

# Package ‘coxme’

May 15, 2012

**Title** Mixed Effects Cox Models.

**Maintainer** Terry Therneau <terry.therneau@mayo.edu>

**Priority** optional

**Version** 2.2-3

**Date** 2012-05-15

**Depends** survival (>= 2.36.14), bdsmatrix (>= 1.3), nlme, Matrix (>= 1.0), methods, R (>= 2.12.0)

**Suggests** mvtnorm, kinship2

**LinkingTo** bdsmatrix

**LazyData** Yes

**LazyLoad** Yes

**Author** Terry Therneau

**Description** Cox proportional hazards models containing Gaussian random effects, also known as frailty models.

**License** LGPL-2

**URL** <http://r-forge.r-project.org>

**Repository** CRAN

**Date/Publication** 2012-05-15 15:35:39

## R topics documented:

anova.coxme . . . . .	2
coxme . . . . .	3
coxme.control . . . . .	5
coxmeFull . . . . .	6
coxmeMlist . . . . .	7
eortc . . . . .	8

expand.nested . . . . .	9
fixef.coxme . . . . .	10
fixef.lmekin . . . . .	11
lmekin . . . . .	12
lmekin.object . . . . .	13
print.coxme . . . . .	14
print.lmekin . . . . .	15

<b>Index</b>	<b>16</b>
--------------	-----------

---

anova.coxme	<i>Analysis of Deviance for a Cox model.</i>
-------------	--

---

## Description

Compute an analysis of deviance table for one or more Cox model fits.

## Usage

```
## S3 method for class 'coxme'
anova(object, ..., test = 'Chisq')
```

## Arguments

object	An object of class <code>coxme</code> or <code>coxph</code>
...	Further <code>coxme</code> objects
test	a character string. The appropriate test is a chisquare, all other choices result in no test being done.

## Details

Specifying a single object gives a sequential analysis of deviance table for that fit. That is, the reductions in the model log-likelihood as each term of the formula is added in turn are given in as the rows of a table, plus the log-likelihoods themselves.

If more than one object is specified, the table has a row for the degrees of freedom and loglikelihood for each model. For all but the first model, the change in degrees of freedom and loglik is also given. (This only make statistical sense if the models are nested.) It is conventional to list the models from smallest to largest, but this is up to the user.

The table will optionally contain test statistics (and P values) comparing the reduction in loglik for each row.

## Value

An object of class `"anova"` inheriting from class `"data.frame"`.

**Warning**

The comparison between two or more models by anova or will only be valid if they are fitted to the same dataset. This may be a problem if there are missing values.

**See Also**

[coxme](#), [anova](#).

**Examples**

```
# Testing a shrunken estimate of ECOG performace status
fit1 <- coxph(Surv(time, status) ~ age + sex, data=lung,
             subset=(!is.na(ph.ecog)))
fit2 <- coxme(Surv(time, status) ~ age + sex + (1|ph.ecog), lung)
anova(fit1,fit2)
```

---

 coxme

*Fit a mixed effects Cox model*


---

**Description**

Fit a Cox model containing mixed (random and fixed) effects. Assume a Gaussian distribution for the random effects.

**Usage**

```
coxme(formula, data, weights, subset, na.action, init, control,
      ties = c("efron", "breslow"),
      varlist, vfixed, vinit,
      sparse = c(50, 0.02), x = FALSE, y = TRUE,
      refine.n = 0, random, fixed, variance, ...)
```

**Arguments**

formula	a two-sided formula with a survival object as the left hand side of a ~ operator and the fixed and random effects on the right.
data	an optional data frame containing the variables named in the formula.
subset, weights, na.action	further model specifications arguments as in <code>lm</code> ; see there for details.
init	optional initial values for the fixed effects.
control	optional list of control options. See <code>coxme.control</code> for details.
ties	method for handling exact ties in the survival time.
varlist	the variance family to be used for each random term. If there are multiple terms it will be a list of variance functions. The default is <code>coxmeFull</code> . Alternatively it can be a list of matrices, in which case the <code>coxmeMlist</code> function is used.

