# Package ‘multiRec’

February 3, 2026

**Type** Package

**Title** Analysis of Multi-Type Recurrent Events

**Version** 1.0.6

**Maintainer** Alokananda Ghosh <`aghosh@alum.mit.edu`>

**Author** Naji Younes [aut],
Alokananda Ghosh [aut, cre],
Medhasweta Sen [aut]

**Depends** R (>= 4.2.0)

**Description** Implements likelihood-based estimation and diagnostics for multi-type recurrent event data with dynamic risk that depends on prior events and accommodates terminating events. Methods are described in Ghosh, Chan, Younes and Davis (2023) ``A Dynamic Risk Model for Multitype Recurrent Events'' <doi:10.1093/aje/kwac213>.

**Imports** survival, numDeriv, MASS, Rfast

**License** GPL-2

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.3.3

**NeedsCompilation** no

**Suggests** knitr, rmarkdown

**VignetteBuilder** knitr

**Repository** CRAN

**Date/Publication** 2026-02-03 13:20:02 UTC

## Contents

---

## `AIC.multiRec` *Akaike's Information Criterion*

### Description

Akaike's Information Criterion

### Usage

```
## S3 method for class 'multiRec'
AIC(object, ..., k = 2)
```

### Arguments

| | |
|---|---|
| `object` | an object of type multiRec |
| `...` | not used |
| `k` | numeric, the penalty per parameter to be used; the default k = 2 is the classical AIC. |

### Value

a number

---

## `BIC.multiRec` *Bayesian Information Criteron*

### Description

Bayesian Information Criteron

### Usage

```
## S3 method for class 'multiRec'
BIC(object, ...)
```

### Arguments

| | |
|---|---|
| `object` | an object of type multiRec |
| `...` | not used |

### Value

a number

---

## `confint.multiRec` *Confidence Intervals for Model Parameters*

### Description

Confidence Intervals for Model Parameters

### Usage

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

### Arguments

| | |
|---|---|
| `object` | a fitted model object. |
| `parm` | a specification of which parameters are to be given confidence intervals, either a vector of numbers or a vector of names. If missing, all parameters are considered. |
| `level` | the confidence level required. |
| `...` | not used |

### Value

a matrix

---

`deviance.multiRec` *Deviance of a fitted model*

---

## Description

Deviance of a fitted model

## Usage

```
## S3 method for class 'multiRec'
deviance(object, ...)
```

## Arguments

| | |
|---|---|
| `object` | an object of type multiRec |
| `...` | not used |

## Value

a number

---

`logLik.multiRec` *Log likelihood of a fitted model*

---

## Description

Log likelihood of a fitted model

## Usage

```
## S3 method for class 'multiRec'
logLik(object, ...)
```

## Arguments

| | |
|---|---|
| `object` | an object of type multiRec |
| `...` | not used |

## Value

a number

---

## multiRec *Fit the multi-type recurrent event model*

---

### Description

multi-type events, may be recurrent or terminating

### Usage

```
multiRec(
  ...,
  data,
  eventVar = "event",
  startTimeVar = "tstart",
  stopTimeVar = "tstop",
  idVar = "id",
  na.action = na.omit,
  link = c("log", "identity", "yj"),
  SANN.init = 0,
  noEvent = c("", NA),
  robust = FALSE,
  method = c("BFGS", "Nelder-Mead", "CG", "SANN"),
  method.seed = NULL,
  hazardPrefix = "nPrior.",
  maxit,
  trace = FALSE,
  fitDetails = FALSE,
  hessian = c("optim", "numDeriv")
)
```

### Arguments

| | |
|---|---|
| `...` | one or more models (see details) |
| `data` | the dataset |
| `eventVar` | character, the name of the event variable in the data= dataset |
| `startTimeVar` | character, the name of the variable that holds the start time of each interval in the data= dataset |
| `stopTimeVar` | character, the name of the variable that holds the stop time of each interval in the data= dataset |
| `idVar` | character, the name of the id variable which uniquely identifies individuals in the data= dataset |
| `na.action` | function, a function which indicates what should happen when the data contain NAs. The default is na.omit. |
| `link` | the link used to model the hazard |

