

# Package ‘reReg’

October 31, 2015

**Type** Package

**Title** Recurrent Event Regression

**Version** 1.0-0

**Date** 2015-10-30

**Author** Sy Han (Steven) Chiou

**Maintainer** Sy Han (Steven) Chiou <schiou@hsph.harvard.edu>

**Description** A collection of regression models for recurrent event process and failure time.

**License** GPL (>= 3)

**LazyLoad** yes

**Imports** BB, MASS, SQUAREM, aftgee, survival

**NeedsCompilation** yes

**Repository** CRAN

**Date/Publication** 2015-10-31 15:52:02

## R topics documented:

reReg-package . . . . .	2
plot.reReg . . . . .	2
readmission . . . . .	4
reReg . . . . .	5
reSurv . . . . .	7
<b>Index</b>	<b>9</b>

---

reReg-package

*Recurrent event regression*

---

### Description

Recurrent event data arise frequently in various fields such as biomedical sciences, public health, engineering, and social sciences. In many instances, the observation of the recurrent event process can be stopped by the occurrence of a correlated failure event, such as death or treatment failure. As such, we implement a collection of regression models for recurrent event process and failure time.

### Details

Package: reReg Type: Package Version: 1.0-0 Date: 2015-10-30 License: GPL(>=3) LazyLoad: yes

### Author(s)

Sy Han Chiou <schiou@hsph.harvard.edu>

### References

Wang, M.C., Qin, J., and Chiang, C.T. (2001). Analyzing Recurrent Event Data with Informative Censoring. *Journal of the American Statistical Association* **96**(455): 1057–1065.

Ghosh, D. and D.Y. Lin (2003). Semiparametric Analysis of Recurrent Events Data in the Presence of Dependent Censoring. *Biometrics*, **59**: 877 – 885.

Huang, C.Y. and Wang, M.C. (2004). Joint Modeling and Estimation for Recurrent Event Processes and Failure Time Data. *Journal of the American Statistical Association* **99**(468), 1153–1165.

---

plot.reReg

*Plot a reReg Object.*

---

### Description

Displays graph of the estimated baseline cumulative rate function and hazard function.

### Usage

```
## S3 method for class 'reReg'  
plot(x, se = FALSE, B = 200, breaks = 1000, ...)
```

**Arguments**

x	an object of class reReg, usually returned by the reReg function.
se	a logical value, if TRUE and x is estimated with joint scale-change model, standard error will be estimated via a efficient resampling-based procedure. Empirical pointwise 95% confidence intervals will be displayed as well.
B	a numeric value specifies the resampling number for variance estimation when se == TRUE.
breaks	a numeric value specifies the number of breakpoints.
...	for future methods

**Value**

a list containing the coordinates of the breakpoints on each of the curves as well as the empirical pointwise confidence bounds. The output contains the following components:

x	a vector giving the breakpoints used in graph.
ly	a vector giving the estimated baseline cumulative rate function evaluated at x
lyU	a vector giving the empirical upper bound for baseline cumulative rate function evaluated at x
lyL	a vector giving the empirical lower bound for baseline cumulative rate function evaluated at x
hy	a vector giving the estimated baseline cumulative hazard function evaluated at x
hyU	a vector giving the empirical upper bound for baseline cumulative hazard function evaluated at x
hyL	a vector giving the empirical lower bound for baseline cumulative hazard function evaluated at x

**References**

Wang, M.C., Qin, J., and Chiang, C.T. (2001). Analyzing Recurrent Event Data with Informative Censoring. *Journal of the American Statistical Association* **96**(455): 1057–1065.

Huang, C.Y. and Wang, M.C. (2004). Joint Modeling and Estimation for Recurrent Event Processes and Failure Time Data. *Journal of the American Statistical Association* **99**(468): 1153–1165.