vfixed	optional named list or vector used to fix the value of one or more of the variance terms at a constant.
vinit	optional named list or vector giving suggested starting values for the variance.
sparse	rule for deciding sparsity of a random effect. See the main documentation for discussion of the issue.
x	if TRUE the X matrix (fixed effects) is included in the output object
y	if TRUE the y variable (survival time) is included in the output object
refine.n	number of samples to be used in a monte-carlo estimate of possible error in the log-likelihood of the fitted model due to inadequacy of the Laplace approximation.
fixed, random, variance	In the preliminary version of coxme the fixed and random effects were separate arguments. These arguments are included for backwards compatability, but are depreciated. The variance argument is a depreciated alias for vfixed.
...	any other arguments are passed forward to coxme.control.

**Value**

An object of class coxme.

**Author(s)**

Terry Therneau

**References**

S Ripatti and J Palmgren, Estimation of multivariate frailty models using penalized partial likelihood, *Biometrics*, 56:1016-1022, 2000.

T Therneau, P Grambsch and VS Pankratz, Penalized survival models and frailty, *J Computational and Graphical Statistics*, 12:156-175, 2003.

**See Also**

[coxmeFull](#), [coxmeMlist](#)

**Examples**

```
## Not run: # Random treatment effects per institution
fit1 <- coxme(Surv(pgtime, pgstat) ~ stage + trt + (1+trt |institution),
              data=colon2)
fit2 <- coxme(Surv(pgtime, pgstat) ~ stage + trt + (trt |institution) +
              strata(institution), data=colon2)

## End(Not run)
# Shrinkage effects (equivalent to ridge regression)
temp <- with(lung, scale(cbind(age, wt.loss, meal.cal)))
rfit <- coxme(Surv(time, status) ~ ph.ecog + (temp | 1), data=lung)
```

---

coxme.control	<i>Auxillary parameters for controlling coxme fits.</i>
---------------	---

---

### Description

Auxillary function which packages the optional parameters of a coxme fit as a single list.

### Usage

```
coxme.control(eps = 1e-08, toler.chol = .Machine$double.eps^0.75,
  iter.max = 20, inner.iter = Quote(max(4, fit0$iter+1)),
  sparse.calc = NULL,
  optpar = list(method = "BFGS", control=list(reltol = 1e-5)),
  refine.df=4, refine.detail=FALSE, refine.method="control")
```

### Arguments

<code>eps</code>	convergence criteria for the partial likelihood
<code>toler.chol</code>	tolerance for the underlying Cholesky decomposition. This is used to detect singularity (redundant variables).
<code>iter.max</code>	maximum number of iterations for the final fit
<code>inner.iter</code>	number of iterations for the ‘inner loop’ fits, i.e. when the partial likelihood is the objective function of <code>optim</code> . The default is to use one more iteration than the baseline <code>coxph</code> model <code>fit0</code> . The baseline model contains only the fixed effects, and is as part of the setup by the main program. The minimum value of 4 applies most often to the case where there are no fixed effects.
<code>sparse.calc</code>	choice of method 1 or 2 for a particular portion of the calculation. This can have an effect on run time for problems with thousands of random effects.
<code>optpar</code>	parameters passed forward to the <code>optim</code> routine.
<code>refine.df</code>	the degrees of freedom for the t-distribution used to draw random samples for the <code>refine.n</code> option
<code>refine.detail</code>	this option is mostly for debugging. If TRUE then an extra component <code>refine.detail</code> will be present in the output which contains intermediate variables from the iterative refinement calculation.
<code>refine.method</code>	method by which the control calculations are done. This is a current research/development question, the option will likely disappear at some future date, and users should ignore it.

### Details

The main flow of `coxme` is to use the `optim` routine to find the best values for the variance parameters. For any given trial value of the variance parameters, an inner loop maximizes the partial likelihood to select the regression coefficients  $\beta$  (fixed) and  $b$  (random). Within this loop cholesky decomposition is used. It is critical that the convergence criteria of inner loops be less than outer ones, thus  $\text{toler.chol} < \text{eps} < \text{reltol}$ .

**Value**

a list of control parameters

**Author(s)**

Terry Therneau

**See Also**

[coxme](#)

---

coxmeFull

*Variance family function for coxme fits.*

---

**Description**

This function sets up the default variance family information for a mixed effects survival model fit with `coxme`.

