Package: gcmrec (via r-universe)

October 12, 2026

Type Package

Version 2.0.0

Date 2026-08-31

Title General Class of Models for Recurrent Event Data

Description Parameter estimation for the general class of semiparametric models for recurrent event data proposed by Peña and Hollander (2004, <ISBN:978-1-4020-7737-6>). The model incorporates an effective age function encoding the impact of interventions after each event occurrence, the effect of accumulating event occurrences, a link function for possibly time-dependent covariates, and optional gamma frailties to induce dependence among inter-event times. It also fits the extension for cancer relapses of González et al. (2005) <doi:10.1002/sim.2410>. Estimation is performed by profile likelihood, with an expectation-maximization algorithm for the frailty model, and the package provides descriptive, diagnostic and predictive tools for the fitted models.

Depends R (>= 3.5.0)

Imports Rcpp, survival, ggplot2, scales, rlang, stats, graphics, grDevices

LinkingTo Rcpp, RcppArmadillo

Suggests testthat (>= 3.0.0), knitr, rmarkdown, BiocStyle

VignetteBuilder knitr

Config/testthat/edition 3

License GPL (>= 2)

URL https://github.com/isglobal-brge/gcmrec

BugReports https://github.com/isglobal-brge/gcmrec/issues

Encoding UTF-8

LazyData true

RoxygenNote 7.3.3

NeedsCompilation yes

Author Dolors Pelegri-Siso [aut, cre] (ORCID:
<https://orcid.org/0000-0002-5993-3003>), Juan R. González
[aut], Elizabeth H. Slate [aut], Edsel A. Peña [aut]

Maintainer Dolors Pelegri-Siso <dolors.pelegri@isglobal.org>

Repository https://cran.r-universe.dev

Date/Publication 2026-09-12 14:10:22 UTC

RemoteUrl https://github.com/cran/gcmrec

RemoteRef HEAD

RemoteSha 1496baad88fffcad3c75830fa7a4210b51f4df09

Contents

addCenTime . . . 3
anova.gcmrec . . . 4
as_gcmrec_data . . . 5
coef.gcmrec . . . 6
gcmrec . . . 7
gcmrecThreads . . . 9
GeneratedData . . . 10
graph.caltimes . . . 11
hydraulic . . . 12
is.Survr . . . 13
lines.gcmrec . . . 13
List.to.Dataframe . . . 14
logLik.gcmrec . . . 15
lymphoma . . . 15
mcf . . . 16
plot.gcmrec . . . 17
plotBaseline . . . 18
plotForest . . . 19
plotPredict . . . 20
predict.gcmrec . . . 21
print.gcmrec . . . 22
print.summary.gcmrec . . . 23
readmission . . . 24
summary.gcmrec . . . 25
Survr . . . 26
theme_gcmrec . . . 26
vcov.gcmrec . . . 27

Index **28**

addCenTime *Add a censored time equal to 0*

Description

Adds a new row with a censored time equal to 0 for those subjects whose end of follow-up coincides with the last occurrence (i.e. whose last record is an event). `Survr()` requires every subject to have exactly one censored record.

Usage

```
addCenTime(datin, id = 1, time = 2, event = 3)
```

Arguments

`datin`	Data frame containing id, time and event variables; other covariates are allowed and are copied to the added row.
`id`, `time`, `event`	Column positions of the id, time and event variables in `datin`.

Value

A data frame with an added row (censored time equal to 0) for those subjects whose end of follow-up coincides with the last occurrence.

See Also

`Survr()`, `gcmrec()`

Examples

```
# The second subject has no censored record: its follow-up ends at the
# last event
dat <- data.frame(
  id    = c(1, 1, 2, 2),
  time  = c(5, 3, 7, 4),
  event = c(1, 0, 1, 1)
)
addCenTime(dat)
```

`anova.gcmrec` *Likelihood ratio tests for gcmrec models*

Description

Compares nested `gcmrec` models with a likelihood ratio test. The typical use is to ask whether there is real event accumulation (is α different from 1?) or whether a covariate is worth keeping.

Usage

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

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

Arguments

`object`	An object of class `gcmrec`.
`...`	Ignored; kept for compatibility with the generic.
`test`	Ignored; kept for compatibility with the generic.
`x`	An object of class `anova.gcmrec`.
`digits`	Number of significant digits.

Details

Given a single model, the test compares it against the same model with $\rho \equiv 1$ (that is, $\alpha = 1$: occurrences carry no effect), refitting it internally. Given two or more models, they are compared in sequence, each against the previous one, in increasing order of complexity.

