be slightly better.

Details

Comparison of REML fits is a priori not suitable for performing likelihood ratio tests. Nevertheless, it is possible to contrive them for testing purposes (Wehlam & Thompson 1997). This function generalizes some of Wehlam & Thompson's methods to GLMMs.

See Details in LRT for details of the bootstrap procedures.

Value

An object of class `fixedLRT`, actually a list with as-yet unstable format, but here with typical elements (depending on the options)

`fullfit`	the HLfit object for the full model;
`nullfit`	the HLfit object for the null model;
`LRTori`	A likelihood ratio chi-square statistic
`LRTprof`	Another likelihood ratio chi-square statistic, after a profiling step, if any.
`df`	the number of degrees of freedom of the test.
`trace.info`	Information on various steps of the computation.
`bootreps`	A table of fitted likelihoods for bootstrap replicates.
`meanbootLRT`	The mean likelihood ratio chi-square statistic for boostrap replicates.

References

Rousset F., Ferdy, J.-B. (2014) Testing environmental and genetic effects in the presence of spatial autocorrelation. Ecography, 37: 781-790. http://dx.doi.org/10.1111/ecog.00566

Welham, S. J., and Thompson, R. (1997) Likelihood ratio tests for fixed model terms using residual maximum likelihood, J. R. Stat. Soc. B 59, 701-714.

See Also

See also `corrHLfit` and `LRT`.

Examples

```
data(blackcap)
## result comparable to the corrHLfit examples based on blackcap
fixedLRT(null.formula=migStatus ~ 1 + Matern(1|latitude+longitude),
       formula=migStatus ~ means + Matern(1|latitude+longitude),
       HLmethod='ML',data=blackcap)

## Not run:
## longer version with bootstrap
fixedLRT(null.formula=migStatus ~ 1 + Matern(1|latitude+longitude),
       formula=migStatus ~ means + Matern(1|latitude+longitude),
       HLmethod='ML',data=blackcap, boot.repl=100)

## End(Not run)
```

HLCor *Fits a (spatially) correlated mixed model, for given correlation parameters*

Description

A convenient interface for HLfit, constructing the correlation matrix of random effects from the arguments, then estimating fixed effects and dispersion parameters using HLfit.

Usage

```
HLCor(formula,ranPars,data,distMatrix,uniqueGeo,adjMatrix,
      corrMatrix,verbose=c(warn=TRUE,trace=FALSE,summary=FALSE),...)
```

Arguments

formula	A predictor, i.e. a formula with attributes (see Predictor), or possibly simply a simple formula if an offset is not required.
ranPars	A list of values for correlation parameters (some of which are mandatory), and possibly also dispersion parameters (optional, but passed to HLfit if present). By default, the Matern.corr model is assumed, where the correlation parameters are rho (scale parameter(s)), nu (smoothness parameter), and (optional) Nugget. The rho parameter can itself be a vector with different values for different geographic coordinates. If corr.model is "adjacency", the only correlation parameter is a scalar rho. The dispersion parameters are the random effect variance, lambda, and the residual variance, phi.
data	The data frame to be analyzed.
distMatrix	A distance matrix between geographic locations, forwarded to Matern.corr
uniqueGeo	A matrix of non-redundant geographic locations. This is useful if the rho parameter is a vector with different values for different geographic coordinates, in which case a scaled distance matrix has to be reconstructed from uniqueGeo and rho.
adjMatrix	An adjacency matrix, used if a random effect of the form y ~ adjacency(1|<location index>) is present. Its rows and columns must be ordered as increasing values of the levels of the geographic location index specifying the spatial random effect. For example, if the model formula is y ~ adjacency(1|geo.loc) and <data>$geo.loc is 2,4,3,1,... the first row/column of the matrix refers to the fourth row of the data.
corrMatrix	A arbitrary (valid) correlation matrix, used if a random effect term of the form corrMatrix(1|<stuff>) is present. Each row corresponds to levels of a variable <stuff>. This allows to analyze non-spatial model by giving for example a matrix of genetic correlations. Internally, corrMatrix will determine the "**L**" matrix, which may be modified by additional matrices (see Details in Predictor).

verbose	A vector of booleans. trace controls various diagnostic (possibly messy) messages about the iterations. summary controls whether a summary of the fit is called by HLfit. warn is for programming purposes and best ignored.
...	Further parameters passed to HLfit or to designL.from.corr.

Details

The correlation matrix for random effects can be specified by various combination of formula terms and other arguments (see Examples):

Basic Matérn model Matern(1|<...>) with ranPars argument, using the spatial coordinates in <...>. This will construct a correlation matrix according to the Matérn correlation function (see Matern.corr);

Matérn model with given distance matrix Matern(1|<...>) with distMatrix and ranPars argument;

Given correlation matrix corrMatrix(1|<...>) with corrMatrix argument;

CAR model with given adjacency matrix adjacency(1|<...>) with adjMatrix and ranPars argument. This will construct the correlation matrix $(\mathbf{I}-\rho$ adjMatrix$)^{-1}$ for a conditional autoregressive (CAR) model. This was implemented for comparison purposes as it is widely used, but is not really recommended for irregular lattices (see Wall, 2004 and Martellosio, 2012 for some insights on the implications of autoregressive models). Accordingly, tricks that allow faster analysis of CAR models were not implemented.

Value

The return value of an HLfit call, with the following additional attributes:

HLCorcall	the HLCor call
info.uniqueGeo	Unique geographic locations.

References

Wall M.M. (2004) A close look at the spatial structure implied by the CAR and SAR models: Journal of Statistical Planning and Inference 121: 311-324.

Martellosio, F. (2012) The correlation structure of spatial autoregressions, Econometric Theory 28, 1373-1391.

See Also

Matern.corr, HLfit, corrHLfit

Examples

```
#### Matérn correlation using only the Matern() syntax
data(Loaloa)
HLCor(cbind(npos,ntot-npos)~elev1+elev2+elev3+elev4+maxNDVI1+seNDVI
              +Matern(1|longitude+latitude),data=Loaloa,
```

```
      family=binomial(),ranPars=list(nu=0.5,rho=1/0.7)) ## takes ~ 6s


#### Using a corrMatrix (not necessarily Matérn)
data(blackcap)
## Here we manually reconstruct the correlation matrix
##  of the ML fit produced by corrHLfit:
MLcorMat <- Matern.corr(proxy::dist(blackcap[,c("latitude","longitude")]),
                        nu=0.6285603,rho=0.0544659)
HLCor(migStatus ~ means+ corrMatrix(1|latitude+longitude),data=blackcap,
      corrMatrix=MLcorMat,HLmethod="ML")

#### Matérn correlation using a distMatrix
MLdistMat <- as.matrix(proxy::dist(blackcap[,c("latitude","longitude")]))
HLCor(migStatus ~ means+ Matern(1|latitude+longitude),data=blackcap,
      distMatrix=MLdistMat,HLmethod="ML",ranPars=list(nu=0.6285603,rho=0.0544659))


#### example with an adjacency matrix (autoregressive model)
data(scotlip)
HLCor(cases~I(prop.ag/10) +adjacency(1|gridcode)+offset(log(scotlip$expec)),
      ranPars=list(rho=0.174),
      adjMatrix=Nmatrix,family=poisson(),data=scotlip)
```

HLfit *Fit mixed models with given correlation matrix*

Description

This fonction fits GLMMs as well as some hierarchical generalized linear models (HGLM; Lee and Nelder 2001). HLfit fits both fixed effects parameters, and dispersion parameters i.e. the variance of the random effects and the variance of the residual error. The linear predictor is of the standard form `offset+ X beta + Z v`, where X is the design matrix of fixed effects and Z is a design matrix of random effects. The function also handles a linear predictor (with only fixed effects) for the residual variance.

Usage

```
HLfit(formula, data, family = gaussian(), rand.family = gaussian(),
      resid.formula = ~1, REMLformula = NULL,
      verbose=c(warn=TRUE,trace=FALSE,summary=FALSE),
      HLmethod = "HL(1,1)", control.HLfit = list(), init.HLfit = list(),
     ranFix = list(), etaFix = list(), prior.weights = rep(1, nobs), processed = NULL)
## see 'rand.family' argument for inverse.Gamma
## see 'control.HLfit' argument for AIC
```

Arguments

formula	A formula; or a predictor, i.e. a formula with attributes created by Predictor, if design matrices for random effects have to be provided. See Details in spaMM for allowed terms in the formula (except spatial ones).
data	A data frame containing the variables named in the model formula.
family	A family object describing the distribution of the response variable. Possible values include the gaussian, poisson, binomial and Gamma families. Possible links are identity, log, inverse, logit, probit, and cloglog. Possible combinations of family and link are those allowed by the family objects (see family).
rand.family	A family object describing the distribution of the random effect, or a list of family objects for different random effects (see Examples). Possible options are gaussian(), Gamma(log), Beta(logit), inverse.Gamma(-1/mu), and inverse.Gamma(log). For discussion of these alternatives see Lee and Nelder 2001 or Lee et al. 2006, p. 178-. Here the family gives the distribution of a random effect u and the link gives v as function of u (see Details). If there are several random effects and only one family is given, this family holds for all random effects.
resid.formula	A formula (without left-hand side) for the variance phi of the residual error. Currently can only contain fixed effects, including an offset. A log link is assumed by default, but an identity link can be used (see control.HLfit).
REMLformula	A model formula that allows the estimation of dispersion parameters, and computation of restricted likelihood (p_bv) under a model different from the predictor formula. For example, if only random effects are included in REMLformula, an ML fit is performed and p_bv equals the marginal likelihood (or its approximation), p_v. This ML fit can be performed more simply by setting HLmethod="ML" and leaving REMLformula at its default NULL value.
verbose	A vector of booleans. trace controls various diagnostic (possibly messy) messages about the iterations. summary controls whether a summary of the fit is called by HLfit. warn is for programming purposes and best ignored.
HLmethod	Allowed values are "REML", "ML", "EQL-" and "EQL+" for all models; "PQL" (="REPQL") and "PQL/L" for GLMMs only; and (only for those curious to experiment) expressions of the form "HL(<...>)", "ML(<...>)" and "RE(<...>)". HL and RE are equivalent. The default behaviour is RE(1,1), which **by default performs REML** (standard REML for LMMs, an extended definition for other models). But it can also perform non-standard forms of REML (indeed including ML), depending on the REMLformula given. EQL stands for the EQL method of Lee and Nelder (2001). The '+' version includes the d v/ d tau correction described p. 997 of that paper, and the '-' version ignores it. PQL can be seen as the version of EQL- for GLMMs. PQL/L is PQL without the leverage corrections that define REML estimation of random-effect parameters. For **GLMs**, the default is still REML. For binomial and Poisson GLMs, this cannot be distinguished from an exact ML fit. Note that the glm function performs an EQL analysis, which will differ from the REML (and ML) one for Gamma GLMs.

