

Package ‘reda’

October 29, 2016

Title Recurrent Event Data Analysis

Version 0.2.1.9000

Date 2016-10-29

Description Functions that fit gamma frailty model with spline or piecewise constant baseline rate function for recurrent event data, compute and plot parametric mean cumulative function (MCF) from a fitted model as well as nonparametric sample MCF (Nelson-Aalen estimator) are provided. Most functions are S4 methods that produce S4 class objects.

Imports ggplot2, grDevices, methods, plyr, splines2, stats, utils

Suggests knitr

Depends R (>= 3.2.3)

License GPL (>= 3)

LazyData true

VignetteBuilder knitr

Collate 'class.R' 'aic.R' 'baseline.R' 'coef.R' 'data.R' 'dataCheck.R' 'fit.R' 'mcf.R' 'plot.R' 'reda.R' 'show.R' 'summary.R'

URL <https://github.com/wenjie2wang/reda>

BugReports <https://github.com/wenjie2wang/reda/issues>

RoxygenNote 5.0.1

NeedsCompilation no

Author Wenjie Wang [aut, cre],
Haoda Fu [aut],
Jun Yan [ctb]

Maintainer Wenjie Wang <wenjie.2.wang@uconn.edu>

R topics documented:

reda-package	2
AIC,rateReg-method	2
baseRate	3
BIC,rateReg-method	4
coef,rateReg-method	5
confint,rateReg-method	5
mcf	6

plotMcf	9
rateReg	10
rateReg-class	13
rateRegMcf-class	14
sampleMcf-class	15
show-method	15
simuDat	16
summary,rateReg-method	17
summaryRateReg-class	18
Surv	18
Index	20

reda-package	<i>Recurrent Event Data Analysis</i>
--------------	--------------------------------------

Description

The package **reda** mainly provides function [rateReg](#) to fit parametric gamma frailty model with spline or piecewise constant baseline rate function. Another main function [mcf](#) computes and plots the parametric mean cumulative function (MCF) from a fitted model as well as the nonparametric sample MCF (Nelson-Aelson estimator) for recurrent event data.

Details

See vignettes for introduction and demonstration.

AIC,rateReg-method	<i>Akaike Information Criterion (AIC)</i>
--------------------	---

Description

AIC,rateReg-method is an S4 class method calculating Akaike information criterion (AIC) for one or several rateReg-class objects, according to the formula $-2 * \log\text{-likelihood} + 2 * nPar$, where nPar represents the number of parameters in the fitted model.

Usage

```
## S4 method for signature 'rateReg'
AIC(object, ..., k = 2)
```

Arguments

- | | |
|--------|--|
| object | An object used to dispatch a method. |
| ... | Optionally more fitted model objects. |
| k | An optional numeric value used as the penalty per parameter. The default $k = 2$ is the classic AIC. |

Details

When comparing models fitted by maximum likelihood to the same data, the smaller the AIC, the better the fit. `help(AIC, stats)` for other details.

Value

If just one object is provided, a numeric value representing calculated AIC. If multiple objects are provided, a data frame with rows corresponding to the objects and columns `df` and `AIC`, where `df` means degree of freedom, which is the number of parameters in the fitted model.

See Also

[rateReg](#) for model fitting; [summary, rateReg-method](#) for summary of a fitted model; [BIC, rateReg-method](#) for BIC.

Examples

```
## See examples given in function rateReg.
```

baseRate	<i>Estimated Coefficients of Baseline Rate Function</i>
----------	---

Description

An S4 class generic function that returns the estimated coefficients of baseline rate function. For [rateReg-class](#) object, it returns either coefficients of pieceswise (including one piece) constant rate function or coefficients of B-spline bases.

Usage

```
baseRate(object, ...)

## S4 method for signature 'rateReg'
baseRate(object, ...)
```

Arguments

<code>object</code>	An object used to dispatch a method.
<code>...</code>	Other arguments for future usage.

Value

A named numeric vector.

Methods (by class)

- `rateReg`: Extract estimated coefficients of baseline rate function from `rateReg-class` object.

See Also

[rateReg](#) for model fitting; [summary, rateReg-method](#) for summary of a fitted model.