Value

A data frame of class `anova.gcmrec` with the log-likelihood, degrees of freedom, chi-square statistic and p-value of each comparison.

For `print`, invisibly `x`.

What can be compared

A likelihood ratio test is only meaningful for **nested models fitted to the same data**, so the following are checked and refused:

Different data all the models must have the same number of subjects and records.

Different effective age models must share the effective age scale, that is, the same `s`, `typeEffage`, `effageData` and `cancer`. A perfect repair fit and a minimal repair one are *not* nested and cannot be tested against each other.

Non-nested covariates the covariates of each model must be a subset of those of the next one.

Frailties the log-likelihood of a frailty fit is conditional on the estimated frailties and is not on the same scale as that of a fit without them, so the two cannot be compared. Judge the frailties from ξ and its jackknife standard error instead.

Comparing two frailty models with each other is allowed but only approximate, since each is conditional on its own estimated frailties; a warning is issued.

See Also

`logLik.gcmrec()`, `gcmrec()`

Examples

```
data(readmission)
mod <- gcmrec(Survr(id, time, event) ~ as.factor(dukes),
              data = readmission, s = 3000)

# Is there event accumulation? (alpha = 1 against alpha free)
anova(mod)

# Is sex worth adding to the model?
mod2 <- gcmrec(Survr(id, time, event) ~ as.factor(dukes) + sex,
               data = readmission, s = 3000)
anova(mod, mod2)
```

---

as_gcmrec_data *Normalize input data for gcmrec*

Description

Generic that converts the supported input formats of `gcmrec()` and `graph.caltimes()` into the canonical `data.frame` layout with `id`, `time` and `event` columns. New input formats can be supported by adding methods to this generic.

Usage

```
as_gcmrec_data(data, ...)

## S3 method for class 'data.frame'
as_gcmrec_data(data, ...)

## S3 method for class 'list'
as_gcmrec_data(data, ...)

## Default S3 method:
as_gcmrec_data(data, ...)
```

Arguments

data	The data in one of the supported formats.
...	Additional arguments passed to methods.

Details

Currently supported inputs:

- data.frame (including tibbles and data.tables): returned as is.
- Legacy list format with elements n and subject (see List.to.Dataframe()): converted to a data frame. Any other list is returned unchanged so it can be used as a variable environment for the model formula, as in base R modelling functions.

Value

A data.frame (or the original object when no conversion applies).

See Also

gcmrec(), List.to.Dataframe()

Examples

```
data(hydraulic)
head(as_gcmrec_data(hydraulic))
```

---

coef.gcmrec *Extract coefficients from a gcmrec model*

Description

Extract coefficients from a gcmrec model

Usage

```
## S3 method for class 'gcmrec'
coef(object, ...)
```

Arguments

object	An object of class gcmrec.
...	Ignored; kept for compatibility with the generic.

Value

Named vector of estimated parameters. With rhoFunc = "alpha to k" the first element is α.

See Also

gcmrec()

---

gcmrec *Fit the general class of models for recurrent event data*

Description

Fits the general semiparametric model for recurrent events proposed by Peña and Hollander (2004). This class of models incorporates an effective age function which encodes the changes that occur after each event occurrence (such as the impact of an intervention), allows for the modelling of the impact of accumulating event occurrences on the unit, admits a link function in which the effect of possibly time-dependent covariates is incorporated, and allows the incorporation of unobservable frailty components which induce dependence among the inter-event times for each unit.

Usage

```
gcmrec(
  formula,
  data,
  effageData = NULL,
  s,
  Frailty = FALSE,
  alphaSeed,
  betaSeed,
  xiSeed,
  tol = 10^(-6),
  maxit = 100,
  rhoFunc = "alpha to k",
  typeEffage = "perfect",
  maxXi = "Newton-Raphson",
  se = "Information matrix",
  cancer = NULL,
  keep.data = TRUE
)
```

Arguments

formula	A formula object with a Survr() object as response (on the left of the ~ operator) and covariate terms on the right. Covariates are required.
data	A data frame containing the variables named in the formula, including id, time and event (see Survr()). Alternatively, a list with elements n and subject in the legacy format described in List.to.Dataframe(); it is converted internally via as_gcmrec_data().
effageData	Optional list with effective age information per subject (elements intercepts, slopes, lastperrep, perrepind, effagebegin, effage). If NULL, effective ages are generated under the perfect or minimal repair model according to typeEffage.