See Details for the more general syntax.

`control.HLfit` A list of parameters controlling (1) AIC computation; and (2) the fitting algorithms.

`AIC=TRUE` provides the corrected AIC of Ha et al. 2007 [i.e., AIC(D*) in their eq. 7], but the default is `FALSE` to (modestly) encourage users to think twice before applying model selection automatically, which is no better although more fashionable than misuses of simple null hypothesis testing. Also, alternative procedures for model choice can be considered (e.g. Cox and Donnelly, 2011, p. 130-131).

`resid.family` allows one to change the link for modeling of residual variance ϕ, which is `"log"` by default. The family is always Gamma, so the non-default possible values of `resid.family` are `Gamma(identity)` or `Gamma(inverse)`. Only the default value ensures that the fitted ϕ is positive.

Controls for the fitting algorithms should be ignored in routine use. They are

`conv.threshold`, a convergence threshold for the iterative algorithm, which controls whether linear predictor terms (fixed effects and inferred random effects), and dispersion parameter estimates have converged. Defaults to 1e-05;

`iter.mean.dispFix`, the number of iterations of the iterative algorithm for coefficients of the linear predictor, if no dispersion parameters are estimated by the iterative algorithm. Defaults to 200;

`iter.mean.dispVar`, the number of iterations of the iterative algorithm for coefficients of the linear predictor, if some dispersion parameter(s) is estimated by the iterative algorithm. Defaults to 50;

`max.iter`, the number of iterations of the iterative algorithm for joint estimation of dispersion parameters and of coefficients of the linear predictor. Defaults to 200. This is typically much more than necessary, unless there is little information to separately estimate λ and ϕ parameters.

`init.HLfit` A list of initial values for the iterative algorithm, with possible elements of the list are `fixef` for fixed effect estimates (beta), `v_h` for random effects vector **v** in the linear predictor, `lambda` for the parameter determining the variance of random effects u as drawn from the `rand.family` distribution `phi` for the residual variance. However, this argument can be ignored in routine use.

`ranFix` A list of fixed values of random effect parameters, with possible elements `lambda`, and also `phi` for gaussian and Gamma HGLMs. Inhibits the estimation of these parameters.

`etaFix` A list of fixed values of the coefficients of the linear predictor, with possible elements `beta` and `v_h` or `u_h`.

`prior.weights` An optional vector of prior weights as in `glm`. This fits the data to a model with residual variance `phi/prior.weights`, so that increasing the weights by a constant factor *f* will yield estimates of `phi` also increased by *f*. Note that this is what `glm` does, but that some other widely used packages can behave differently (and obscurely).

`processed` A list of preprocessed arguments, for programming purposes only (as in `corrHLfit` code).

Details

Fitting methods: Many approximations for likelihood have been defined to fit mixed models (e.g. Noh and Lee (2007) for some overview), and this function only considers a subset of them, but it adds a new complication in terms of REML methods. For example, PQL as originally defined by Breslow and Clayton uses REML to estimate dispersion parameters, but this function allows one to use ML instead. Moreover, it allows some non-standard specification of the model formula that determines the conditional distribution used in REML.

In the more general syntax for `HLmethod`, used as e.g. `HLmethod="RE(1,1)"` the first '1' means that a first order Laplace approximation to the likelihood is used to estimate fixed effects (a '0' would instead mean that the h likelihood is used as the objective function). The second '1' means that a first order Laplace approximation to the likelihood or restricted likelihood is used to estimate dispersion parameters, including the dv/d tau term specifically discussed by Lee & Nelder 2001, p. 997 (a '0' would instead mean that these terms are ignored).

It is possible to enforce the EQL approximation for estimation of dispersion parameter by adding a third index with value 0. `"HL(0,1,0)"` is Lee & Nelder's (2001) method, i.e. `"EQL+"`.

For a Gamma GLM with log link, ML and EQL results will differ in their phi estimates, and the EQL estimate will match that from the `glm` function.

Random effects are constructed in several steps. first, a vector **u** of independent and identically distributed (iid) random effects is drawn from some distribution; second, a transformation v=f(u) is applied to each element (this defines **v** which elements are still iid); third, correlated random effects are obtained as **Lv** where **L** is the "square root" of a correlation matrix (this may be meaningful only for Gaussian random effects). Finally, a matrix **Z** (or sometimes **ZA**, see `Predictor`) allows to specify how the correlated random effects affect the response values. In particular, **Z** is the identity matrix if there is a single observation (response) for each location, but otherwise its elements z_{ji} are 1 for the jth observation in the ith location. The design matrix for **v** is then of the form **ZL**.

The specification of the random effects u and v handles the following cases:

Gaussian with zero mean, unit variance, and identity link;

Beta Beta-distributed random effects, where ($u \tilde{} B(1/(2\lambda), 1/(2\lambda))$ (mean=1/2, var= $\lambda/[4(1+\lambda)]$), with logit link `v=logit(u)`)

Gamma Gamma-distributed random effects, where $u \tilde{} \Gamma(1/\lambda, \lambda)$, so u has mean 1 and variance λ.

Inverse-Gamma inverse-Gamma distributed random effects, where $u \tilde{}$ inverse-Gamma$(1+1/\lambda, 1/\lambda)$ (mean=1, var=$\lambda/(1-\lambda)$), with `v=log(u)` or `v=-1/u`.

The standard errors reported may sometimes be misleading. For each set of parameters among $beta$, $lambda$, and phi parameters these are computed assuming that the other parameters are known without error. This is why they are labelled `Cond. SE` (conditional standard error). This is most uninformative in the unusual case where $lambda$ and phi are not separately estimable parameters. Further, the SEs for $lambda$ and phi are rough approximations as discussed in particular by Smyth et al. (2001; V_1 method).

Value

An object of class `HLfit`, actually a list with many elements, several of which represent input arguments. Some elements may be undocumented.

A few extractor functions are available (see extractors), and should be used as far as possible as they should be backward-compatible from version 1.4 onwards, while the structure of the return object may still evolve (to further deal with this, the return object includes a version tag as element spaMM.version). The following information will be useful for extracting further elements of the object.

Elements describing the fit include:

fixef	The fixed effects coefficients, β (returned by the fixef function)
ranef	The random effects u (returned by the ranef function)
fv	Fitted values (μ =<inverse-link>(η)) of the response variable (returned by the fitted function)
APHLs	A list with usually four elements, the conditional likelihood, the h-likelihood, and the two adjusted profile h-likelihoods: the (approximate) marginal likelihood p_v and the (approximate) restricted likelihood p_bv (the latter two available through the logLik function).
beta_cov	Covariance matrix of β estimates
v_h	The random effects on the linear scale, v
phi	The residual variance ϕ
phi.object	A possibly more complex object describing ϕ
lambda	The random effects (u) variance λ
lambda.object	A possibly more complex object describing λ
eta	Fitted values on the linear scale (including the predicted random effects)

It may also be worth checking the following elements

HL	A set of indices that characterize the approximations used for likelihood.
warnings	A list of warnings for events that may have occurred during the fit.

Additional information about the fit is contained in

ZALmatrix	The design matrix for random effects (see Details).

and in (possibly reformatted) input arguments

predictor	The response predictor.
data	The input data.
family	a family object corresponding to the family input
rand.family	a family object corresponding to the rand.family input
y	the response vector; for binomial data, the frequency response
formulaDisp	Model formula for dispersion response
resid.family	Family for estimation of dispersion from residuals
X	The design matrix for fixed effects
ranFix	A ranFix input
corrPars	Additional information on correlation parameters, not necessarily used by HLfit itself but in upper calling functions such as HLCor or corrHLfit
models	Additional information on model structure for η, λ and ϕ
weights	(binomial data only) the binomial denominators

References

Breslow, NE, Clayton, DG. (1993). Approximate Inference in Generalized Linear Mixed Models. Journal of the American Statistical Association 88, 9-25.

Cox, D. R. and Donnelly C. A. (2011) Principles of Applied Statistics. Cambridge Univ. Press.

Ha, I. D., Lee, Y. and MacKenzie, G. (2007) Model selection for multi-component frailty models. Statistics in Medicine 26: 4790-4807.

Lee, Y., Nelder, J. A. (2001) Hierarchical generalised linear models: A synthesis of generalised linear models, random-effect models and structured dispersions. Biometrika 88, 987-1006.

Lee, Y., Nelder, J. A. and Pawitan, Y. (2006). Generalised linear models with random effects: unified analysis via h-likelihood. Chapman & Hall: London.

Noh, M., and Lee, Y. (2007). REML estimation for binary data in GLMMs, J. Multivariate Anal. 98, 896-915.

Smyth GK, Huele AF, Verbyla AP (2001). Exact and approximate REML for heteroscedastic regression. Statistical Modelling 1, 161-175.

See Also

`HLCor` for estimation with given spatial correlation parameters; `corrHLfit` for joint estimation with spatial correlation parameters.