**Usage**

```
coxmeFull(collapse = FALSE)
```

**Arguments**

`collapse` Form for fitting a nested effect, either standard or collapsed. The latter appears to be more numerically stable (still under research).

**Details**

Coxme variance families create a list with three functions: `initialize`, `generate`, and `wrapup`, that determine how the variance structure of a fit is modeled.

**Value**

an object of class `coxvar`.

**Author(s)**

Terry Therneau

**See Also**

[coxme](#)

---

`coxmeMlist`*Coxme variance function*

---

**Description**

This variance function accepts a list of matrices, which define a correlation structure for a coxme fit.

**Usage**

```
coxmeMlist(varlist, rescale = FALSE, pdcheck = TRUE, positive = TRUE)
```

**Arguments**

<code>varlist</code>	a list containing one or more matrix or bdsmatrix objects.
<code>rescale</code>	if TRUE, each input matrix is rescaled to have a diagonal of 1. (Kinship matrices for instance are often generated with a diagonal of .5 and would be multiplied by 2).
<code>pdcheck</code>	check each matrix to ensure that it is positive definite
<code>positive</code>	constrain coefficients to be positive. This may also be a vector of the same length as <code>varlist</code>

**Details**

If two matrices  $A$  and  $B$  were given, this fits the variance structure  $V = \sigma_1^2 A + \sigma_2^2 B$ , where the variances  $\sigma_1^2$  and  $\sigma_2^2$  are parameters that will be optimized by coxme, treating  $A$  and  $B$  as fixed.

**Value**

a coxme variance family object, used by coxme in the fitting process.

**Author(s)**

Terry Therneau

**See Also**

[coxme](#)

---

`eortc`*Simulated data set based on an EORTC trial*

---

**Description**

This is a simulated survival data set for investigating random center effects. To make it realistic, the number of centers and their sizes is based on an EORTC cancer trial.

**Usage**

```
data(eortc)
```

**Format**

A data frame with 2323 observations on the following 4 variables.

`y` survival time

`uncens` 0=alive, 1=dead

`center` enrolling center, a number from 1 to 37

`trt` treatment arm, 0 or 1

**Details**

This is used in the test suite for the code.

**Source**

PhD thesis work of Jose Cortinas Abrahantes

**References**

Cortinas Abrahantes, Jose; Burzykowski, Tomasz (2002), A version of the EM algorithm for proportional hazards models with random effects , Published in: Lecture Notes of the ICB Seminars. p. 15-20

**Examples**

```
data(eortc)
coxme(Surv(y, uncens) ~ trt + (trt| center) + strata(center), eortc)
```

---

expand.nested	<i>Expand nested factors</i>
---------------	------------------------------

---

### Description

Expand out the data frame for a nested factor such as (1| a/b). This is used by the variance function routines of coxme.

### Usage

```
expand.nested(x)
```

### Arguments

x                    A data frame containing the nesting variables

### Details

The initialize function of a coxme variance family is passed, as one of its arguments, a data frame G containing the grouping variables, each of which is a factor.. Assume a nested factor (1| a/b) in the model formula and a data set whose first few rows are:

a	b
1	1
1	2
2	1

The function will replace the second column with a variable named a/b and values of 1/1, 1/2, 2/1, etc.

### Value

an updated data frame

### Author(s)

Terry Therneau

### See Also

[coxme](#), [coxmeMlist](#)

**Description**

Extract the fixed effects, random effects, variance of the fixed effects, or variance of the random effects from a coxme model.

**Usage**

```
## S3 method for class 'coxme'  
fixef(object, ...)  
## S3 method for class 'coxme'  
ranef(object, ...)  
## S3 method for class 'coxme'  
vcov(object, ...)  
## S3 method for class 'coxme'  
VarCorr(x, ...)
```

**Arguments**

object	an object inheriting from class coxme representing the result of a mixed effects Cox model.
x	an object inheriting from class coxme representing the result of a mixed effects Cox model.
...	some methods for this generic require additional arguments. None are used in this method.