s	A selected calendar time.
Frailty	Logical. If TRUE, the model with gamma frailties is fitted via EM.
alphaSeed, betaSeed, xiSeed	Initial values for α, β and ξ.
tol	Tolerance of the maximization procedures.
maxit	Maximum number of iterations of the maximization procedures.
rhoFunc	Function $\rho(k;\alpha)$ for the effect of accumulating event occurrences: "alpha to k" for $\rho(k;\alpha) = \alpha^k$ (default) or "Identity" for $\rho(k;\alpha) = 1$. Partial matching is allowed.
typeEffage	Effective age model: "perfect" (perfect repair, default) or "minimal" (minimal repair). Partial matching is allowed.
maxXi	Maximization method for the marginal likelihood of ξ: "Newton-Raphson" (default) or "Brent".
se	Standard error estimation: "Information matrix" (default) or "Jacknife" (leave-one-out; can be time consuming).
cancer	Optional character vector with the response achieved after each treatment for the cancer model of González et al. (2005), coded "CR", "PR" or "SD". See the lymphoma data set.
keep.data	Keep the prepared data in the fitted object? It lets anova() refit the model without the original data, at the cost of a larger object. Set to FALSE for very large data sets.

Details

Estimation with frailties is implemented using an expectation-maximization (EM) algorithm. The marginal likelihood for ξ is maximized either by Newton-Raphson or by Brent's method, re-parameterizing $\xi^* = \log(\xi)$ to avoid negative estimates. Iteration terminates when successive values of $\xi/(1+\xi)$ differ by no more than tol. Estimation under the frailty model can be slow for large data sets.

Value

An object of class gcmrec with components including coef (estimated α and β), var (covariance matrix), loglik, Xi and frailties (frailty model only), Lambda and Survival (baseline estimates at the distinct effective ages diseff). Methods are provided for print(), summary(), plot(), lines(), coef(), vcov() and logLik().

References

Peña, E. and Hollander, M. (2004). Models for recurrent events in reliability and survival analysis. In *Mathematical Reliability: An Expository Perspective* (Chapter 6, pp. 105–123). Kluwer Academic Publishers.

González, J.R., Peña, E. and Slate, E. (2005). Modelling intervention effects after cancer relapses. *Statistics in Medicine*, 24(24), 3959–3975.

Gail, M., Santner, T. and Brown, C. (1980). An analysis of comparative carcinogenesis experiments based on multiple times to tumor. *Biometrics*, 36, 255–266.

See Also

Survr(), addCenTime(), graph.caltimes()

Examples

```
data(readmission)

# Perfect repair model with rho(k; alpha) = alpha^k
mod <- gcmrec(Survr(id, time, event) ~ as.factor(dukes),
              data = readmission, s = 3000)
mod

# Minimal repair model
mod.min <- gcmrec(Survr(id, time, event) ~ as.factor(dukes),
                  data = readmission, s = 3000, typeEffage = "min")

# Legacy list format converted internally
data(hydraulic)
gcmrec(Survr(id, time, event) ~ covar.1 + covar.2,
       data = hydraulic, s = 4753)


# Model with frailties (EM algorithm)
mod.fra <- gcmrec(Survr(id, time, event) ~ as.factor(dukes),
                  data = readmission, s = 3000, Frailty = TRUE)

# Cancer model with effective age given by treatment response
data(lymphoma)
mod.can <- gcmrec(Survr(id, time, event) ~ as.factor(distrib),
                  data = lymphoma, s = 1000, cancer = lymphoma$effage)
```

---

gcmrecThreads *Number of threads used by gcmrec*

Description

Gets or sets the maximum number of threads used by the parallel parts of the package, currently the jackknife standard errors, which refit the model once per subject.

Usage

```
gcmrecThreads(n)
```

Arguments

n	Maximum number of threads, or NULL to restore the default (let OpenMP decide). If missing, the current setting is returned without changing it.

Details

By default the package lets OpenMP decide, which means it honours the `OMP_NUM_THREADS` environment variable. Set this option to cap the number of threads regardless of that variable, for instance when running inside a shared machine or a job scheduler.

If the package was built without OpenMP support (which is the case with the default compiler on macOS), the computation runs sequentially and this setting has no effect.

Value

The current setting, invisibly when it is being changed. `NULL` means "let OpenMP decide".