Examples

```
data(wafers)

## Gamma GLMM with log link
HLfit(y ~X1+X2+X1*X3+X2*X3+I(X2^2)+(1|batch),family=Gamma(log),
          resid.formula = ~ X3+I(X3^2) ,data=wafers)


## Gamma - inverseGamma HGLM with log link
HLfit(y ~X1+X2+X1*X3+X2*X3+I(X2^2)+(1|batch),family=Gamma(log),
          HLmethod="HL(1,1)",rand.family=inverse.Gamma(log),
          resid.formula = ~ X3+I(X3^2) ,data=wafers)
```

Loaloa *Loa loa prevalence in North Cameroon, 1991-2001*

Description

This data set describes prevalence of infection by the nematode *Loa loa* in North Cameroon, 1991-2001. This is a superset of the data discussed by Diggle and Ribeiro (2007) and Diggle et al. (2007). The study investigated the relationship between altitude, vegetation indices, and prevalence of the parasite.

Usage

```
data(Loaloa)
```

Format

The data frame includes 197 observations on the following variables:

latitude latitude, in degrees.

longitude longitude, in degrees.

ntot sample size per location

npos number of infected individuals per location

maxNDVI maximum normalised-difference vegetation index (NDVI) from repeated satellite scans

seNDVI standard error of NDVI

elev1 altitude, in m.

elev2,elev3,elev4 Additional altitude variables derived from the previous one, provided for convenience: respectively, positive values of altitude-650, positive values of altitude-1000, and positive values of altitude-1300

maxNDVI1 `maxNDVI1[maxNDVI1>0.8] <- 0.8`

Source

The data were last retrieved on March 1, 2013 from P.J. Ribeiro's web resources at www.leg.ufpr.br/doku.php/pessoais:paulojus:mbgbook:datasets.

References

Diggle, P., and Ribeiro, P. 2007. Model-based geostatistics, Springer series in statistics, Springer, New York.

Diggle, P. J., Thomson, M. C., Christensen, O. F., Rowlingson, B., Obsomer, V., Gardon, J., Wanji, S., Takougang, I., Enyong, P., Kamgno, J., Remme, J. H., Boussinesq, M., and Molyneux, D. H. 2007. Spatial modelling and the prediction of Loa loa risk: decision making under uncertainty, Ann. Trop. Med. Parasitol. 101, 499-509.

Examples

```
## Not run:
data(Loaloa)
## Variations on the model fit by Diggle et al. on a subset of these data

## With HLmethod=HL(0,1), this takes ~35s on a recent laptop

corrHLfit(cbind(npos,ntot-npos)~elev1+elev2+elev3+elev4+maxNDVI1+seNDVI
                  +Matern(1|longitude+latitude),HLmethod="HL(0,1)",
                data=Loaloa,family=binomial(),ranFix=list(nu=0.5))
## nu=0.5 is the exponential correlation
```

```
## the following fits take ~ 1 -- 5 minutes each.

corrHLfit(cbind(npos,ntot-npos)~elev1+elev2+elev3+elev4+maxNDVI1+seNDVI
            +Matern(1|longitude+latitude),
              data=Loaloa,family=binomial(),ranFix=list(nu=0.5))

## Diggle and Ribeiro (2007) assumed (in this package notation) Nugget=2/7:
corrHLfit(cbind(npos,ntot-npos)~elev1+elev2+elev3+elev4+maxNDVI1+seNDVI
           +Matern(1|longitude+latitude),
             data=Loaloa,family=binomial(),ranFix=list(nu=0.5,Nugget=2/7))

## with nugget estimation:
corrHLfit(cbind(npos,ntot-npos)~elev1+elev2+elev3+elev4+maxNDVI1+seNDVI
           +Matern(1|longitude+latitude),
             data=Loaloa,family=binomial(),
             init.corrHLfit=list(Nugget=0.1),ranFix=list(nu=0.5))

## End(Not run)
```

LRT *Likelihood ratio test of fixed effects.*

Description

LRT performs a likelihood ratio (LR) test between two model fits, the "full" and the "null" model fits, currently differing only in their fixed effects. Parametric bootstrap p-values can be computed, either using the raw bootstrap distribution of the likelihood ratio, or a a bootstrap estimate of the Bartlett correction of the LR statistic. This function differ from fixedLRT in its arguments (model fits for LRT, but all arguments required to fit the models for fixedLRT), and in the format of its return value. The function will stop or return possibly incorrect results for models differing beyond their fixed effects. By conceptual drift, anova works as an alias for LRT.

Usage

```
## S3 method for class 'HLfit'
anova(object,object2,...)
LRT(object,object2,boot.repl=0)
LRT(object,object2,boot.repl=0)
```

Arguments

object,object2	Two models fits being compared (their order does not matter).
boot.repl	the number of bootstrap replicates.
...	Further arguments passed to or from other methods.

Details

A raw bootstrap p-value can also be computed from the simulated distribution as `(1+sum(t >= t0))/(N+1)` where `t0` is the original likelihood ratio, `t` the vector of bootstrap replicates and `N` its length. See Davison & Hinkley (1997, p. 141) for discusion of the adjustments in this formula. The bootstrap can also be used to provide a Bartlett correction for the likelihood ratio test in small sample. According to this correction, the mean value m of the likelihood ratio statistic under the null hypothesis is computed (here estimated by a parametric bootstrap) and the original LR statistic is multiplied by n/m where n is the number of degrees of freedom of the test.

Value

An object of class `fixedLRT`, actually a list with as-yet unstable format, but here with typical elements (depending on the options)

`fullfit`	the HLfit object for the full model;
`nullfit`	the HLfit object for the null model;
`basicLRT`	A data frame including values of the likelihood ratio statistic, its degrees of freedom, and the p-value;

and, if a bootstrap was performed:

`rawBootLRT`	A data frame including values of the likelihood ratio statistic, its degrees of freedom, and the raw bootstrap p-value;
`BartBootLRT`	A data frame including values of the Bartlett-corrected likelihood ratio statistic, its degrees of freedom, and its p-value;
`bootInfo`	a list with the following elements:
	bootreps A table of fitted likelihoods for bootstrap replicates;
	meanbootLRT The mean likelihood ratio chi-square statistic for boostrap replicates;

References

Bartlett, M. S. (1937) Properties of sufficiency and statistical tests. Proceedings of the Royal Society (London) A 160: 268-282.

Davison A.C., Hinkley D.V. (1997) Bootstrap methods and their applications. Cambridge Univ. Press, Cambridge, UK.

See Also

See also `fixedLRT`.

Examples

```
data(wafers)
## Gamma GLMM with log link
m1 <- HLfit(y ~X1+X2+X1*X3+X2*X3+I(X2^2)+(1|batch),family=Gamma(log),
          resid.formula = ~ X3+I(X3^2) ,data=wafers,HLmethod="ML")
m2 <- update(m1,formula.= ~ . -I(X2^2))
anova(m1,m2)
```

`make.scaled.dist` *Scaled Euclidian distances between unique locations*

Description

This fonction computes scaled Euclidian distances from whichever relevant argument it can use (see Details). The result can directly by used as input for computation of the Matérn correlation matrix. It is usually called internally by HLCor, so that users may ignore it, except if they wish to control the parametrizaion of the scaling through the `rho.mapping` argument.

Usage

```
make.scaled.dist(distMatrix,uniqueGeo,rho,rho.mapping=seq_len(length(rho)))
```

Arguments

`distMatrix`	A distance matrix.
`uniqueGeo`	A matrix of geographical coordinates (e.g. 2 columns for latitude and longitude), without replicates o the same location.
`rho`	A scalar or vector of positive values. Scaled distance is computed as `<distances in each coordinate>` unless a non-trivial `rho.mapping` is used.
`rho.mapping`	A set of indices controlling which elements of the `rho` scale vector scales which dimension(s) of the space in which (spatial) correlation matrices of random effects are computed. Scaled distance is generally computed as `<distances in each coordinate> * rho` As shown inthe Example, if one wishes to combine isotropic geographical distance and some environmental distance, the coordinates being latitude, longitude and one environmental variable, the scaled distance may be computed as (say) `(lat,long,env) *rho[c(1,1,2)]` so that the same scaling `rho[1]` applies for both geographical coordinates. In this case, `rho` should have length 2 and `rho.mapping` should be `c(1,1,2)`.

Details

The function uses the `distMatrix` argument if provided, in which case rho must be a scalar. Vectorial `rho` (i.e., different scaling of different dimensions) is feasible only by providing `uniqueGeo`.

Value

A matrix or `dist` object.

Examples

```
## a biologically not very meaningful, but syntactically correct example of rho.mapping
data(blackcap)
corrHLfit(migStatus ~ 1 + Matern(1|latitude+longitude+means),data=blackcap,
        objective="p_v",HLmethod="ML",ranFix=list(nu=0.5),init.corrHLfit=list(rho=c(1,1)),
          rho.mapping=c(1,1,2))
```

mapMM *Colorful plots of predicted responses in two-dimensional space.*

Description

These functions provide either a map of predicted response in analyzed locations, or a predicted surface. The former is a straightforward representation of the analysis of the data, while the latter cope with the fact that all predictor variables may not be known in all locations on a fine spatial grid, but may involve questionable choices as a result (see `map.formula` argument).