**Value**

the fixed effects are a vector and the variance of the fixed effects is a matrix. The random effects will be a list with one element for each random effects terms, as will be their variance.

**Author(s)**

Terry Therneau

**See Also**

[coxme](#), [random.effects](#), [fixed.effects](#)

**Examples**

```
rat1 <- coxme(Surv(time, status) ~ rx + (1|litter), rats)  
fixed.effects(rat1)  
vcov(rat1)  
random.effects(rat1)[[1]] #one value for each of the 50 litters  
VarCorr(rat1)
```

---

fixef.lmekin	<i>Extraction functions for Lmekin</i>
--------------	--

---

## Description

Extract the fixed effects, random effects, variance of the fixed effects, or variance of the random effects from a linear mixed effects model fit with lmekin.

## Usage

```
## S3 method for class 'lmekin'  
fixef(object, ...)  
## S3 method for class 'lmekin'  
ranef(object, ...)  
## S3 method for class 'lmekin'  
vcov(object, ...)  
## S3 method for class 'lmekin'  
VarCorr(x, ...)
```

## Arguments

object	an object inheriting from class lmekin representing the result of a mixed effects model.
x	an object inheriting from class lmekin representing the result of a mixed effects model.
...	some methods for this generic require additional arguments. None are used in this method.

## Value

the fixed effects are a vector and the variance of the fixed effects is a matrix. The random effects will be a list with one element for each random effects terms, as will be their variance.

## Author(s)

Terry Therneau

## See Also

[lmekin](#), [random.effects](#), [fixed.effects](#)

## Examples

```
efit <- lmekin(effort ~ Type + (1|Subject), ergoStool)  
ranef(efit)
```

---

lmekin

*Fit a linear mixed effects model*


---

### Description

The `lmekin` function fits a linear mixed effects model, with random effects specified in the same structure as in the `coxme` function.

### Usage

```
lmekin(formula, data, weights, subset, na.action, control,
        varlist, vfixed, vinit, method = c("ML", "REML"),
        sparse = c(1, 0), x = FALSE, y = TRUE,
        random, fixed, variance, ...)
```

### Arguments

<code>formula</code>	a two-sided formula with the response as the left hand side of a <code>~</code> operator and the fixed and random effects on the right.
<code>data</code>	an optional data frame containing the variables named in the formula.
<code>subset</code> , <code>weights</code> , <code>na.action</code>	further model specifications arguments as in <code>lm</code> ; see there for details.
<code>control</code>	optional list of control options. See <code>coxme.control</code> for details.
<code>varlist</code>	the variance family to be used for each random term. If there are multiple terms it will be a list of variance functions. The default is <code>coxmeFull</code> . Alternatively it can be a list of matrices, in which case the <code>coxmeMlist</code> function is used.
<code>vfixed</code>	optional named list or vector used to fix the value of one or more of the variance terms at a constant.
<code>vinit</code>	optional named list or vector giving suggested starting values for the variance.
<code>method</code>	fit using either maximum likelihood or restricted maximum likelihood
<code>sparse</code>	rule for deciding sparsity of a random effect. See the main documentation for discussion of the issue.
<code>x</code>	if <code>TRUE</code> the X matrix (fixed effects) is included in the output object
<code>y</code>	if <code>TRUE</code> the y variable is included in the output object
<code>fixed</code> , <code>random</code> , <code>variance</code>	In an earlier version of <code>lmekin</code> the fixed and random effects were separate arguments. These arguments are included for backwards compatibility, but are depreciated. The <code>variance</code> argument is a depreciated alias for <code>vfixed</code> .
<code>...</code>	any other arguments are passed forward to <code>coxme.control</code> .

**Details**

This routine was originally written as a check for the `coxme` routine; it uses the same code to process input arguments and form the random effects, comparison of its output with `lme` helped validate those operations. It is possible to specify some models in this framework that can not be fit with `lme`, in particular models with familial genetic effects, i.e., a *kinship* matrix, and hence the name of the routine. Using user-specified variance functions an even wider range of models is possible.

For simple models the specification of the random effects follows the same form as the `lmer` function. For any model which can be fit by both `lmeKin` and `lmer`, the latter routine would normally be preferred due to a much wider selection of post-fit tools for residuals, prediction, plotting, etc.