Examples

```
gcmrecThreads()        # current setting

old <- gcmrecThreads(2)
gcmrecThreads(old)     # restore
```

GeneratedData *Simulated data set generated under the minimal repair model*

Description

Recurrence times simulated under the minimal repair model with probability of perfect repair equal to 0.6, including the generated effective age information. Useful as an example of the `effageData` argument of `gcmrec()`.

Usage

```
GeneratedData
```

Format

A list in the legacy gcmrec format (elements `n`, `paramlist` and `subject`), where each subject includes the effective age components `intercepts`, `slopes`, `lastperrep`, `perrepind`, `effagebegin` and `effage`.

Examples

```
data(GeneratedData)
temp <- List.to.Dataframe(GeneratedData)
head(temp)
```

graph.caltimes *Event chart of recurrent events in calendar time*

Description

Draws one row per subject with a segment covering its follow-up, a point at each recurrence and a marker at the end of follow-up. It is the first look you should take at recurrent event data: it shows at a glance how many subjects recur, how often, and whether the events cluster early or late.

Usage

```
graph.caltimes(
  data,
  var = NULL,
  effageData = NULL,
  sortevents = c("followup", "events", "none"),
  width = 2,
  lines = TRUE,
  max.subjects = 300,
  ...
)
```

Arguments

data	A data frame with id, time and event variables (and possibly covariates), or a legacy gcmrec list (see as_gcmrec_data()).
var	Optional variable used to colour the subjects, of the same length as nrow(data). Categorical variables colour by group; continuous ones use a gradient.
effageData	Optional effective age information; recurrences with perrepind == 0 (imperfect repair) are drawn as hollow triangles.
sortevents	How to order the subjects along the vertical axis: "followup" (default) by length of follow-up, "events" by number of recurrences, or "none" to keep the order of the data. In earlier versions this argument was accepted but ignored.
width	Size of the plotting symbols.
lines	Draw the follow-up segment of each subject?
max.subjects	Maximum number of subjects to draw. With more, a random sample of this size is taken (with a message), since the chart becomes unreadable beyond a few hundred rows. Inf draws all.
...	Ignored; kept for compatibility with earlier versions.

Value

A ggplot object.

See Also

`mcf()` for the cumulative view of the same data.

Examples

```
data(readmission)
graph.caltimes(readmission[readmission$id %in% 1:40, ])

# Coloured by a covariate, ordered by number of events
graph.caltimes(readmission[readmission$id %in% 1:40, ],
               var = readmission$sex[readmission$id %in% 1:40],
               sortevents = "events")
```

---

hydraulic *Hydraulic load-haul-dump (LHD) subsystems*

Description

Hydraulic load-haul-dump (LHD) subsystems used in moving ore and rock in underground mines in Sweden. The data set provides the calendar times (in hours), excluding repair or down times, of the successive failures of 6 such systems during the two-year development phase.

Usage

```
hydraulic
```

Format

A list in the legacy gcmrec format, with elements `n` (number of systems) and `subject` (a list with `k`, `tau`, `caltimes`, `gaptimes`, effective age components and covariates for each system). Use `List.to.Dataframe()` or pass it directly to `gcmrec()`, which converts it via `as_gcmrec_data()`.

Source

Kumar, D. and Klefsjö, B. (1992). Reliability analysis of hydraulic systems of LHD machines using the power law process model. *Reliability Engineering and System Safety*, 35, 217–224.

Examples

```
data(hydraulic)
head(List.to.Dataframe(hydraulic))
```

is.Survr *Test for a survival recurrent object*

Description

Test for a survival recurrent object

Usage

```
is.Survr(x)
```

Arguments

x	An R object.

Value

TRUE if x inherits from class Survr.

See Also

Survr()

Examples

```
is.Survr(Survr(id = c(1, 1), time = c(5, 3), event = c(1, 0)))
```

lines.gcmrec *Add gcmrec baseline curves to a base graphics plot*

Description

[Deprecated]

Adds the baseline curve of a fitted model to an open **base graphics** plot. Since version 2.0 plot.gcmrec() returns a ggplot object, so lines() can no longer add a curve to it. Use plotBaseline() with a list of models instead, which draws all of them on the same axes with a legend.

Usage

```
## S3 method for class 'gcmrec'
lines(x, type.plot = "survival", ...)
```

Arguments

x	An object of class gcmrec.
type.plot	Curve to add: "survival" (default) or "hazard".
...	Further graphical parameters passed to graphics::lines().

Value

Invisibly, x.