Usage

```
mapMM(fitobject,coordinates=NULL,xrange=NULL,yrange=NULL,
      margin=1/20,add.map= FALSE, nlevels = 20,
      color.palette = spaMM.colors,map.asp=NULL,
      col = color.palette(length(levels) - 1), plot.title,
      plot.axes, decorations, key.title, key.axes, xaxs = "i",
      yaxs = "i", las = 1, axes = TRUE, frame.plot = axes, ...)
filled.mapMM(fitobject,coordinates,xrange=NULL,yrange=NULL,
             margin=1/20,map.formula,phi=1e-05,gridSteps=41,
             add.points=quote(points(pred[,coordinates],cex=1,lwd=2)),
             add.map=FALSE,axes = TRUE, plot.axes,map.asp=NULL,...)
```

Arguments

`fitobject`	The return object of a corrHLfit call.
`coordinates`	The geographical coordinates. By default they are deduced from the model formula. For example if this formula is `resp ~ 1 + Matern(1| x + y )` the default coordinates are c("x","y"). If this formula is `resp ~ 1 + Matern(1| x + y + z )`, the user must choose two of the three coordinates.
`xrange`	The x range of the plot (a vector of length 2); by default defined to cover all analyzed points.
`yrange`	The y range of the plot (a vector of length 2); by default defined to cover all analyzed points.
`margin`	This controls how far (in relative terms) the plot extends beyond the x and y ranges of the analyzed points, and is overriden by explicit `xrange` and `yrange` arguments.
`map.formula`	Plotting a filled contour generally requires prediction in non-oberved locations, where predictor variables used in the original data analysis may be missing. In that case, the original model formula cannot be used and an alternative `map.formula` must be used to interpolate (not smooth) the predicted values in observed locations (these predictions still resulting from the original analysis based on predictor variables). As a result (1) `filled.mapMM` will be slower than a mere plotting function, since it involves the analysis of spatial data; (2) the results may have little useful meaning if the effects of the original predictor variables is not correctly represented by this interpolation step. For example, it may involve biases

analogous to predicting temperature in non-oberved locations while ignoring effect of variation in altitude in such locations.

`phi` This controls the phi valu assumed in the interpolation step. Ideally `phi` would be zero, but problems with numerically singular matrices may arise when `phi` is too small.

`gridSteps` The number of levels of the grid of x and y values

`add.points` Either a boolean or an explicit expression, enclosed in `quote` (the default value illustrates the latter syntax). This controls whether and how analyzed locations are represented on the map.

`add.map` Either a boolean or an explicit expression, enclosed in `quote` (see Examples). If `TRUE`, the `map` function from the `maps` package (which much therefore the loaded) is used to add a map from its default `world` database. `xrange` and `yrange` are used to select the area, so it is most convenient if the `coordinates` are longitude and latitude (in this order and in standard units). An explicit expression can also be used for further control.

`levels` a set of levels which are used to partition the range of z. Must be strictly increasing (and finite). Areas with z values between consecutive levels are painted with the same color.

`nlevels` if levels is not specified, the range of z, values is divided into approximately this many levels.

`color.palette` a color palette function to be used to assign colors in the plot.

`map.asp` the y/x aspect ratio of the 2D plot area (not of the full figure including the scale). Default is (plotted y range)/(plotted x range) (i.e., scales for x are identical).

`col` an explicit set of colors to be used in the plot. This argument overrides any palette function specification. There should be one less color than levels

`plot.title` statements which add titles to the main plot. If provided, further `...` arguments are ignored (see Details).

`plot.axes` statements which draw axes (and a box) on the main plot. Default axes are drawn by default when this argument is missing, given `axes = TRUE`.

`decorations` Additional graphic statements (`points`, `polygon`, etc.).

`key.title` statements which add titles for the plot key.

`key.axes` statements which draw axes on the plot key.

`xaxs` the x axis style. The default is to use internal labeling.

`yaxs` the y axis style. The default is to use internal labeling.

`las` the style of labeling to be used. The default is to use horizontal labeling.

`axes, frame.plot`
logicals indicating if axes and a box should be drawn, as in plot.default.

`...` further arguments passed to or from other methods. For `mapMM`, currently only additional graphical parameters passed to `title()` (see Details). For `filled.mapMM`, these parameters are those that can be passed to `spaMM.filled.contour`.

Details

If you have values for all predictor variables in all locations of a fine spatial grid, `filled.mapMM` may not be a good choice. Rather, use `predict(<fitobject>,newX= <all predictor variables >)` to generate all predictions, and then either `spaMM.filled.contour` or some other raster functions.

These functions handle some of their arguments as `filled.contour` does. For `mapMM` in particular, this means that either `plot.title` is missing, or `...` is ignored. Thus, one can provide an optional `xlab` either in the `plot.title` argument, or in the `...` `plot.title` is missing. `filled.mapMM` calls `spaMM.filled.contour` which behaves identically, so the `...` argument of `filled.mapMM` should contain either a `plot.title` or further arguments.

A side effet is that `filled.mapMM`, like `filled.contour`, does not does not provide axis labels (`xlab` and `ylab`) by default.

Value

No return value. Plots are produced as side-effects.

Examples

```
data(blackcap)
bfit <- corrHLfit(migStatus ~ means+ Matern(1|longitude+latitude),data=blackcap,
                  HLmethod="ML",
                  ranFix=list(lambda=0.5537,phi=1.376e-05,rho=0.0544740,nu=0.6286311))
mapMM(bfit,color.palette = function(n){spaMM.colors(n,redshift=1/2)})


## showing add.map
filled.mapMM(bfit,add.map=TRUE,plot.axes={axis(1);axis(2)},
             plot.title=title(main="Inferred migration propensity of blackcaps",
                              xlab="longitude",ylab="latitude"))
## filled.mapMM takes a bit longer
filled.mapMM(bfit,nlevels=30,plot.axes={axis(1);axis(2)},
              plot.title=title(main="Inferred migration propensity of blackcaps",
                              xlab="longitude",ylab="latitude"))



## showing more involved use of add.map
data(Loaloa)
lfit <- corrHLfit(cbind(npos,ntot-npos)~elev1+elev2+elev3+elev4+maxNDVI1+seNDVI
                    +Matern(1|longitude+latitude),HLmethod="HL(0,1)",data=Loaloa,
                    family=binomial(),ranFix=list(nu=0.5,rho=2.255197,lambda=1.075))

if(require(maps)) {
  mapMM(lfit,add.map=quote(map(,xlim=xrange,ylim=yrange,add=TRUE)))
}



## longer computation requiring interpolation of 197 points
filled.mapMM(lfit,add.map=TRUE,plot.axes={axis(1);axis(2)},
```

```
add.points=quote(points(pred[,coordinates],pch=15,cex=0.3)),
plot.title=title(main="Inferred prevalence, North Cameroon",
                 xlab="longitude",ylab="latitude"))
```

`Matern.corr` *Matern correlation function.*

Description

A family of correlation function describing realizations of Gaussian spatial processes with different smoothnesses (i.e. either smooth or rugged surfaces). Also includes a scaling and a 'nugget' parameter.

Usage

```
Matern.corr(d, rho = 1, smoothness, nu = smoothness, Nugget = 0L)
```

Arguments

`d`	A distance, typically an Euclidian distance
`rho`	A scaling factor for distance. The 'range' considered in some formulations is the reciprocal of this scaling factor
`smoothness`	The smoothness parameter, >0. $\nu = 0.5$ corresponds to the exponential correlation function, and the limit function when μ goes to ∞ is the squared exponential function (as in a Gaussian).
`nu`	Same as smoothness
`Nugget`	(Following the jargon of Kriging) a parameter describing a discontinuous decrease in correlation at zero distance. Correlation will always be 1 at $d = 0$, and from which it immediately drops to (1-Nugget)

Details

The correlation at distance $d > 0$ is

$$(1 - \text{Nugget}) \frac{(\rho d)^\nu K_\nu(\rho d)}{2^{(\nu-1)}\Gamma(\nu)}$$

where K_ν is the `besselK` function of order ν.

Value

Scalar/vector/matrix depending on input.

References

Stein, M.L. (1999) Statistical Interpolation of Spatial Data: Some Theory for Kriging. Springer, New York.

See Also

By default the Nugget is set to 0. See one of the examples on data set Loaloa for a fit including the estimation of the Nugget.

Examples

```
## The Matérn function can be used in Euclidian spaces of any dimension:
set.seed(123)
randpts <- matrix(rnorm(20),nrow=5)
distMatrix <- as.matrix(proxy::dist(randpts))
Matern.corr(distMatrix,nu=2)
```

---

multinomial *Analyzing multinomial data*

Description

These functions facilite the conversion and analysis of multinomial data as as series of nested binomial data. The main function is multi, to be used in the family argument of the fitting functions. It calls binomialize, which can be called directly to check how the data are converted to nested binomial data. The fitted.HLfitlist method of the fitted generic function returns a matrix of fitted multinomial probabilities. The logLik.HLfitlist method of the logLik generic function returns a log-likelihood for the joint fits.

Usage

```
multi(binResponse=c("npos","nneg"),binfamily=binomial(),input="types",...)
binomialize(data,responses,sortedTypes=NULL,binResponse=c("npos","nneg"),
            depth=Inf,input="types")
## S3 method for class 'HLfitlist'
fitted(object,...)
## S3 method for class 'HLfitlist'
logLik(object,REML,...)
```

Arguments

data	The data frame to be analyzed.
object	A list of binomial fits returned by a multinomial analysis
responses	column names of the data, such that <data>[,<responses>] contain the multinomial response data, as levels of factor variables.
sortedTypes	Names of multinomial types, i.e. levels of the multinomial response factors. Their order determines which types are taken first to define the nested binomial samples. By default, the most common types are considered first.
binResponse	The names to be given to the number of "success" and "failures" in the binomial response.

depth	The maximum number of nested binomial responses to be generated from the multinomial data.
binfamily	The family applied to each binomial response.
input	If input="types", then the responses columns must contain factor levels of the binomial response. If input="counts", then the responses columns must contain counts of different factor levels, and the column names are the types.
REML	If TRUE, the function returns the log restricted likelihood (exact or approximated).
...	Other arguments passed from or to other functions.

Details

A multinomial response, say counts 17, 13, 25, 8, 3, 1 for types type1 to type6 can be represented as a series of nested binomials e.g. type1 against others (17 vs 50) then among these 50 others, type2 versus others (13 vs 37), etc. The binomialize function generates such a representation. By default the representation considers types in decreasing order of the number of positives, i.e. first type3 against others (25 vs 42), then type1 against others within these 42, etc. It stops if it has reached depth nested binomial responses. This can be modified by the sortedTypes argument, e.g. sortedTypes=c("type6","type4","type2"). binomialize returns a list of data frames which can be directly provided as a data argument for the fitting functions, with binomial response.

Alternatively, one can provide the multinomial response data frame, which will be internally converted to nested binomial data if the family argument is a call to multinomial (see examples).