**See Also**

[reReg](#)

**Examples**

```
## readmission data
data(readmission)
fit <- reReg(reSurv(t.start, id, death, t.stop) ~ sex + dukes + chemo, data = readmission, B = 0)
plot(fit)

## simulation data
```

```

simDat <- function(n, a, b, tau) {
  z <- rgamma(n, 2, 0.2)
  X1 <- rbinom(n, 1, 0.5)
  X2 <- runif(n, 0, 1)
  lambda <- z * (1/10) * exp(cbind(X1, X2) %*% a)
  mt <- rpois(n, tau * lambda)
  eventT <- lapply(sapply(mt, function(x) runif(x) * tau), sort)
  D <- exp(- cbind(X1, X2) %*% b) * sqrt(800 * (-log(runif(n)) / z))
  C <- (X1 == 1) * rexp(n, 0.1) + (X1 == 0) * rexp(n, ifelse(z == 0, 10^-10, z)^2/300)
  Y <- pmin(C, tau, D)
  Delta <- 1 * (D <= Y)
  for (i in 1:n) {
    eventT[[i]] <- c(eventT[[i]][eventT[[i]] < Y[i]], Y[i])
  }
  out <- data.frame(id = rep(1:n, unlist(lapply(eventT, length))),
    T = unlist(eventT),
    X1 = rep(X1, unlist(lapply(eventT, length))),
    X2 = rep(X2, unlist(lapply(eventT, length))),
    Delta = rep(Delta, unlist(lapply(eventT, length))))
  out
}
set.seed(123)
mydat <- simDat(30, a = c(-1, 1), b = c(-1, 1), tau = 10)
(fit.sim <- reReg(reSurv(T, id, Delta) ~ X1 + X2, data = mydat, B = 0))
plot(fit.sim, se = TRUE, B = 10)

```

---

readmission

*Rehospitalization colorectal cancer*


---

## Description

This contains rehospitalization times after surgery in patients diagnosed with colorectal cancer

## Usage

```
data(readmission)
```

## Format

This data frame contains the following columns:

**id** identifier of each subject (repeated for each recurrence)

**enum** observation number within patient

**t.start** start of interval (0 or previous recurrence time)

**t.stop** recurrence or censoring time

**time** interoccurrence or censoring time

**event** rehospitalization status. All event are 1 for each subject excepting last one that it is 0

**chemo** Treatment (chemotherapy) indicator

**sex** Gender

**dukes** Tumour stage under Dukes's classification: A-B, C, or D.

**charlson** Comorbidity Charlson's index, modelled as a time dependent covariate and classified into three groups: Index 0, Index 1-2, and Index  $\geq 3$

**death** death indicator: Dead = 1; alive = 0

### Source

Gonzalez, JR., Fernandez, E., Moreno, V., Ribes, J., Peris, M., Navarro, M., Cambray, M. and Borras, JM (2005). Sex differences in hospital readmission among colorectal cancer patients. *Journal of Epidemiology and Community Health*, **59**, 6, 506-511.

---

reReg	<i>Fits semiparametric regression models for recurrent events and failure time</i>
-------	--

---

### Description

Fits a joint model for the recurrent event process and the failure time that allows the censoring time to be informative about the recurrent event process.

Estimating procedures include a joint scale-change model, Huang-Wang's method (Huang and Wang, 2004), and Ghosh-Lin's method (Ghosh and Lin, 2003). The variance is estimated with an efficient bootstrap approach.

### Usage

```
reReg(formula, data, subset, method = "jsc", B = 200, contrasts = NULL)
```

### Arguments

formula	a formula object, with the response on the left of a '~' operator, and the terms on the right. The response must be a recurrent event survival object as returned by function reSurv.
data	an optional data frame in which to interpret the variables occurring in the 'formula'.
subset	an optional vector specifying a subset of observations to be used in the fitting process.
method	a character string specifying the underlying model and methods to estimate the regression parameter. The following are permitted: 'jsc': joint scale-change model. 'HW': Huang-Wang's method. 'GL': Ghosh-Lin method. Among these, 'jsc' is recommended.
B	a numeric value specifies the resampling number for variance estimation. When B = 0, only the point estimates will be displayed.
contrasts	an optional list.