See Also

plotBaseline(), the recommended replacement.

Examples

```
data(readmission)
mod.per <- gcmrec(Survr(id, time, event) ~ as.factor(dukes),
                  data = readmission, s = 3000, typeEffage = "perfect")
mod.min <- gcmrec(Survr(id, time, event) ~ as.factor(dukes),
                  data = readmission, s = 3000, typeEffage = "minimal")

# Recommended: both curves in one call
plotBaseline(list(perfect = mod.per, minimal = mod.min))
```

List.to.Dataframe *Convert the legacy gcmrec list format to a data frame*

Description

Converts a data set in the legacy list format (elements n and subject, where each subject holds k, tau, gaptimes and a covariate matrix; see the hydraulic and GeneratedData data sets) into a data frame with columns id, time, event and covar.*.

Usage

```
List.to.Dataframe(data)
```

Arguments

data	A list with elements n (number of subjects) and subject (a list with one entry per subject).

Value

A data frame with one row per recurrence plus the censoring time of each subject.

See Also

as_gcmrec_data(), gcmrec()

Examples

```
data(hydraulic)
head(List.to.Dataframe(hydraulic))
```

logLik.gcmrec *Extract the log-likelihood from a gcmrec model*

Description

Extract the log-likelihood from a gcmrec model

Usage

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

Arguments

object	An object of class gcmrec.
...	Ignored; kept for compatibility with the generic.

Value

An object of class logLik. For frailty models it is the partial log-likelihood conditional on the estimated frailties.

See Also

gcmrec()

lymphoma *Indolent non-Hodgkin's lymphomas*

Description

Cancer relapse times after first treatment in patients diagnosed with low grade lymphoma.

Usage

```
lymphoma
```

Format

A data frame with 112 rows and 9 columns:

id identifier of each subject, repeated for each recurrence

time inter-occurrence or censoring time

event censoring status: 1 for each relapse, 0 for the last (censored) record of each subject

enum which relapse

delay delay between first symptom and date of first treatment

age age at diagnosis

sex gender: 1: Males, 2: Females

distrib lesions involved at diagnosis: 0: Single, 1: Localized, 2: More than one nodal site, 3: Generalized

effage response achieved after the treatment at each relapse: `"CR"` complete remission, `"PR"` partial remission, `"SD"` stable disease or null response. Used as the `cancer` argument of `gcmrec()`

Source

González, J.R., Peña, E. and Slate, E. (2005). Modelling intervention effects after cancer relapses. *Statistics in Medicine*, 24(24), 3959–3975.

Servitje, O., Gallardo, F., Estrach, T. et al. (2002). Primary cutaneous marginal zone B-cell lymphoma. *British Journal of Dermatology*, 147, 1147–1158.

Examples

```
data(lymphoma)
head(lymphoma)
```

---

mcf *Mean cumulative function of recurrent events*

Description

Estimates the mean cumulative function (MCF), that is, the expected number of events accumulated by a subject up to each time. It is the Nelson-Aalen estimator adapted to recurrent events, and it answers the question *"how many events does an average subject accumulate?"* without assuming any model.

Usage

```
mcf(data, group = NULL, level = 0.95)

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

## S3 method for class 'mcf'
plot(x, conf.int = TRUE, ...)
```

Arguments

`data`	A data frame with `id`, `time` and `event` variables, or a legacy gcmrec list (see `as_gcmrec_data()`).
`group`	Optional grouping variable of length `nrow(data)`; one MCF is estimated per group.

level	Confidence level of the pointwise interval.
x	An object of class mcf.
...	Ignored; kept for compatibility with the generic.
conf.int	Draw the confidence band?

Details

The estimator adds, at each event time t, the number of events observed at t divided by the number of subjects still under follow-up, and the pointwise variance is accumulated in the same way. Use it before modelling: comparing the MCF of two groups shows whether their event burden really differs, and its shape suggests whether the risk grows or falls with time.

Value

An object of class mcf: a data frame with columns time, estimate, lower, upper, n.risk, n.event and, if group was given, group. It has plot() and print() methods.

For print, invisibly x.

See Also

graph.caltimes() for the individual view of the same data.

Examples

```
data(readmission)

# Overall event burden
m <- mcf(readmission)
m
plot(m)

# Does it differ by tumour stage?
m.dukes <- mcf(readmission, group = readmission$dukes)
plot(m.dukes)
```

---

plot.gcmrec *Plot the baseline functions of a gcmrec model*

Description

Plots the estimated baseline survivor function or the baseline cumulative hazard of a fitted gcmrec() model as a step function over the effective age, with a pointwise confidence band.