For mixed models, the multinomial data can be fitted to a model with the same correlation parameters, and either the same or different variances of random effects, for all binomial responses. Which analysis is performed depends on the init.corrHLfit argument (see corrHLfit and the Examples).

Value

binomialize returns a list of data frames appropriate for analysis as binomial response. Each data frame contains the original one plus Two columns named according to binResponse. multi returns a list.

Examples

```
## An example considering pseudo-data at one diploid locus for 50 individuals
set.seed(123)
genecopy1 <- sample(4,size=50,prob=c(1/2,1/4,1/8,1/8),replace=TRUE)
genecopy2 <- sample(4,size=50,prob=c(1/2,1/4,1/8,1/8),replace=TRUE)
alleles <- c("122","124","126","128")
genotypes <- data.frame(type1=alleles[genecopy1],type2=alleles[genecopy2])
## Columns "type1","type2" each contains an allele type => input is "types" (the default)
datalist <- binomialize(genotypes,responses=c("type1","type2"))

## two equivalent fits:
f1 <- HLfit(cbind(npos,nneg)~1,data=datalist, family=binomial())
f2 <- HLfit(cbind(npos,nneg)~1,data=genotypes, family=multi(responses=c("type1","type2")))
fitted(f2)
```

```
## distinct fits for spatial data
genoInSpace <- data.frame(type1=alleles[genecopy1],type2=alleles[genecopy2],x=runif(50),y=runif(50))
## Fitting distinct variances of random effects for each binomial response
corrHLfit(cbind(npos,nneg)~1+Matern(1|x+y),data=genoInSpace,
          family=multi(responses=c("type1","type2")),
          ranFix=list(rho=1,nu=0.5))
## Fitting the same variance for all binomial responses
corrHLfit(cbind(npos,nneg)~1+Matern(1|x+y),data=genoInSpace,
          family=multi(responses=c("type1","type2")),
          ranFix=list(rho=1,nu=0.5),init.corrHLfit=list(lambda=1))
```

`options` *spaMM options settings*

Description

Allow the user to set and examine a variety of *options* which affect operations of the spaMM package.

Usage

```
spaMM.options(...)

spaMM.getOption(x)
```

Arguments

`x`	a character string holding an option name.
`...`	A named value or a list of named values. The following values, with their defaults, are used in `spaMM`:

`RHOMAX=100000`**:** Maximum value of the scale parameter ρ of the Matérn correlation function

`NUMAX=50`**:** Maximum value of the smoothness parameter ν of the Matérn correlation function

`TRACE.UNLINK=FALSE`**:** Whether to delete preexisting files named as `HLCor.args.*.RData` when the trace option is used in corrHLfit or HLCor.

`MESSAGES.FULL.STACK=TRUE`**:** Whether to give information on the stack of calls in some warning messages.

`INIT.HLFITNAME=NA`**:** Something users should not care about.

`USEEIGEN=TRUE`**:** Whether to use the Eigen C++ library for some matrix computations.

`maxLambda=1e10`**:** The maximum value of lambda: higher fitted lambda values in HLfit are reduced to this.

and possibly other undocumented values for development purposes.

Details

Invoking `spaMM.options()` with no arguments returns a list with the current values of the options. Invoking `spaMM.getOption(<option name>)` returns the value of the option rather than a list.

spaMM.options() provides an interface for changing maximal values of parameters of the Matérn correlation function. However, it is not recommended to change these values unless a spaMM message specifically suggests so. Errors may occur if too low values are chosen as these may conflit with default initial values for the parameters.

Value

For `spaMM.getOption`, the current value set for option `x`, or `NULL` if the option is unset.

For `spaMM.options()`, a list of all set options sorted by category. For `spaMM.options(name)`, a list of length one containing the set value, or `NULL` if it is unset. For uses setting one or more options, a list with the previous values of the options changed (returned invisibly).

Examples

```
  spaMM.options()
  spaMM.getOption("NUMAX")
  ## Not run:
  spaMM.options(maxLambda=1e06)

## End(Not run)
```

`plot.HLfit` *Model checking plots for mixed models*

Description

This function provides diagnostic plots for residual errors from the mean model and for random effects. Plots for the mean models are similar to those for GLMs, as described in Lee et al. 2006. Plots for residual errors consider the *standardized* deviance residuals (Lee et al. 2006, p.52), and plots for random effects likewise consider standardized values, i.e. each random deviate divided by $\sqrt{(1-q)}$ where q is the corresponding leverage for λ.

Usage

```
## S3 method for class 'HLfit'
plot(x, which = c("mean", "ranef"),
      titles = list(
          meanmodel=list(outer="Mean model",devres="Deviance residuals",
                    absdevres="|Deviance residuals|", resq="Residual quantiles",
                         devreshist="Deviance residuals"),
        ranef=list(outer="Random effects and leverages",qq="Random effects Q-Q plot",
                     levphi=expression(paste("Leverages for ",phi)),
          levlambda=expression(paste("Leverages for ",lambda)))
        ),
      control= list(), ...)
```

Arguments

x	The return object of an HLCor / HLfit / corrHLfit call.
which	By default, two types of are presented on different devices: diagnostic plots for mean values, and diagnostic plots for random effects. Either one can be selected using this argument.
titles	A list of the main (inner and outer) titles of the plots. See the default value for the format.
control	A list of default options for the plots. Defaults are pch="+" and pcol="blue" for points, and lcol="red" for curves.
...	Options passed from plot.HLfit to par, or from plot.HLCor or plot.corrHLfit to plot.HLfit.

Details

The standardized deviance residuals are defined as the deviance residuals divided by $\phi\sqrt{(1-q)}$, where q is the corresponding leverage for ϕ, and the deviance residuals are defined as for a GLM. The leverages are zero for ML methods. Otherwise, they depend on the fitting method used, as defined in the Details of HLfit. The PQL and EQL- method use leverages obtained as diagonal elements of the "hat" matrix; more elaborate methods will introduce corrections for non-Gaussian response and for non-Gaussian random effects; and "(.,1)" methods will add another correction taking into account the variation of the GLM weights in the logdet Hessian term of restricted likelihood.

In principle the deviance residuals for the mean model should have a nearly Gaussian distribution hence form a nearly straight line on a Q-Q plot. However this is (trivially) not so for well-specified (nearly-)binary response data nor even for well-specified Poisson response data with moderate expectations. Hence this plot is not so useful.

Value

Returns the input object invisibly.

References

Lee, Y., Nelder, J. A. and Pawitan, Y. (2006). Generalised linear models with random effects: unified analysis via h-likelihood. Chapman & Hall: London.

Examples

```
## see example for data(scotlip)
```

predict *Prediction from a model fit.*

Description

Predictions of the response variable, based on given values of the predictor variables for fixed effects, and/or on predicted random effects.

Usage

```
## S3 method for class 'HLfit'
predict(
                object,newX = NULL,coeffs=NULL,re.form= NULL,
            variances=list(fixef=FALSE,ranef=FALSE,resid=FALSE,sum=FALSE,cov=FALSE),
                predVar=variances$ranef,residVar=variances$resid,
                binding = if(is.vector(newX)) {FALSE} else {"fitted"},...)
```

Arguments

object	The return object of an HLfit or similar function.
newX	**Either** a matrix or data frame containing all required variables for evaluating fixed and random effects, including an offset. If NULL, the original data are reused. **or** a numeric vector, which names (if any) are ignored. This makes it easier to use predict as an objective function for an optimization procedure such as optim, which calls the objective function on unnamed vectors. However, one must make sure that the order of elements in the vector is the order of first occurrence of the variables in the model formula. This order can be checked in the error message returned when calling predict on a newX vector of clearly wrong size, e.g. predict(<object>,newX=numeric(0)).
coeffs	Precomputed coefficients for the prediction (see Details).
re.form	formula for random effects to include. If NULL, include all random effects; if NA, include no random effects
variances	A list which elements control whether to compute different estimated variances (and their sum). fixef=TRUE will provide the variances of $\mathbf{X}\beta$; ranef=TRUE will provide the prediction variance of the random effects; resid=TRUE will provide the residual variances (for Gaussian or Gamma responses). These different variances are returned as attributes "fixefVar", "predVar" and "residVar", respectively. If sum=TRUE, all components are computed. If more than one of the components is computed, their sum is returned as attribute "sumVar". If cov=TRUE, the full covariance matrices are returned for any of the requested terms (except for "residVar", as the covariance matrix of the residuals is diagonal). See Details for how "predVar" is computed.
predVar	(for back-compatibility: variances should now be used) predVar=TRUE corresponds to variances=list(ranef=TRUE), and predVar="Cov" corresponds to variances=list(ranef=TRUE,cov=TRUE).

residVar	(for back-compatibility: variances should now be used) residVar=TRUE corresponds to variances=list(resid=TRUE).
binding	If binding is a valid variable name for a data frame, the predicted values are bound (under the given name) with the data frame used for prediction and the resulting frame is returned. If binding is FALSE, The predicted values are returned as a matrix and the data frame used for prediction is returned as an attribute (unless it was NULL).
...	further arguments passed to or from other methods.

Details

If newX is NULL, predict only returns the fitted responses, including random effects, from the object. Otherwise it computes new predictions including random effects as far as possible. For spatial random effects it constructs a correlation matrix **C** between new locations and locations in the original fit. Then it infers the random effects in the new locations as **C** $(\mathbf{L}')^{-1}$ **v** (see spaMM for notation). If the predictor is used many times, it may be useful to precompute $(\mathbf{L}')^{-1}$ **v** and to provide this vector through the coeffs argument (see Examples). For non-spatial random effects, it checks whether any group (i.e., level of a random effect) in the new data was represented in the original data, and it adds the inferred random effect for this group to the prediction for individuals in this group.

The **prediction variance** is the variance of the linear predictor (η). "predVar" is the prediction variance of the random effect terms in (η). It takes into account the uncertainty in estimation of β, and is computed as described in Gotway and Wolfinger (2003) based on earlier works for LMMs.