**Value**

An object of S3 class "reReg" representing the fit, with the following components:

alpha	a vector of point estimates associated with rate function, alpha
beta	a vector of point estimates associated with hazard function, beta
va	estimated covariance matrix for alpha
vb	estimated covariance matrix for beta
lambda	a vector giving the estimated baseline cumulative rate function evaluated at the observation time of recurrence
haz	a vector giving the estimated baseline cumulative hazard function evaluated at the observation time of recurrence
zHat	a vector giving the estimated frailty random variable at the observation time of recurrence

**References**

Ghosh, D. and D.Y. Lin (2003). Semiparametric Analysis of Recurrent Events Data in the Presence of Dependent Censoring. *Biometrics*, **59**: 877 – 885.

Huang, C.Y. and Wang, M.C. (2004). Joint Modeling and Estimation for Recurrent Event Processes and Failure Time Data. *Journal of the American Statistical Association* **99**(468): 1153–1165.

**See Also**

[reSurv](#), [plot.reReg](#)

**Examples**

```
## readmission data
data(readmission)
(fit <- reReg(reSurv(t.start, id, death, t.stop) ~ sex + dukes + chemo,
  data = readmission, B = 0))
summary(fit)
(fit.HW <- reReg(reSurv(t.start, id, death, t.stop) ~ sex + dukes + chemo,
  data = readmission, method = "HW", B = 0))
summary(fit.HW)

## simulation data
simDat <- function(n, a, b, tau) {
  z <- rgamma(n, 2, 0.2)
  X1 <- rbinom(n, 1, 0.5)
  X2 <- runif(n, 0, 1)
  lambda <- z * (1/10) * exp(cbind(X1, X2) %*% a)
  mt <- rpois(n, tau * lambda)
  eventT <- lapply(sapply(mt, function(x) runif(x) * tau), sort)
  D <- exp(- cbind(X1, X2) %*% b) * sqrt(800 * (-log(runif(n)) / z))
  C <- (X1 == 1) * rexp(n, 0.1) + (X1 == 0) * rexp(n, ifelse(z == 0, 10^-10, z)^2/300)
  Y <- pmin(C, tau, D)
  Delta <- 1 * (D <= Y)
  for (i in 1:n) {
```

```

    eventT[[i]] <- c(eventT[[i]][eventT[[i]] < Y[i]], Y[i])
  }
  out <- data.frame(id = rep(1:n, unlist(lapply(eventT, length))),
                  T = unlist(eventT),
                  X1 = rep(X1, unlist(lapply(eventT, length))),
                  X2 = rep(X2, unlist(lapply(eventT, length))),
                  Delta = rep(Delta, unlist(lapply(eventT, length)))
                  )
  out
}
set.seed(123)
mydat <- simDat(50, a = c(-1, 1), b = c(-1, 1), tau = 10)
(fit.sim <- reReg(reSurv(T, id, Delta) ~ X1 + X2, data = mydat, B = 0))
summary(fit.sim)

```

---

reSurv

*Create a Survival Object for Recurrent Event Data*


---

## Description

Create a recurrent event survival object, used as a response variable in reReg model formula.

## Usage

```

reSurv(time1, id, event, time2 = NULL)
is.reSurv(x)

```

## Arguments

id	Observation subject's ID.
time1	When 'time2' is provided, this vector is treated as the starting time for the gap time between two successive recurrent events. In the absence of 'time2', this is the observation time of recurrence on calendar time scale, in which, the time corresponds to the time since entry/inclusion in the study. Furthermore, the last observation time is censored for each subject.
time2	An optional vector for ending time for the gap time between two successive recurrent events.
event	A binary vector used as the status indicator for the terminal event.
x	An reSurv object.

## See Also

[reReg](#)

**Examples**

```
data(readmission)
head(readmission)
(with(readmission[1:10,], reSurv(t.start, id, death, t.stop)))
(with(readmission[1:10,], reSurv(t.stop, id, death)))
```



# Index

\*Topic **datasets**

readmission, 4

\*Topic **plot.reReg**

plot.reReg, 2

reSurv, 7

\*Topic **reReg**

plot.reReg, 2

reReg, 5

reReg-package, 2

reSurv, 7

is.reSurv (reSurv), 7

plot.reReg, 2, 6

readmission, 4

reReg, 3, 5, 7

reReg-package, 2

reSurv, 6, 7