**Value**

An object of class `lmeKin`.

**Author(s)**

Terry Therneau

**See Also**

[lmeKin.object](#), [coxme](#)

**Examples**

```
fit1 <- lme(effort ~ Type, data=ergoStool, random= ~1|Subject,
           method="ML")
fit2 <- lmeKin(effort ~ Type + (1|Subject), data=ergoStool)
```

---

`lmeKin.object`

*lmeKin object*

---

**Description**

This class of object is returned by the `lmeKin` function to represent a fitted mixed effect linear model. Objects of this class currently have methods for `print` and `residuals`.

**Details**

The random effects are modeled internally as  $N(0, \sigma^2 A)$  where  $\sigma^2$  is the residual variance. The parameters of `vcoef` refer to  $A$ . The printed results for the model have been multiplied by  $\sigma$ , removing the factorization.

The variance matrix `vvar` is for the transformed parameters. For the details of the transformation see the individual variance functions. The `coxmeFull` routine, for instance, uses a log transform for variances. Use this with caution.

**Value**

A list with the following components:

coefficients	a list with components fixed and random; the first will be NULL for a model with no fixed effects. The random component is itself a list, with an element for each random effect.
var	the variance matrix of the fixed effects
vcoef	the parameters of the variance matrix of the random effects.
vvar	variance matrix for vcoef
residuals	vector of residuals from the fit
method	either "ML" or "REML"
loglik	the log-likelihood for the fitted model
sigma	the estimated residual error
n	number of observations used
call	a copy of the call
na.action	this will be present if any observations were removed due to missing values

**Author(s)**

Terry Therneau

**See Also**

[lmekin](#), [coxmeFull](#), [coxmeMlist](#)

---

print.coxme

*Print method for a coxme fit.*

---

**Description**

Print out the result of a coxme fit.

**Usage**

```
## S3 method for class 'coxme'
print(x, rcoef=FALSE, digits = options()$digits, ...)
```

**Arguments**

x	an object of class coxme, from the fit of a mixed effects survival model.
rcoef	print the random (penalized) coefficients, as well as the fixed ones.
digits	number of significant digits to print
...	optional arguments

**Author(s)**

Terry Therneau

**See Also**

[coxme](#)

---

`print.lmekin`      *Print function for lmekin*

---

**Description**

Print out the result of an `lmekin` fit.

**Usage**

```
## S3 method for class 'lmekin'  
print(x, ...)
```

**Arguments**

`x`                    an object of class `lmekin`.  
`...`                generic arguments to `print`, unused.

**Details**

The `print` function current has no options. This should one day improve.

**Author(s)**

Terry Therneau

**See Also**

[lmekin](#)

# Index

## \*Topic **datasets**

eortc, 8

## \*Topic **models**

anova.coxme, 2

fixef.coxme, 10

fixef.lmekin, 11

lmekin, 12

lmekin.object, 13

print.lmekin, 15

## \*Topic **regression**

anova.coxme, 2

## \*Topic **survival**

anova.coxme, 2

coxme, 3

coxme.control, 5

coxmeFull, 6

coxmeMlist, 7

expand.nested, 9

fixef.coxme, 10

print.coxme, 14

anova, 3

anova.coxme, 2

anova.coxmelist (anova.coxme), 2

coxme, 3, 3, 6, 7, 9, 10, 13, 15

coxme.control, 5

coxmeFull, 4, 6, 14

coxmeMlist, 4, 7, 9, 14

eortc, 8

expand.nested, 9

fixed.effects, 10, 11

fixef.coxme, 10

fixef.lmekin, 11

lmekin, 11, 12, 14, 15

lmekin.object, 13, 13

print.coxme, 14

print.lmekin, 15

random.effects, 10, 11

ranef.coxme (fixef.coxme), 10

ranef.lmekin (fixef.lmekin), 11

VarCorr.coxme (fixef.coxme), 10

VarCorr.lmekin (fixef.lmekin), 11

vcov.coxme (fixef.coxme), 10

vcov.lmekin (fixef.lmekin), 11