Unobserved levels of non-spatial random effects are handled as follows. In the **point prediction** of the linear predictor, the expected value of u is assigned to the realizations of u for unobserved groups (this value is 0 in LMMs). Corresponding realizations of v are then deduced using the link function(s) for the random effects (the identity link in LMMs). The same computation is performed in all other models, for good or bad. For **prediction covariance**, it matters whether a single or multiple new levels are used: see Examples.

Value

A matrix or data frame (according to the binding argument), with optionally one or more prediction variance vector or (co)variance matrices as attributes.

References

Gotway, C.A., Wolfinger, R.D. (2003) Spatial prediction of counts and rates. Statistics in Medicine 22: 1415-1432.

Examples

```
data(blackcap)
fitobject <- corrHLfit(migStatus ~ 1 + Matern(1|latitude+longitude),data=blackcap,
                       ranFix=list(nu=4,rho=0.4,phi=0.05))
predict(fitobject)

predict(fitobject,blackcap) ## same computation, different format
```

```
## same result using precomputed 'coeffs':
coeffs <- predictionCoeffs(fitobject) ## using dedicated extractor function
predict(fitobject,coeffs=coeffs,variances=list(sum=TRUE)) -> pf
attr(pf,"sumVar")


###### handling of unobserved groups
## (1) fit with an additional random effect
grouped <- cbind(blackcap,grp=c(rep(1,7),rep(2,7)))
fitobject <- corrHLfit(migStatus ~ 1 +  (1|grp) +Matern(1|latitude+longitude),
                       data=grouped,  ranFix=list(nu=4,rho=0.4,phi=0.05))
## (2) comparison of covariance matrices for two types of new data
moregroups <- grouped[1:5,]
rownames(moregroups) <- paste("newloc",1:5,sep="")
moregroups$grp <- rep(3,5) ## all new data belong to an unobserved third group
cov1 <- attr(predict(fitobject,newX=moregroups,
                     variances=list(ranef=TRUE,cov=TRUE)),"predVar")
moregroups$grp <- 3:7 ## all new data belong to distinct unobserved groups
cov2 <- attr(predict(fitobject,newX=moregroups,
                     variances=list(ranef=TRUE,cov=TRUE)),"predVar")
cov1-cov2 ## the expected off-diagonal covariance due to the common group in the first fit.

## Not run:
## Effects of numerically singular correlation matrix C:
fitobject <- corrHLfit(migStatus ~ 1 + Matern(1|latitude+longitude),data=blackcap,
                       ranFix=list(nu=10,rho=0.001)) ## numerically singular C
predict(fitobject) ## predicted mu computed as X beta + L v
predict(fitobject,newX=blackcap) ## predicted mu computed as X beta + C

## End(Not run)
```

`Predictor` *Interface for model formulas*

Description

`Predictor(...)` performs some minimal syntax checking, and returns a formula with attributes. It serves as a unified interface for the set of descriptors for a linear predictor, including design matrices for random effects.

In the current version this function could be ignored by users.

Usage

```
Predictor(formula, offset=NULL, LMatrix = NULL, AMatrix = NULL, ZALMatrix = NULL)
```

Arguments

`formula`	a `formula`, which can include fixed effects, random effects, and offsets.

offset	a offset can be provided in this way, as a numeric vector. However, it **may be better** to provide the offset as an offset formula term (see scotlip example), in particular for later use with predict where the formula can be reevaluated on new data.
LMatrix	The "square root" of the correlation matrix between unique locations, see Details.
AMatrix	A matrix that relates observed (unique) locations to unobserved locations, see Details.
ZALMatrix	The design matrix for random effects, see Details.

Details

In a spatial model a vector of correlated random effects **Lv** can be constructed from uncorrelated ones, **v**, for some matrix **L** (this may be meaningful only for Gaussian random effects). Typically **L** is the Cholesky "square root" of a correlation matrix determined by the random effect specification (e.g., Matern(...)), or given as the corrMatrix argument of HLCor.

If there is one realized random effect per response value, the linear predictor contains **Lv**, where **L** is a square matrix which dimension is the number of observations.

Several observations may be taken in the same location, and a matrix **Z** (usually automatically constructed) tells which element of **Lv** affects each observation. The linear predictor then contains **ZLv**, where dim(Z) is (number of observations,number of locations).

Finally, in some applications the realized random effects in response locations may be viewed as linear combinations **ALv** of random effects **Lv** in distinct locations. In that case the dimension of **L** is the number of such distinct locations, **A** maps them to the observed locations, and **Z** again maps them to possiby repeated observations in observed locations.

Thus, in general the random term in the linear predictor is written **Mv**, where **M=ZAL** is reconstructed from the element matrices (usually automatically constructed if needed), unless **ZAL** is given as argument.

Value

A formula with attributes. This return object has classes formula and predictor.

Examples

```
# In the current version this function can be ignored by users,
# so examples are not required.
# (Use of AMatrix could perhaps be shown)
```

---

salamander *Salamander mating data*

Description

Data from a salamander mating experiment discussed by McCullagh and Nelder (1989, Ch. 14). Twenty males and twenty females from two populations (Rough Butt and Whiteside) were each paired with 6 individuals from their own or from the other population. The experiments were later published by Arnold et al. (1996).

Usage

```
data(salamander)
```

Format

The data frame includes 360 observations on the following variables:

Female Index of the female;

Male Index of the male;

Mate Whether the pair successfully mated or not;

TypeF Population of origin of female;

TypeM Population of origin of male;

Cross Interaction term between `TypeF` and `TypeM`;

Season A factor with levels `Summer` and `Fall`;

Experiment Index of experiment

Source

The data frame was borrowed from the `HGLMMM` package (Molas and Lesaffre, 2011), version 0.1.2.

References

Arnold, S.J., Verrell, P.A., and Tilley S.G. (1996) The evolution of asymmetry in sexual isolation: a model and a test case. Evolution 50, 1024-1033.

McCullagh, P. and Nelder, J.A. (1989). Generalized Linear Models, 2nd edition. London: Chapman & Hall.

Molas, M., Lesaffre, E. (2011) Hierarchical Generalized Linear Models: The R Package HGLMMM. Journal of Statistical Software 39, 1-20.

Examples

```
data(salamander)
HLfit(cbind(Mate,1-Mate)~TypeF+TypeM+TypeF*TypeM+(1|Female)+(1|Male),
    family=binomial(),data=salamander,HLmethod="ML")
```

scotlip *Lip cancer in Scotland 1975 - 1980*

Description

This data set provides counts of lip cancer diagnoses made in Scottish districts from 1975 to 1980, and additional information relative to these data from Clayton and Kaldor (1987) and Clayton and Caldor (1993). The data set contains (for each district) counts of disease events and estimates of the fraction of the population involved in outdoor industry (agriculture, fishing, and forestry) which exposes it to sunlight.

`data("scotlip")` actually loads a data frame, `scotlip`, and an adjacency matrix, `Nmatrix`, between 56 Scottish districts, as given by Clayton and Kaldor (1987, Table 1).

Usage

```
data(scotlip)
```

Format

The data frame includes 56 observations on the following 7 variables:

gridcode alternative district identifier.

id numeric district identifier (1 to 56).

district district name.

cases number of lip cancer cases diagnosed 1975 - 1980.

population total person years at risk 1975 - 1980.

prop.ag percent of the population engaged in outdoor industry.

expec offsets considered by Breslow and Clayton (1993, Table 6, 'Exp' variable)

The rows are ordered according to `gridcode`, so that they match the rows of `Nmatrix`.

References

Clayton D, Kaldor J (1987). Empirical Bayes estimates of age-standardized relative risks for use in disease mapping. Biometrics, 43: 671 - 681.

Breslow, NE, Clayton, DG. (1993). Approximate Inference in Generalized Linear Mixed Models. Journal of the American Statistical Association: 88 9-25.

Examples

```
data(scotlip)
scfit <- HLCor(cases~I(prop.ag/10) +adjacency(1|gridcode)+offset(log(scotlip$expec)),
      ranPars=list(rho=0.174),adjMatrix=Nmatrix,family=poisson(),data=scotlip)
summary(scfit)
plot(scfit)
```

seaMask *Masks of seas or lands*

Description

These convenient masks can be added to maps of (parts of) the world to mask map information for these areas.

Usage

```
data(seaMask)
data(landMask)
```

Format

seaMask and landMask are data frames with two variables, x and y for longitude and latitude. Its contents are suitable for use with polypath: they define different polygones, each separated by a row of NAs.

Details

polypath requires polygons, while map(interior=FALSE,plot=FALSE) returns small segments. landMask is the result of reconnecting the segments into full coastlines of all land blocks.

Examples

```
data(seaMask)
## plot of predictions of behaviour for a land bird:
if (require(maps)){
 data(blackcap)
 bfit <- corrHLfit(migStatus ~ means+ Matern(1|longitude+latitude),data=blackcap,
                 HLmethod="ML",
                 ranFix=list(lambda=0.5537,phi=1.376e-05,rho=0.0544740,nu=0.6286311))
 ## We add small masks to the points on small islands to see the predictions there
 ll <- blackcap[,c("longitude","latitude")]
 pointmask <- function(xy,r=1,npts=12) {
  theta <- 2*pi/npts *seq(npts)
  hexas <- lapply(seq(nrow(xy)),function(li){
    p <- as.numeric(xy[li,])
    hexa <- cbind(x=p[1]+r*cos(theta),y=p[2]+r*sin(theta))
    rbind(rep(NA,2),hexa) ## inital NA before each polygon
  })
  do.call(rbind,hexas)
 }
 pmasks <- pointmask(ll[c(2,4,5,6,7),],r=0.8) ## small islands only
 filled.mapMM(bfit,add.map=TRUE,
             plot.title=title(main="Inferred migration propensity of blackcaps",
                              xlab="longitude",ylab="latitude"),
             add.points=quote(points(pred[,coordinates],cex=1,pch="+")),
```

```
              plot.axes={axis(1);axis(2);
                         polypath(rbind(seaMask,pmasks),border=FALSE,
                                  col="grey", rule="evenodd")
              })
}
```

---

seeds *Seed germination data*

Description

A classic toy data set, "from research conducted by microbiologist Dr P. Whitney of Surrey University. A batch of tiny seeds is brushed onto a plate covered with a certain extract at a given dilution. The numbers of germinated and ungerminated seeds are subsequently counted" (Crowder, 1978). Two seed types and two extracts are here considered in a 2x2 factorial design.