| | |
|---|---|
| `SANN.init` | integer, if this is greater than 0 it indicates the number of simulated annealing iterations used to refine the initial estimate of the parameters before calling the method specified in method=. |
| `noEvent` | character, the value(s) of the event variable that indicate that no event occurred in an interval |
| `robust` | logical, if TRUE robust standard errors are used |
| `method` | character, the optimization method see the optim function for details |
| `method.seed` | numeric, the seed used |
| `hazardPrefix` | character, the prefix used to denote prior event functions in the formulas for the hazard |
| `maxit` | integer, the maximum number of iteration for the optimization method. Defaults to 500 for Nelder-Mead, 100 for all other methods. |
| `trace` | logical, if TRUE additional information is printed by the algorithm |
| `fitDetails` | logical, if TRUE the returned fit object contains additional information used to create fit diagnostics. |
| `hessian` | character, use the Hessian calculated in optim or calculated separately in numDeriv::hessian(). |

## Details

models for the hazard are of the form

lhs ~ rhs

where lhs is one of the possible events and rhs is the usual model right hand side, with one addition: the pseudo-function "nPrior." can be used to indicate a dependence of the hazard on prior events. This function has the form

nPrior.event(~formula, alpha=value)

where

**event** should be replaced by the appropriate event. For example, "nPrior.stroke" indicates a dependence of the hazard on the number of strokes prior to the current interval

**formula** is a possibly empty formula which expresses the possible covariate effects on the extent to which prior events impact the hazard. For example "nPrior.stroke(~hba1c+sbp)" indicates a dependence on the number of prior strokes which may be modulated by HbA1c (variable hba1c) and systolic blood pressure (variable sbp)

**value** is the power to which the number of prior events should be raised. The default is 1, i.e. the hazard depends simply on the number of prior events. A value of 0.5 means that the hazard depends on the square root of the number of prior events, i.e. that additional events have a diminishing effect on the hazard. Specifying alpha=NA means that the power is unknown and should be estimated.

For example a model of myocardial infaction (event='mi') might be

mi ~ bmi + nPrior.mi(alpha=0.5) + nPrior.stroke(~hba1c+sbp)

to indicate that the hazard of mi depends on body mass index (variable bmi), on the square root of the number of prior MI's and on the number of prior strokes, modulated by hba1c and systolic blood pressure.

## Value

an object of type agMTRE

## Examples

```
# Fit a model for the multiRecCVD2 dataset
fit = multiRec(hf   ~ nPrior.afib(~male),
               afib ~ age + nPrior.afib() + nPrior.hf(),
               data=multiRecCVD2,
               idVar='id',
               link='log')

# Fit the model with robust variances
fit = multiRec(hf   ~ nPrior.afib(~male),
               afib ~ age + nPrior.afib() + nPrior.hf(),
               data=multiRecCVD2,
               idVar='id',
               link='log',
               robust = TRUE)   # Request robust (sandwich) variances

# Use a specified power
fit = multiRec(hf   ~ nPrior.afib(~male),
               afib ~ age + nPrior.afib(alpha=0.5) + nPrior.hf(),
               data=multiRecCVD2,
               idVar='id',
               link='log',
               robust = TRUE)

# Display the coefficients
coef(fit)
confint(fit)

# Display intervals with a poor fit
boxplot(resid(fit))
```

---

# `multiRecCVD2` *Simulated dataset with two types of multi-type recurrent events*

## Description

A dataset containing 1403 rows and 5 variables. Each row represents an interval, with individuals having one or more intervals.

## Usage

```
multiRecCVD2
```

### Format

A data frame with 1403 rows and 8 variables:

**id** integer, a unique participant id

**age** integer, age of the participant in years

**male** integer, 1 if the participant is male, 0 if female

**bmi** numeric, body mass index

**tstart** numeric, the start time for the interval

**tstop** numeric, the stop time for the interval

**event** character, the event that occurred at the end of the interval. this is either the event type ('afib' or 'hf') or an empty string if no event occurred

---

## `multiRecCVD4` *Simulated dataset with four types of multi-type recurrent events*

### Description

A dataset containing 3119 rows and 5 variables. Each row represents an interval, with individuals having one or more intervals.