Examples

```
## See examples given in function rateReg.
```

BIC,rateReg-method	<i>Bayesian Information Criterion (BIC)</i>
--------------------	---

Description

BIC,rateReg-method is an S4 class method calculating Bayesian information criterion (BIC) or so-called Schwarz's Bayesian criterion (SBC) for one or several rateReg-class objects, according to the formula $-2 * \log\text{-likelihood} + \ln(n\text{Obs}) * n\text{Par}$, where nPar represents the number of parameters in the fitted model and nObs is the number of observations.

Usage

```
## S4 method for signature 'rateReg'
BIC(object, ...)
```

Arguments

object	An object used to dispatch a method.
...	Optionally more fitted model objects.

Details

When comparing models fitted by maximum likelihood to the same data, the smaller the BIC, the better the fit. `help(BIC, stats)` for other details.

Value

If just one object is provided, a numeric value representing calculated BIC. If multiple objects are provided, a data frame with rows corresponding to the objects and columns df and BIC, where df means degree of freedom, which is the number of parameters in the fitted model.

See Also

[rateReg](#) for model fitting; [summary,rateReg-method](#) for summary of a fitted model; [AIC,rateReg-method](#) for AIC.

Examples

```
## See examples given in function rateReg.
```

coef,rateReg-method *Estimated Coefficients of Covariates*

Description

coef,rateReg-method is a S4 class method that extracts estimated coefficients of covariates from [rateReg-class](#) object produced by function [rateReg](#).

Usage

```
## S4 method for signature 'rateReg'
coef(object, ...)
```

Arguments

object [rateReg-class](#) object.
 ... Other arguments for future usage.

Value

A named numeric vector.

See Also

[rateReg](#) for model fitting; [confint,rateReg-method](#) for confidence intervals for covariate coefficients; [summary,rateReg-method](#) for summary of a fitted model.

Examples

```
## See examples given in function rateReg.
```

confint,rateReg-method *Confidence Intervals for Covariate Coefficients*

Description

confint,rateReg-method is a S4 class method for [rateReg](#) object, which returns approximate confidence intervals for all or specified covariates.

Usage

```
## S4 method for signature 'rateReg'
confint(object, parm, level = 0.95, ...)
```

Arguments

<code>object</code>	rateReg-class object.
<code>parm</code>	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.
<code>level</code>	An optional numeric value to specify the confidence level required. By default, the value is 0.95, which produces 95% confidence intervals.
<code>...</code>	Other arguments for future usage.

Details

Under regularity condition (Shao, 2003, Theorem 4.16 and Theorem 4.17, page 287, 290), the approximate confidence intervals are constructed loosely based on Fisher information matrix and estimates of coefficients. See Fu et al. (2014) for details also.

Value

A numeric matrix with rownames and colnames.

References

Fu, H., Luo, L., & Qu Y. (2014). Hypoglycemic Events Analysis via Recurrent Time-to-Event (HEART) Models. *Journal of biopharmaceutical statistics*, Epub 2014 Dec 1.

Shao, J. (2003), *Mathematical statistics*, Springer texts in statistics, New York: Springer, 2nd Edition.

See Also

[rateReg](#) for model fitting; [coef,rateReg-method](#) for point estimates of covariate coefficients; [summary,rateReg-method](#) for summary of a fitted model.

Examples

```
## See examples given in function rateReg.
```

mcf

Mean Cumulative Function (MCF)

Description

An S4 class generic function that estimates mean cumulative function (MCF) from a fitted model or computing the sample nonparametric MCF (also called Nelson-Aalen estimator) from data.