Usage

```
data(seeds)
```

Format

The data frame includes 21 observations on the following variables:

plate Factor for replication;

seed Seed type, a factor with two levels O73 and O75;

extract Root extract, a factor with two levels Bean and Cucumber;

r Number of seeds that germinated;

n Total number of seeds tested

Source

Crowder (1978), Table 3.

References

Crowder, M.J., 1978. Beta-binomial anova for proportions. Appl. Statist., 27, 34-37. Y. Lee and J. A. Nelder. 1996. Hierarchical generalized linear models (with discussion). J. R. Statist. Soc. B, 58: 619-678.

Examples

```
data(seeds)
## An extended quasi-likelihood (EQL) fit as considered by Lee and Nelder (1996):
HLfit(cbind(r,n-r)~seed*extract+(1|plate),family=binomial(),
  rand.family=Beta(),
  HLmethod="HL(0,0)",
  data=seeds)
```

`simulate.HLfit` *Simulate realizations of a fitted mixed model.*

Description

From an HLfit object, `simulate.HLfit` function generates new samples given the estimated fixed effects and dispersion parameters.

This does not yet work for new locations in a mixed model involving both spatial and non-spatial random effects.

Usage

```
## S3 method for class 'HLfit'
simulate(object, nsim = 1, seed = NULL, newX=NULL, sizes=object$weights,...)
## S3 method for class 'HLfitlist'
simulate(object, nsim = 1, seed = NULL,
                          newX=object[[1]]$data, sizes=object[[1]]$weights,...)
```

Arguments

`object`	The return object of HLfit or similar function.
`nsim`	number of response vectors to simulate. Defaults to '1'.
`seed`	A seed for `set.seed`. If such a value is provided, the initial state of the random number generator at a global level is restored on exit from simulate.
`newX`	A data frame closely matching the original data, except that response values are not needed. May provide new values of fixed predictor variables, new spatial locations, or new individuals within a block.
`sizes`	A vector of sample sizes to simulate in the case of a binomial fit. Defaults to the sizes in the original data.
`...`	further arguments passed to or from other methods.

Value

For the `HLfitlist` method (i.e., the result of a multinomial fit), a list of simulated responses. Otherwise, a vector (if nsim=1) or a matrix with nsim columns, each containing a simulated response.

Examples

```
data(Loaloa)
HLC <- HLCor(cbind(npos,ntot-npos)~Matern(1|longitude+latitude),
           data=Loaloa,family=binomial(),
           ranPars=list(lambda=1,nu=0.5,rho=1/0.7))
simulate(HLC,nsim=2)
```

spaMM *Inference in spatial GLMMs (and other non-spatial mixed models)*

Description

spaMM has been first designed to fit spatial mixed models. It can also fit a wider class of non-spatial mixed models. The random effects are either Gaussian (which defines GLMMs), or other distributions (which defines the wider class of hierarchical GLMs), or simply absent (which makes a GLM).

Details

The standard (G)LMM response families `gaussian`, `binomial`, `poisson`, and `Gamma` are handled, as well as `multinomial` response (with some constraints) and negative binomial response (see Examples). GLMMs and HGLMs are fit via Laplace approximations for (1) the marginal likelihood with respect to random effects and (2) the restricted likelihood (as in REML), i.e the likelihood of random effect parameters given the fixed effect estimates.

The package fits spatial models following either a Matérn correlation model or an adjacency matrix model. It also handles a number of non-spatial models. It accepts several nested or crossed random effects (see Examples). Only one of these can be a spatial random effect. The spaMM code has not been optimized for speed in these models.

The variance(s) of random effects (u) is (are) denoted λ (`lambda` in input and output) and that of residual error is denoted ϕ (`phi`). A fixed-effects linear predictor for ϕ, modeling heteroskedasticity, can be considered (see Examples). Fixed effects are described in the standard form $\mathbf{X}\beta$ where $\mathbf{X}$ is the design matrix of fixed effects and β (`beta`) is a vector of fixed effect parameters.

The structure of the random effects can generally be described by the following steps. First, independent and identically distributed (iid) random effects u are drawn from one of the following distributions: gaussian, Beta-distributed, Gamma and inverse-Gamma distributed random effects, implemented as detailed in the `HLfit` documentation. Second, a transformation $\mathbf{v}= f(\mathbf{u})$ is applied ($\mathbf{v}$ elements are still iid). Third, correlated random effects are obtained as $\mathbf{Mv}$, where the matrix $\mathbf{M}$ can describe spatial correlation between observed locations, block effects (or repeated observations in given locations), and correlations involving unobserved locations. See Details in `Predictor` for the general form of $\mathbf{M}$. In most cases $\mathbf{M}$ is determined from the model formula, but it can also be input directly (e.g., to describe genetic correlations).

The package has been extensively tested mainly for analysis of spatial GLMMs (Rousset and Ferdy 2014), where the random effects are Gaussian. Other models have been checked against literature results and a few simulations.

Author(s)

François Rousset and Jean-Baptiste Ferdy.

The syntax of formula terms mostly matches the one in the `lme4` package, so bits of the `lme4` code for parsing formulas has been recycled in `spaMM`.

References

Lee, Y., Nelder, J. A. and Pawitan, Y. (2006). Generalised linear models with random effects: unified analysis via h-likelihood. Chapman & Hall: London.

Rousset F., Ferdy, J.-B. (2014) Testing environmental and genetic effects in the presence of spatial autocorrelation. Ecography, 37: 781-790. http://dx.doi.org/10.1111/ecog.00566

See Also

spaMM is designed to be used through the high-level functions corrHLfit, HLCor, HLfit, and fixedLRT

Examples

```
## Fit a Poisson GLMM with adjacency correlation model


data(scotlip) ## loads 'scotlip' data frame, but also 'Nmatrix'
corrHLfit(cases~I(prop.ag/10) +adjacency(1|gridcode)+offset(log(scotlip$expec)),
          data=scotlip,family=poisson(),
          adjMatrix=Nmatrix,lower=list(rho=0),upper=list(rho=0.1745))

## Adding a Gamma random effect to fit a negative-binomial response:
corrHLfit(cases~I(prop.ag/10) +(1|gridcode)+adjacency(1|gridcode)
                +offset(log(scotlip$expec)),
          data=scotlip,family=poisson(),rand.family=list(Gamma(log),gaussian()),
          adjMatrix=Nmatrix,lower=list(rho=0),upper=list(rho=0.1745))



## fit non-spatial crossed random effects with distinct families
data(salamander)
HLfit(cbind(Mate,1-Mate)~1+(1|Female)+(1|Male),family=binomial(),
        rand.family=list(gaussian(),Beta(logit)),data=salamander,HLmethod="ML")

## Nested effects
# lmer syntax allowing several degrees of nesting
HLfit(cbind(Mate,1-Mate)~1+(1|Female/Male),
       family=binomial(),rand.family=Beta(logit),data=salamander,HLmethod="ML")

# A syntax described in ?formula
HLfit(cbind(Mate,1-Mate)~1+(1|Female)+(1|Male %in% Female),
       family=binomial(),rand.family=Beta(logit),data=salamander,HLmethod="ML")


## fit a non-spatial, Gamma GLMM:
data(wafers)
HLfit(y ~X1*X3+X2*X3+I(X2^2)+(1|batch),family=Gamma(log),
          data=wafers)

## Same with fixed-effects predictor for residual variance
## ( = structured-dispersion model):
```

```
HLfit(y ~X1*X3+X2*X3+I(X2^2)+(1|batch),family=Gamma(log),
          resid.formula = ~ X3+I(X3^2) ,data=wafers)

## Random-slope model (mind the output!)
HLfit(y~X1+(X2|batch),data=wafers)

## fit a GLM (not mixed) with structured dispersion:
HLfit( y ~X1+X2+X1*X3+X2*X3+I(X2^2),family=Gamma(log),
      resid.formula = ~ X3+I(X3^2) ,data=wafers)

## Fit of binary data using PQL/L. See ?arabidopsis
## Not run:
data(arabidopsis)
HLCor(cbind(pos1046738,1-pos1046738)~seasonal+Matern(1|LAT+LONG),
                   ranPars=list(rho=0.129,lambda=4.28,nu=0.291),
                   family=binomial(),HLmethod="PQL/L",data=arabidopsis)

## End(Not run)
```

`spaMM.colors` *A flashy color palette.*

Description

`spaMM.colors` is the default color palette for some color plots in `spaMM`.

Usage

```
spaMM.colors(n = 64, redshift = 1)
```

Arguments

`n`	Number of color levels.
`redshift`	The higher it is, the more the palette blushes....

Details

If you don't like this color palette, have a look at the various ones provided by the `fields` package.

Value

A vector giving the colors in a hexadecimal format.

Examples

```
## see mapMM examples
```

spaMM.filled.contour *Level (Contour) Plots with better aspect ratio control (for geographical maps, at least)*

Description

This function is derived from `filled.contour` in the `graphics` package, and this documentation is likewise heavily based on that of `filled.contour`.

This function likewise produces a contour plot with the areas between the contours filled in solid color, and a key showing how the colors map to z values is likewise shown to the right of the plot. The only difference is the way the aspect ratio is determined and can be controlled (using the `map.asp` parameter instead of `asp`), They thus easily provide nice-looking maps with meaningful latitude/longitude ratio (see Examples). However, this does not work well with rstudio.