Usage

```
## S3 method for class 'gcmrec'
plot(x, type.plot = "survival", conf.int = TRUE, level = 0.95, ...)
```

Arguments

x	An object of class gcmrec.
type.plot	Curve to plot: "survival" (default) or "hazard". Partial matching is allowed.
conf.int	Draw the confidence band?
level	Confidence level of the band.
...	Ignored; kept for compatibility with the generic.

Details

The band is built from the Aalen variance of the cumulative hazard, transformed to the log scale for the hazard and to the log-log scale for the survivor function, so that the limits stay positive and inside $[0, 1]$ respectively (the default transformation of survival::survfit()).

Value

A ggplot object, which can be further styled with the usual ggplot2 syntax.

See Also

plotBaseline() to compare several models, plotPredict() for covariate-specific curves.

Examples

```
data(readmission)
mod <- gcmrec(Survr(id, time, event) ~ as.factor(dukes),
              data = readmission, s = 3000)
plot(mod)
plot(mod, type.plot = "hazard", level = 0.99)

# It is a ggplot, so it can be restyled
plot(mod) + ggplot2::labs(title = "Colorectal cancer readmissions")
```

---

plotBaseline *Compare the baseline functions of several models*

Description

Draws the baseline survivor or cumulative hazard functions of two or more fitted models on the same axes, which is the natural way to see what an assumption (for instance perfect versus minimal repair) changes.

Usage

```
plotBaseline(models, type.plot = "survival", conf.int = FALSE, level = 0.95)
```

Arguments

models	A named list of gcmrec objects. The names are used in the legend.
type.plot	Curve to plot: "survival" (default) or "hazard".
conf.int	Draw confidence bands? Off by default, since overlapping bands are hard to read.
level	Confidence level of the bands.

Value

A ggplot object.

See Also

plot.gcmrec()

Examples

```
data(readmission)
mod.per <- gcmrec(Survr(id, time, event) ~ as.factor(dukes),
                  data = readmission, s = 3000, typeEffage = "perfect")
mod.min <- gcmrec(Survr(id, time, event) ~ as.factor(dukes),
                  data = readmission, s = 3000, typeEffage = "minimal")
plotBaseline(list(perfect = mod.per, minimal = mod.min))
```

---

plotForest *Forest plot of the hazard ratios*

Description

Draws the exponentiated covariate coefficients of a fitted gcmrec() model with their confidence intervals, on a logarithmic axis with a reference line at 1. This is the figure that usually accompanies the table returned by summary.gcmrec().

Usage

```
plotForest(
  x,
  level = 0.95,
  labels = NULL,
  sort = FALSE,
  title = "Hazard ratios",
  subtitle = NULL
)
```

Arguments

x	An object of class gcmrec.
level	Confidence level of the intervals.
labels	Optional named character vector to relabel the covariates: c(sex = "Female vs male"). Names are matched against the coefficient names.
sort	Order the covariates by hazard ratio instead of keeping the order of the model?
title, subtitle	Plot title and subtitle; NULL for none.

Value

A ggplot object.

See Also

summary.gcmrec()

Examples

```
data(readmission)
mod <- gcmrec(Survr(id, time, event) ~ as.factor(dukes) + sex,
              data = readmission, s = 3000)
plotForest(mod)

# Readable labels for a report
plotForest(mod, labels = c("as.factor(dukes)2" = "Dukes C vs A-B",
                           "as.factor(dukes)3" = "Dukes D vs A-B",
                           "sex" = "Female vs male"))
```

---

plotPredict *Plot predicted curves for covariate profiles*

Description

Draws the survivor or cumulative hazard curve of each covariate profile, which is the way to show what the model implies for concrete subjects (for instance, the three Dukes stages side by side).

Usage

```
plotPredict(object, newdata, type = c("survival", "hazard"), labels = NULL)
```

Arguments

object	An object of class gcmrec.
newdata	Data frame of covariate profiles, one row per profile. Its row names, if any, are used as labels in the legend.
type	"survival" (default) or "hazard".
labels	Optional character vector of labels for the profiles, one per row of newdata.

Value

A ggplot object.