Usage

```
mcf(object, ...)

## S4 method for signature 'formula'
mcf(object, data, subset, na.action, level = 0.95, ...)

## S4 method for signature 'rateReg'
mcf(object, newdata, groupName, groupLevels, level = 0.95,
     na.action, control = list(), ...)
```

Arguments

<code>object</code>	An object used to dispatch a method.
<code>...</code>	Other arguments for future usage.
<code>data</code>	An optional data frame, list or environment containing the variables in the model. If not found in data, the variables are taken from <code>environment(formula)</code> , usually the environment from which the function is called.
<code>subset</code>	An optional vector specifying a subset of observations to be used in the fitting process.
<code>na.action</code>	A function that indicates what should the procedure do if the data contains NAs. The default is set by the <code>na.action</code> setting of <code>options</code> . The "factory-fresh" default is <code>na.omit</code> . Other possible values include <code>na.fail</code> , <code>na.exclude</code> , and <code>na.pass</code> . <code>help(na.fail)</code> for details.
<code>level</code>	An optional numeric value indicating the confidence level required. The default value is 0.95.
<code>newdata</code>	An optional data frame. If specified, the data frame should have the same column names as the covariate names appearing in the formula of original fitting.
<code>groupName</code>	An optional length-one character vector to specify the name for grouping each unique row in <code>newdata</code> , such as "gender" for "male" and "female". The default value is "group".
<code>groupLevels</code>	An optional character vector to specify the levels for each unique row in <code>newdata</code> , such as "treatment" and "control". The default values are capital letters starting from "A".
<code>control</code>	An optional list to specify the time grid where the MCF are estimated. The available elements of the control list include <code>grid</code> , <code>length.out</code> , <code>from</code> and <code>to</code> . The time grid can be directly specified via element <code>grid</code> . A dense time grid is suggested. Element <code>length.out</code> represents the length of grid points. The default value is 1,000. Element <code>from</code> means the starting point of grid with default 0. Element <code>to</code> represents the endpoint of grid with the right boundary knot as default. When <code>grid</code> is missing, the grid will be generated by <code>seq</code> (from package base) with arguments <code>from</code> , <code>to</code> and <code>length.out</code> .

Details

For `formula` object with `Surv` object as response, the covariate specified at the right hand side of the formula should be either 1 or any one factor variable in the data. The former computes the overall sample MCF. The latter computes the sample MCF for each level of the factor variable specified, respectively. The sample MCF is also called Nelson-Aalen nonparametric estimator (Nelson, 2003) and computed on each time point from sample data. The point estimate of sample MCF at each time point does not assume any particular underlying model. The variance of estimated MCF (ReliaWiki,

2012) at each time point is estimated and the approximate confidence intervals are provided as well, which are constructed based on the asymptotic normality of log mean cumulative function.

For `rateReg-class` object, `mcf` estimates the baseline MCF and its confidence interval at each time grid if argument `newdata` is not specified. Otherwise, `mcf` estimates MCF and its confidence interval for the given `newdata` based on Delta-method.

Value

`sampleMcf-class` or `rateRegMcf-class` object. Their slots include

- `level`: Confidence level specified.
- `MCF`: Mean cumulative function at each time point.
- `multiGroup`: A logical value indicating whether MCF is estimated for different specified group.
- `newdata`: Given dataset used to estimate MCF.

For the meaning of other slots, see `rateReg`.

Methods (by class)

- `formula`: Sample MCF from data.
- `rateReg`: Estimated MCF from a fitted model.

References

Nelson, W. B. (2003). *Recurrent events data analysis for product repairs, disease recurrences, and other applications* (Vol. 10). SIAM.

ReliaWiki. (2012, March 19). Recurrent Event Data Analysis. Retrieved November 23, 2015, from http://reliawiki.org/index.php/Recurrent_Event_Data_Analysis

See Also

`rateReg` for model fitting; `plotMcf` for plotting MCF.

Examples

```
library(redda)

## sample MCF
sampleMcf <- mcf(Surv(ID, time, event) ~ group,
                 data = simuDat, subset = ID %in% 1:10)

## plot sample MCF
plotMcf(sampleMcf, lty = c(1, 3), col = c("orange", "navy"),
        mark.time = TRUE) + ggplot2::xlab("Days") + ggplot2::theme_bw()

## For estimated MCF from a fitted model,
## see examples given in function rateReg.
```


plotMcf

*Plot Mean Cumulative Function (MCF)***Description**

An S4 class generic function dispatched to a certain method to plot mean cumulative function by using ggplot2 plotting system. The plots generated are able to be further customized properly.