### Usage

```
multiRecCVD4
```

### Format

A data frame with 1403 rows and 8 variables:

**id** integer, a unique participant id

**age** integer, age of the participant in years

**male** integer, 1 if the participant is male, 0 if female

**bmi** numeric, body mass index

**tstart** numeric, the start time for the interval

**tstop** numeric, the stop time for the interval

**event** character, the event that occurred at the end of the interval. this is either the event type ('afib', 'stroke', 'hf' or 'death') or an empty string if no event occurred

---

## `plotProfiles` *Plot profile likelihoods*

---

### Description

Plot profile likelihoods

### Usage

```
plotProfiles(fit, n = 100, showInitial = FALSE, maxiter = 50, tolerance = 0.1)
```

### Arguments

| | |
|---|---|
| `fit` | list, the fit object from multiRec. The fit object must be generated by calling multiRec with fitDetails=TRUE (the default is FALSE) |
| `n` | integer, the number of points on the profile likelihood plot |
| `showInitial` | logical, if TRUE the initial parameter estimates are shown on the figures as a hollow dot. |
| `maxiter` | integer, the number of iterations used to identify the x axis range on each figure. |
| `tolerance` | numeric, a multiplicative factor used the set the y axis range on each figure: the axis will range from approximately the maximum of the log likelihood to (1+tolerance) times the maximum |

### Details

The function generates a series of plots, one for each parameter in the model. Each plot shows the log likelihood on the vertical axis and a range of values for the parameter on the horizontal axis. The plot for a specific parameter is obtained by fixing the remaining parameters at their maximum likelihood value, and varying the plotted parameter over a range of values. The range of values is selected in such a way that the resulting likelihood ranges from 90

Plotting profile likelihoods is particularly important when using the identity link as it tends to have ill behaved likelihoods. When looking at profile plots, looks for multiple maxima, especially ones for which the corresponding likelihoods are similar in magnitude: if they exist, the interpretation of the fit returned by the model is questionable as there are other solutions that fit the data (almost) as well. Also look for discontinuities and sharp changes in slope (i.e discontinuities in the first derivatives) as these will often cause convergence problems.

### Value

No return value; called for side effects (creates profile likelihood plots).

### Examples

```
fit = multiRec(hf   ~ nPrior.afib(~male),
               afib ~ age + nPrior.afib() + nPrior.hf(),
               data=multiRecCVD2,
               idVar='id',
               link='log',
               fitDetails=TRUE)
plotProfiles(fit)
```

---

## `print.multiRec` *Print a short summary of the model fit*

---

### Description

Print a short summary of the model fit

### Usage

```
## S3 method for class 'multiRec'
print(x, digits = 4, ...)
```

### Arguments

| | |
|---|---|
| `x` | an object of type multiRec |
| `digits` | integer, the number of decimal digits to display for parameters. |
| `...` | not used |

### Value

Invisibly returns the input object `x`. Called for side effects (printing).

---

## `resid.multiRec` *Residuals from a fitted model*

---

### Description

This returns analogs of martingale residuals, as a matrix with one column for each hazard.

### Usage

```
## S3 method for class 'multiRec'
resid(object, ..., type = c("martingale", "score"))
```

## Arguments

| | |
|---|---|
| `object` | an object of type multiRec |
| `...` | not used |
| `type` | character, the type of residuals. The default is "martingale" for an analog of martingale residuals in coxph. Specifying "score" retrieves a matrix score components (see details). |

## Details

This function returns a matrix with one row for each observation (i.e. interval) in the input data= dataset, and one column for each hazard model in the call to multiRec().

For each hazard model, the martingale-like residuals are defined as the difference delta.i-cumHaz.i where delta.i is an indicator for whether the corresponding event occurred and cumHaz.i is the estimate of the event specific cumulative hazard.

For each hazard model, the score residual is the component of the score vector.

## Value

a numeric matrix with one row per interval and one column per hazard

---

# `vcov.multiRec` *Variance covariance matrix of a fitted model*

## Description

If the model was fitted using robust=TRUE, this is the robust variance. Otherwise it is the naive variance (i.e. the inverse of the information matrix)

## Usage

```
## S3 method for class 'multiRec'
vcov(object, ...)
```

## Arguments

| | |
|---|---|
| `object` | an object of type multiRec |
| `...` | not used |

## Value

a matrix

# Index