Usage

```
spaMM.filled.contour(x = seq(0, 1, length.out = nrow(z)),
                     y = seq(0, 1, length.out = ncol(z)),
                     z,
                     xlim = range(x, finite = TRUE),
                     ylim = range(y, finite = TRUE),
                     zlim = range(z, finite = TRUE),
                     levels = pretty(zlim, nlevels), nlevels = 20,
                     color.palette = spaMM.colors,
                     col = color.palette(length(levels) - 1),
                     plot.title, plot.axes, key.title, key.axes,
                     map.asp = NULL, xaxs = "i", yaxs = "i", las = 1,
                     axes = TRUE, frame.plot = axes, ...)
```

Arguments

`x, y`	locations of grid lines at which the values in `z` are measured. These must be in ascending order. (The rest of this description does not apply to `.filled.contour`.) By default, equally spaced values from 0 to 1 are used. If `x` is a `list`, its components `x$x` and `x$y` are used for `x` and `y`, respectively. If the list has component `z` this is used for `z`.
`z`	a numeric matrix containing the values to be plotted.. Note that `x` can be used instead of `z` for convenience.
`xlim`	x limits for the plot.
`ylim`	y limits for the plot.
`zlim`	z limits for the plot.
`levels`	a set of levels which are used to partition the range of `z`. Must be **strictly** increasing (and finite). Areas with `z` values between consecutive levels are painted with the same color.

nlevels	if levels is not specified, the range of z, values is divided into approximately this many levels.
color.palette	a color palette function to be used to assign colors in the plot.
col	an explicit set of colors to be used in the plot. This argument overrides any palette function specification. There should be one less color than levels
plot.title	statements which add titles to the main plot.
plot.axes	statements which draw axes (and a box) on the main plot. This overrides the default axes.
key.title	statements which add titles for the plot key.
key.axes	statements which draw axes on the plot key. This overrides the default axis.
map.asp	the y/x aspect ratio of the 2D plot area (not of the full figure including the scale). Default is (plotted y range)/(plotted x range) (i.e., scales for x are identical).
xaxs	the x axis style. The default is to use internal labeling.
yaxs	the y axis style. The default is to use internal labeling.
las	the style of labeling to be used. The default is to use horizontal labeling.
axes, frame.plot	logicals indicating if axes and a box should be drawn, as in plot.default.
...	additional graphical parameters, currently only passed to title().

Details

The values to be plotted can contain NAs. Rectangles with two or more corner values are NA are omitted entirely: where there is a single NA value the triangle opposite the NA is omitted.

Values to be plotted can be infinite: the effect is similar to that described for NA values.

Note

spaMM.filled.contour uses the layout function and so is restricted to a full page display.

The output produced by spaMM.filled.contour is actually a combination of two plots; one is the filled contour and one is the legend. Two separate coordinate systems are set up for these two plots, but they are only used internally – once the function has returned these coordinate systems are lost. If you want to annotate the main contour plot, for example to add points, you can specify graphics commands in the plot.axes argument. See the Examples.

Author(s)

Heavily based on filled.contour by Ross Ihaka and R-core.

References

Cleveland, W. S. (1993) *Visualizing Data*. Summit, New Jersey: Hobart.

See Also

contour, image, palette; contourplot and levelplot from package lattice.

Examples

```
spaMM.filled.contour(volcano, color = spaMM.colors) # simple

## Comparing the layout with that of filled.contour:
#  (except that it does not achieve the intended effect
#  in RStudio Plots pane).

x <- 10*1:nrow(volcano)
y <- 10*1:ncol(volcano)
spaMM.filled.contour(x, y, volcano, color = terrain.colors,
    plot.title = title(main = "The Topography of Maunga Whau",
    xlab = "Meters North", ylab = "Meters West"),
    plot.axes = { axis(1, seq(100, 800, by = 100))
                  axis(2, seq(100, 600, by = 100)) },
    key.title = title(main = "Height\n(meters)"),
    key.axes = axis(4, seq(90, 190, by = 10)))  # maybe also asp = 1
mtext(paste("spaMM.filled.contour(.) from", R.version.string),
      side = 1, line = 4, adj = 1, cex = .66)

## compare with

filled.contour(x, y, volcano, color = terrain.colors,
    plot.title = title(main = "The Topography of Maunga Whau",
    xlab = "Meters North", ylab = "Meters West"),
    plot.axes = { axis(1, seq(100, 800, by = 100))
                  axis(2, seq(100, 600, by = 100)) },
    key.title = title(main = "Height\n(meters)"),
    key.axes = axis(4, seq(90, 190, by = 10)))  # maybe also asp = 1
mtext(paste("filled.contour(.) from", R.version.string),
      side = 1, line = 4, adj = 1, cex = .66)
```

`summary.HLfit` *Summary and print methods for fit and test results.*

Description

Summary and print methods for results from HLfit or related functions.

Usage

```
## S3 method for class 'HLfit'
summary(object,...)
## S3 method for class 'HLfitlist'
summary(object,...)
```

```
## S3 method for class 'fixedLRT'
summary(object,verbose=TRUE,...)
## S3 method for class 'HLfit'
print(x,...)
## S3 method for class 'HLfitlist'
print(x,...)
## S3 method for class 'fixedLRT'
print(x,...)
```

Arguments

object	The return object of HLfit or related functions.
x	The return object of HLfit or related functions.
verbose	for summary.fixedLRT, whether to print the model fits or not.
...	further arguments passed to or from other methods.

Value

These methods return the object invisibly. They print details of the (lower level) HLfit results in a convenient form.

Examples

```
## see examples of corrHLfit usage
```

---

update.HLfit *Updates an HLCor or HLfit fit*

Description

update will update and (by default) re-fit a model. It does this mostly by extracting the call stored in the object, updating the call and evaluating that call. (however, currently the predictor argument is processed differently).

Usage

```
## S3 method for class 'HLfit'
update(object, formula., ..., evaluate = TRUE)
```

Arguments

object	A return object from an HLfit call.
formula.	Changes to the formula. Beware of the syntax: see *update.formula* for details.
...	Additional arguments to the call, or arguments with changed values. Use name = NULL to remove the argument name.
evaluate	If TRUE, evaluate the new call else return the call.

Value

An HLCor or HLfit fit of the same type as the input object.

See Also

See also HLCor, HLfit.

Examples

```
data(wafers)
## First the fit to be updated:
wFit <- HLfit(y ~X1*X3+X2*X3+I(X2^2)+(1|batch),family=Gamma(log),
          resid.formula = ~ X3+I(X3^2) ,data=wafers)

# For estimates given by Lee et al., Appl. Stochastic Models Bus. Ind. (2011) 27:  315-328:
# Refit with given beta or/and phi values:

betavals <- c(5.55,0.08,-0.14,-0.21,-0.08,-0.09,-0.09)
# reconstruct fitted phi value from predictor for log(phi)
Xphi <- with(wafers,cbind(1,X3,X3^2)) ## design matrix
phifit <- exp(Xphi %*% c(-2.90,0.1,0.95))
update(wFit,formula.= . ~ offset(wFit$`X.pv` %*% betavals)+(1|batch),
       ranFix=list(lambda=exp(-3.67),phi=phifit))

## There are subtlety in performing REML fits of constrained models:
update(wFit,formula.= . ~ offset(wFit$`X.pv` %*% betavals)+(1|batch))
## ... changes the REML correction. Consider instead
update(wFit,formula.= . ~ offset(wFit$`X.pv` %*% betavals)+(1|batch),
       REMLformula=wFit$predictor)
## Alternatively, show original wFit as differences from betavals:
update(wFit,formula.= . ~ . +offset(wFit$`X.pv` %*% betavals))
```

---

wafers *Data from a resistivity experiment for semiconductor materials.*

Description

This data set was reported and analyzed by Robinson et al. (2006) and reanalyzed by Lee et al. (2011). The data "deal with wafers in a single etching process in semiconductor manufacturing. Wafers vary through time since there are some variables that are not perfectly controllable in the etching process. For this reason, wafers produced on any given day (batch) may be different from those produced on another day (batch). To measure variation over batch, wafers are tested by choosing several days at random. In this data, resistivity is the response of interest. There are three variables, gas flow rate (x1), temperature (x2), and pressure (x3) and one random effect (batch or day)." (Lee et al 2011).

Usage

```
data(wafers)
```

Format

The data frame includes 198 observations on the following variables:

y resistivity.

batch batch, indeed.

X1 gas flow rate.

X2 temperature.

X3 pressure.

Source

This data set was manually pasted from Table 3 of Lee et al. (2011). Transcription errors may have occurred.

References

Robinson TJ, Wulff SS, Montgomery DC, Khuri AI. 2006. Robust parameter design using generalized linear mixed models. Journal of Quality Technology 38: 38–65.

Lee, Y., Nelder, J.A., and Park, H. 2011. HGLMs for quality improvement. Applied Stochastic Models in Business and Industry 27, 315-328.

Examples

```
## see examples in the main Documentation page for the package.
```

---

welding *Welding data set*

Description

The data give the results of an unreplicated experiment for factors affecting welding quality conducted by the National Railway Corporation of Japan (Taguchi and Wu, 1980, cited in Smyth et al., 2001). It is a toy example for heterocedastic models and is also suitable for illustrating fit of overparameterized models.

Usage

```
data("welding")
```

Format

The data frame includes 16 observations on 10 variables:

Strengh response variable;

... nine two-level factors.

Source

The data were downloaded from http://www.statsci.org/data/general/welding.txt on 2014/08/19 and are consistent with those shown in table 5 of Bergman and Hynén (1997).

References

Bergman B, Hynén A (1997) Dispersion effects from unreplicated designs in the 2^{k-p} series. Technometrics, 39, 191–98.

Smyth GK, Huele AF, Verbyla AP (2001). Exact and approximate REML for heteroscedastic regression. Statistical Modelling 1, 161-175.

Taguchi G, Wu Y (1980) Introduction to off-line quality control. Nagoya, Japan: Central Japan Quality Control Association.

Examples

```
data(welding)
## toy example from Smyth et al.
HLfit(Strength ~ Drying + Material,resid.formula = ~ Material+Preheating ,data=welding)
## toy example of overparameterized model
HLfit(Strength ~ Rods+Thickness*Angle+(1|Rods),resid.formula = ~ Rods+Thickness*Angle ,data=welding)
```

Index