Usage

```
plotMcf(object, conf.int = FALSE, ...)

## S4 method for signature 'sampleMcf'
plotMcf(object, conf.int = FALSE, mark.time = FALSE,
        lty, col, ...)

## S4 method for signature 'rateRegMcf'
plotMcf(object, conf.int = FALSE, lty, col, ...)
```

Arguments

object	An object used to dispatch a method.
conf.int	A logical value indicating whether to plot confidence interval. The default value is FALSE.
...	Other arguments for further usage.
mark.time	A logical value with default FALSE. If TRUE, each censoring time is marked by "+" on the MCF curves. Otherwise, the censoring time would not be marked.
lty	An optional numeric vector indicating line types specified to different groups: 0 = blank, 1 = solid, 2 = dashed, 3 = dotted, 4 = dotdash, 5 = longdash, 6 = twodash.
col	An optional character vector indicating line colors specified to different groups.

Value

A ggplot object.

Methods (by class)

- sampleMcf: Plot sample MCF from data.
- rateRegMcf: Plot estimated MCF from a fitted model.

See Also

[mcf](#) for estimation of MCF; [rateReg](#) for model fitting.

Examples

```
## See examples given in function mcf and rateReg.
```

Description

The default model is the gamma frailty model with one piece constant baseline rate function, which is equivalent to negative binomial regression of the same shape and rate parameter in gamma prior. Spline and piecewise constant baseline rate function can be specified and applied to model fitting instead. `rateReg` returns the fitted model through a `rateReg-class` object.

Usage

```
rateReg(formula, df = NULL, knots = NULL, degree = 0L, data, subset,
        na.action, start = list(), control = list(), contrasts = NULL, ...)
```

Arguments

<code>formula</code>	Survvr object produced by function <code>Survvr</code> .
<code>df</code>	An optional nonnegative integer to specify the degree of freedom of baseline rate function. If argument <code>knots</code> or <code>degree</code> are specified, <code>df</code> will be neglected whether it is specified or not.
<code>knots</code>	An optional numeric vector that represents all the internal knots of baseline rate function. The default is <code>NULL</code> , representing no any internal knots.
<code>degree</code>	An optional nonnegative integer to specify the degree of spline bases.
<code>data</code>	An optional data frame, list or environment containing the variables in the model. If not found in <code>data</code> , the variables are taken from <code>environment(formula)</code> , usually the environment from which function <code>rateReg</code> is called.
<code>subset</code>	An optional vector specifying a subset of observations to be used in the fitting process.
<code>na.action</code>	A function that indicates what should the procedure do if the data contains NAs. The default is set by the <code>na.action</code> setting of <code>options</code> . The "factory-fresh" default is <code>na.omit</code> . Other possible values include <code>na.fail</code> , <code>na.exclude</code> , and <code>na.pass</code> . <code>help(na.fail)</code> for details.
<code>start</code>	An optional list of starting values for the parameters to be estimated in the model. See more in section details.
<code>control</code>	An optional list of parameters to control the maximization process of negative log likelihood function and adjust the baseline rate function. See more in section details.
<code>contrasts</code>	An optional list, whose entries are values (numeric matrices or character strings naming functions) to be used as replacement values for the contrasts replacement function and whose names are the names of columns of data containing factors. See <code>contrasts.arg</code> of <code>model.matrix.default</code> for details.
<code>...</code>	Other arguments for future usage.

Details

Function `Surv` in the formula response first checks the dataset and will report an error if the dataset does not fall into recurrent event data framework. Subject's ID is pinpointed if its observation violates any checking rule. See `Surv` for all the checking rules.

Function `rateReg` first constructs the design matrix from the specified arguments: `formula`, `data`, `subset`, `na.action` and `constrasts` before model fitting. The constructed design matrix will be checked again to fit the recurrent event data framework if any observation with missing covariates is removed.

The model fitting process involves minimization of negative log likelihood function, which calls function `nlm` from package `stats` internally. `help(nlm)` for more details.

The argument `start` is an optional list that allows users to specify the initial guess for the parameter values for the minimization of negative log likelihood function. The available numeric vector elements in the list include

- `beta`: Coefficient(s) of covariates, set to be 0.1 by default.
- `theta`: Parameter of frailty random effect, set to be 0.5 by default.
- `alpha`: Coefficient(s) of baseline rate function, set to be 0.05 by default.