See Also

predict.gcmrec()

Examples

```
data(readmission)
mod <- gcmrec(Survr(id, time, event) ~ as.factor(dukes) + sex,
              data = readmission, s = 3000)

profiles <- data.frame(dukes = c(1, 2, 3), sex = 1)
plotPredict(mod, profiles,
            labels = c("Dukes A-B", "Dukes C", "Dukes D"))
```

---

predict.gcmrec *Predictions from a gcmrec model*

Description

Computes the linear predictor, the relative risk or the whole survivor and cumulative hazard curves for given covariate profiles.

Usage

```
## S3 method for class 'gcmrec'
predict(
  object,
  newdata = NULL,
  type = c("risk", "lp", "survival", "hazard"),
  ...
)
```

Arguments

object	An object of class gcmrec.
newdata	Data frame of covariate profiles, one row per profile, with the covariates used in the model. If NULL, the profiles of the data used to fit are taken.
type	Type of prediction: "risk" (relative risk $\exp(x'\beta)$, the default), "lp" (linear predictor $x'\beta$), "survival" or "hazard" (the whole curve for each profile).
...	Ignored; kept for compatibility with the generic.

Details

The model is multiplicative, so a covariate profile x scales the baseline by $\psi(x) = \exp(x'\beta)$: the survivor function of the profile is $S_0(t)^{\psi(x)}$ and its cumulative hazard is $\psi(x)\Lambda_0(t)$. Note that the curves are functions of the **effective age**, not of calendar time, and that they describe the gap to the next event.

Value

For `"risk"` and `"lp"`, a numeric vector with one value per profile. For `"survival"` and `"hazard"`, a data frame with columns `profile`, `time` and `estimate`, ready to be plotted (see `plotPredict()`).

See Also

`plotPredict()`, `plot.gcmrec()`

Examples

```
data(readmission)
mod <- gcmrec(Survr(id, time, event) ~ as.factor(dukes) + sex,
              data = readmission, s = 3000)

# Relative risk of two patient profiles
profiles <- data.frame(dukes = c(1, 3), sex = c(1, 1))
predict(mod, profiles, type = "risk")

# Whole survivor curve of each profile
head(predict(mod, profiles, type = "survival"))
```

---

print.gcmrec *Print a gcmrec model fit*

Description

Prints the coefficient table (with standard errors, z statistics and p-values), the α parameter of the ρ function, the frailty parameter when present, and fit summaries.

Usage

```
## S3 method for class 'gcmrec'
print(x, digits = max(options()$digits - 4, 3), ...)
```

Arguments

`x`	An object of class `gcmrec`.
`digits`	Number of significant digits to print.
`...`	Ignored; kept for compatibility with the generic.

Value

Invisibly, x.

See Also

gcmrec(), summary.gcmrec()

---

print.summary.gcmrec *Print a gcmrec summary*

Description

Print a gcmrec summary

Usage

```
## S3 method for class 'summary.gcmrec'
print(x, digits = 2, lab = "hr", ...)
```

Arguments

x	An object of class summary.gcmrec.
digits	Number of decimal digits in the table.
lab	Label of the hazard ratio column.
...	Ignored; kept for compatibility with the generic.

Value

Invisibly, x.

See Also

summary.gcmrec()

`readmission` *Rehospitalization of colorectal cancer patients*

Description

Rehospitalization times after surgery in patients diagnosed with colorectal cancer.

Usage

```
readmission
```

Format

A data frame with 861 rows and 10 columns:

id identifier of each subject, repeated for each recurrence

enum which readmission

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

t.stop recurrence or censoring time

time inter-occurrence or censoring time

event censoring status: 1 for each readmission, 0 for the last (censored) record of each subject

chemo did the patient receive chemotherapy? 1: No; 2: Yes

sex gender: 1: Males, 2: Females

dukes Dukes' tumoral stage: 1: A-B; 2: C; 3: D

charlson Charlson comorbidity index (time-dependent covariate): 0: index 0; 1: index 1-2; 3: index >= 3

Source

González, J.R., Fernández, E., Moreno, V. et al. (2005). Gender differences in hospital readmission among colorectal cancer patients. *Journal of Epidemiology and Community Health*, 59(6), 506–511.

Examples

```
data(readmission)
head(readmission)
```

`summary.gcmrec` *Summary of a gcmrec model fit*

Description

Computes hazard ratios (exponentiated covariate coefficients) with confidence intervals for a fitted `gcmrec()` model. Unlike older versions of the package, the summary is returned as an object (of class `summary.gcmrec`) with its own print method, following the standard S3 idiom.