The argument `control` is an optional list that allows users to control the process of minimization of negative log likelihood function and to specify the boundary knots, intercept for baseline rate function. The available elements in the list include

- `gradtol`: A positive scalar giving the tolerance at which the scaled gradient is considered close enough to zero to terminate the algorithm. The default value is $1e-6$.
- `stepmax`: A positive scalar that gives the maximum allowable scaled step length. The default value is $1e5$.
- `steptol`: A positive scalar providing the minimum allowable relative step length. The default value is $1e-6$.
- `iterlim`: A positive integer specifying the maximum number of iterations to be performed before the program is terminated. The default value is $1e2$.
- `Boundary.knots`: A length-two numeric vector to specify the boundary knots for baseline rate function. By default, the left boundary knot is zero and the right one takes the largest censoring time from data.
- `intercept`: A logical value specifying whether intercept is included in spline baseline rate function. For piecewise constant baseline (`df=0`), the specified value would be neglected. The default value is `TRUE`, i.e. the intercept is included.

Value

A `rateReg-class` object, whose slots include

- `call`: Function call of `rateReg`.
- `formula`: Formula used in the model fitting.
- `nObs`: Number of observations.
- `knots`: Internal knots specified for the baseline rate function.
- `Boundary.knots`: Boundary knots specified for the baseline rate function.
- `degree`: Degree of spline bases specified in baseline rate function.
- `df`: Degree of freedom of the model specified.

- `estimates`: Estimated coefficients of covariates and baseline rate function, and estimated rate parameter of gamma frailty variable.
- `control`: The control list specified for model fitting.
- `start`: The initial guess specified for the parameters to be estimated.
- `na.action`: The procedure specified to deal with missing values in the covariate.
- `xlevels`: A list that records the levels in each factor variable.
- `contrasts`: Contrasts specified and used for each factor variable.
- `convergCode`: code returned by function `nlm`, which is an integer indicating why the optimization process terminated. `help(nlm)` for details.
- `logL`: Log likelihood of the fitted model.
- `fisher`: Observed Fisher information matrix.

References

Fu, H., Luo, L., & Qu Y. (2014). Hypoglycemic Events Analysis via Recurrent Time-to-Event (HEART) Models. *Journal of biopharmaceutical statistics*, Epub 2014 Dec 1.

See Also

[summary, rateReg-method](#) for summary of fitted model; [coef, rateReg-method](#) for estimated covariate coefficients; [confint, rateReg-method](#) for confidence interval of covariate coefficients; [baseRate, rateReg-method](#) for estimated coefficients of baseline rate function; [mcf, rateReg-method](#) for estimated MCF from a fitted model; [plotMcf, rateRegMcf-method](#) for plotting estimated MCF.

Examples

```
library(rede)

## constant rate function
constFit <- rateReg(Surv(ID, time, event) ~ group + x1,
  data = simuDat, subset = ID %in% 1:50)

## 6 pieces' piecewise constant rate function
piecesFit <- rateReg(Surv(ID, time, event) ~ group + x1,
  data = simuDat, subset = ID %in% 1:50,
  knots = seq(28, 140, by = 28))

## fit rate function with cubic spline
splineFit <- rateReg(Surv(ID, time, event) ~ group + x1,
  data = simuDat, subset = ID %in% 1:50,
  knots = c(56, 84, 112), degree = 3)

## brief summary of fitted models
constFit
piecesFit
splineFit

## more specific summary
summary(constFit)
summary(piecesFit)
summary(splineFit)

## model selection based on AIC or BIC
```

```

AIC(constFit, piecesFit, splineFit)
BIC(constFit, piecesFit, splineFit)

## estimated covariate coefficients
coef(piecesFit)
coef(splineFit)

## confidence intervals for covariate coefficients
confint(piecesFit)
confint(splineFit, "x1", 0.9)
confint(splineFit, 1, 0.975)

## estimated coefficients for baseline rate function
baseRate(piecesFit)
baseRate(splineFit)

## estimated baseline mean cumulative function (MCF) from a fitted model
piecesMcf <- mcf(piecesFit)
plotMcf(piecesMcf, conf.int = TRUE, col = "blueviolet") +
  ggplot2::xlab("Days") + ggplot2::theme_bw()

## estimated MCF for given new data
newDat <- data.frame(x1 = rep(0, 2), group = c("Treat", "Contr"))
splineMcf <- mcf(splineFit, newdata = newDat, groupName = "Group",
  groupLevels = c("Treatment", "Control"))
plotMcf(splineMcf, conf.int = TRUE, lty = c(1, 5)) +
  ggplot2::xlab("Days") + ggplot2::theme_bw()

```

rateReg-class

An S4 Class to Represent a Fitted Model

Description

rateReg-class is an S4 class that represents a fitted model. [rateReg](#) produces objects of this class. See “Slots” for details.

Slots

call Function call.
 formula Formula.
 nObs A positive integer
 knots A numeric vector.
 Boundary.knots A numeric vector.
 degree A nonnegative integer.
 df A list of nonnegative numeric vectors.
 estimates A list.
 control A list.
 start A list.
 na.action A length-one character vector.

`xlevels` A list.
`contrasts` A list.
`convergCode` A nonnegative integer.
`logL` A numeric value.
`fisher` A numeric matrix.

See Also

[rateReg](#) for details of slots.

rateRegMcf-class	<i>An S4 Class to Represent Estimated MCF from a Fitted Model</i>
------------------	---

Description

An S4 class that represents estimated mean cumulative function (MCF) from Models. [mcf](#) produces objects of this class.

Slots

`call` Function call.
`formula` Formula.
`knots` A numeric vector.
`degree` A nonnegative integer.
`Boundary.knots` A numeric vector.
`newdata` A numeric matrix.
`MCF` A data frame.
`level` A numeric value between 0 and 1.
`na.action` A length-one character vector.
`control` A list.
`multiGroup` A logical value.

See Also

[mcf](#), [rateReg-method](#) for details of slots.

sampleMcf-class

*An S4 Class to Represent Sample MCF***Description**

An S4 class that represents sample mean cumulative function (MCF) from data. [mcf](#) produces objects of this class.

Slots

call Function call
 formula Formula.
 na.action A length-one character vector.
 level A numeric value.
 MCF A data frame.
 multiGroup A logical value.

See Also

[mcf](#), [formula-method](#) for details of slots.

show-method

*Show an object.***Description**

An S4 class generic function that displays certain object.

Usage

```
## S4 method for signature 'rateReg'
show(object)

## S4 method for signature 'summaryRateReg'
show(object)

## S4 method for signature 'sampleMcf'
show(object)

## S4 method for signature 'rateRegMcf'
show(object)
```

Arguments

object An object used to dispatch a method.

Details

- For `rateReg-class` object, it prints out brief summary of the fitted model.
- For `summaryRateReg-class` object, it prints out summary of the fitted model.
- For `sampleMcf-class` object, it prints out the function call, formula and the sample MCF data frame.
- For `rateRegMcf-class` object, it prints formula, new data, confidence level, and the estimated MCF data frame.

See Also

`rateReg` for model fitting; `summary, rateReg-method` for summary of a fitted model; `mcf` for estimation of MCF.

simuDat

Simulated Sample Dataset for Demonstration

Description

A simulated data frame with covariates named ID, time, event, group and x1, where

- ID: Subjects identification;
- time: Event or censoring time;
- event: Event indicator, 1 = event, 0 = censored;
- group: Treatment group indicator;
- x1: Continuous variable.

Format

A data frame with 500 rows and 5 variables.

Details

The sample dataset is originally simulated by the thinning method developed by Lewis and Shedler (1979) and further processed for a better demonstration purpose. See Fu et al. (2014) for details also.

References

- Lewis, P. A., & Shedler, G. S. (1979). Simulation of nonhomogeneous Poisson processes by thinning. *Naval Research Logistics Quarterly*, 26(3), 403–413.
- Fu, H., Luo, L., & Qu Y. (2014). Hypoglycemic Events Analysis via Recurrent Time-to-Event (HEART) Models. *Journal of biopharmaceutical statistics*, Epub 2014 Dec 1.

summary,rateReg-method

Summarizing a Fitted Model

Description

Summary of estimated coefficients of covariates, rate function bases, and estimated rate parameter of frailty random variable, etc., which can be printed out by show.