Usage

```
## S3 method for class 'gcmrec'
summary(object, level = 0.95, ...)
```

Arguments

`object`	An object of class `gcmrec`.
`level`	Confidence level for the intervals (default 0.95).
`...`	Ignored; kept for compatibility with the generic.

Value

An object of class `summary.gcmrec`, a list with components:

`call`	the call that produced the fit
`conf.int`	matrix with the hazard ratio and its confidence limits for each covariate
`level`	the confidence level used
`loglik`, `n`, `nk`	fit summaries copied from the model object

See Also

`gcmrec()`, `print.gcmrec()`

Examples

```
data(readmission)
mod <- gcmrec(Survr(id, time, event) ~ as.factor(dukes),
              data = readmission, s = 3000)
summary(mod)
```

Survr *Create a survival recurrent object*

Description

Creates the response object used by `gcmrec()` for recurrent event data. Every subject must have exactly one censored time (its last record, with `event = 0`); use `addCenTime()` when the end of follow-up coincides with the last occurrence.

Usage

```
Survr(id, time, event)
```

Arguments

`id`	Subject identifier, repeated for each recurrence.
`time`	Inter-occurrence or censoring time.
`event`	Event indicator: 1 for an event, 0 for the censoring time.

Value

An object of class `Survr`: a matrix with columns `id`, `time` and `event`.

See Also

`gcmrec()`, `addCenTime()`

Examples

```
data(readmission)
with(readmission, head(Survr(id, time, event)))
```

theme_gcmrec *Plot theme used by the gcmrec graphics*

Description

A clean `ggplot2` theme shared by all the plotting functions of the package: light horizontal grid, no panel border, muted axis text and room for the legend at the bottom. Exported so that the figures of a report can be restyled consistently.

Usage

```
theme_gcmrec(base_size = 12)
```

Arguments

base_size	Base font size in points.

Value

A ggplot2 theme object, to be added to any plot.

Examples

```
data(readmission)
mod <- gcmrec(Survr(id, time, event) ~ as.factor(dukes),
              data = readmission, s = 3000)
plot(mod) + theme_gcmrec(base_size = 14)
```

---

vcov.gcmrec *Extract the covariance matrix from a gcmrec model*

Description

Extract the covariance matrix from a gcmrec model

Usage

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

Arguments

object	An object of class gcmrec.
...	Ignored; kept for compatibility with the generic.

Value

Covariance matrix of the estimated parameters (inverse of the information matrix, or the jackknife estimate if the model was fitted with se = "Jacknife").

See Also

gcmrec()

Index

∗ **datasets**
GeneratedData, 10
hydraulic, 12
lymphoma, 15
readmission, 24

addCenTime, 3
addCenTime(), *9*, *26*
anova(), *8*
anova.gcmrec, 4
as_gcmrec_data, 5
as_gcmrec_data(), *7*, *11, 12*, *14*, *16*

coef(), *8*
coef.gcmrec, 6

gcmrec, 7
gcmrec(), *3*, *5–7*, *10*, *12*, *14–17*, *19*, *23*, *25–27*
gcmrecThreads, 9
GeneratedData, 10, *14*
graph.caltimes, 11
graph.caltimes(), *5*, *9*, *17*
graphics::lines(), *13*

hydraulic, 12, *14*

is.Survr, 13

lines(), *8*
lines.gcmrec, 13
List.to.Dataframe, 14
List.to.Dataframe(), *6, 7*, *12*
logLik(), *8*
logLik.gcmrec, 15
logLik.gcmrec(), *5*
lymphoma, *8*, 15

mcf, 16
mcf(), *12*

plot(), *8*, *17*
plot.gcmrec, 17
plot.gcmrec(), *13*, *19*, *22*
plot.mcf *(*mcf*)*, 16
plotBaseline, 18
plotBaseline(), *13, 14*, *18*
plotForest, 19
plotPredict, 20
plotPredict(), *18*, *22*
predict.gcmrec, 21
predict.gcmrec(), *21*
print(), *8*, *17*
print.anova.gcmrec *(*anova.gcmrec*)*, 4
print.gcmrec, 22
print.gcmrec(), *25*
print.mcf *(*mcf*)*, 16
print.summary.gcmrec, 23

readmission, 24

summary(), *8*
summary.gcmrec, 25
summary.gcmrec(), *19, 20*, *23*
survival::survfit(), *18*
Survr, 26
Survr(), *3*, *7*, *9*, *13*

theme_gcmrec, 26

vcov(), *8*
vcov.gcmrec, 27