Usage

```
## S4 method for signature 'rateReg'
summary(object, showCall = TRUE, showKnots = TRUE, ...)
```

Arguments

object	rateReg-class object.
showCall	A logic value with default TRUE, indicating whether function show prints out the original call information of rateReg. It may be helpful for a more concise printout.
showKnots	A logic value with default TRUE, indicating whether function show prints out the internal and boundary knots. Similar to argument showCall, It may be helpful for a more concise printout.
...	Other arguments for future usage.

Details

summary,rateReg-method returns a [summaryRateReg-class](#) object, whose slots include

- covarCoef: Estimated covariate coefficients.
- frailtyPar: Estimated rate parameter of gamma frailty.
- baseRateCoef: Estimated coefficients of baseline rate function.

For the meaning of other slots, see [rateReg](#).

Value

summaryRateReg-class object

See Also

[rateReg](#) for model fitting; [coef,rateReg-method](#) for point estimates of covariate coefficients; [confint,rateReg-method](#) for confidence intervals of covariate coefficients; [baseRate,rateReg-method](#) for coefficients of baseline rate function.

Examples

```
## See examples given in function rateReg.
```

summaryRateReg-class *An S4 Class to Represent Summary of a Fitted Model*

Description

summaryRateReg-class is an S4 class with selective slots of rateReg-class object. See “Slots” for details. [summary, rateReg-method](#) produces objects of this class.

Slots

call Function call.
 knots A numeric vector.
 Boundary.knots A numeric vector.
 covarCoef A numeric matrix.
 frailtyPar A numeric matrix.
 degree A nonnegative integer.
 baseRateCoef A numeric matrix.
 logL A numeric value.

See Also

[summary, rateReg-method](#) for details of slots.

Surv *Formula Response for Recurrent Event Data*

Description

Surv is an S3 class that represents formula response for recurrent event data modeled by methods based on counts and rate function. The last letter ‘r’ in ‘Surv’ represents ‘rate’.

Usage

```
Surv(ID, time, event)
```

Arguments

ID	Identifier of each subject.
time	Time of recurrence event or censoring.
event	The status indicator, 0 = censored, 1 = event.

Details

This is a similar function to `Surv` in package **survrec** but with a better embedded checking procedure for recurrent event data modeled by methods based on counts and rate function. The checking rules include that

- Identification of each subject cannot be missing.
- Event indicator must be coded as 0 (censored) or 1 (event).
- Event time and censoring time cannot be missing.
- Each subject must have one and only one censoring time.
- Event time cannot not be later than censoring time.

See Also

[rateReg](#) for model fitting.

Index

AIC, `rateReg`-method, 2

`baseRate`, 3

`baseRate`, `rateReg`-method (`baseRate`), 3

BIC, `rateReg`-method, 4

`coef`, `rateReg`-method, 5

`confint`, `rateReg`-method, 5

`mcf`, 2, 6, 9, 14–16

`mcf`, `formula`-method (`mcf`), 6

`mcf`, `rateReg`-method (`mcf`), 6

`model.matrix.default`, 10

`na.exclude`, 7, 10

`na.fail`, 7, 10

`na.omit`, 7, 10

`na.pass`, 7, 10

`nlm`, 11, 12

`options`, 7, 10

`plotMcf`, 8, 9

`plotMcf`, `rateRegMcf`-method (`plotMcf`), 9

`plotMcf`, `sampleMcf`-method (`plotMcf`), 9

`rateReg`, 2–6, 8–10, 10, 13, 14, 16, 17, 19

`rateReg`-class, 13

`rateRegMcf`-class, 14

`reda`-package, 2

`sampleMcf`-class, 15

`show`, `rateReg`-method (`show`-method), 15

`show`, `rateRegMcf`-method (`show`-method), 15

`show`, `sampleMcf`-method (`show`-method), 15

`show`, `summaryRateReg`-method
(`show`-method), 15

`show`-method, 15

`simuDat`, 16

`summary`, `rateReg`-method, 17

`summaryRateReg`-class, 18

`Surv`, 7, 10, 11, 18