Package: SmoothHazard (via r-universe)

September 8, 2024

Title Estimation of Smooth Hazard Models for Interval-Censored Data

Version 2024.04.10

Description Estimation of two-state (survival) models and irreversible illness- death models with possibly interval-censored, left-truncated and right-censored data. Proportional intensities regression models can be specified to allow for covariates effects separately for each transition. We use either a parametric approach with Weibull baseline intensities or a semi-parametric approach with M-splines approximation of baseline intensities in order to obtain smooth estimates of the hazard functions. Parameter estimates are obtained by maximum likelihood in the parametric approach and by penalized maximum likelihood in the semi-parametric approach.

Encoding UTF-8

Copyright See inst/COPYRIGHTS

Depends R (>= 1.9.1), prodlim (>= 1.4.9)

Imports lava (>= 1.4.1), mvtnorm (>= 1.0-3)

Maintainer Thomas Alexander Gerds `<tag@biostat.ku.dk>`

License GPL (>= 2)

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

RoxygenNote 7.3.1

NeedsCompilation yes

Author Celia Touraine [aut], Thomas Alexander Gerds [aut, cre], Pierre Joly [aut] (Author and maintainer of the Fortran code), Cecile Proust-Lima [aut] (Author of the Fortran code), Helene Jacqmin-Gadda [aut] (Author of the Fortran code), Amadou Diakite [aut] (Author of the Fortran code), W.D. Cody [aut] (Author of the Fortran code), A.H. Morris [aut] (Author of the Fortran code), B.W. Brown [aut] (Author of the Fortran code), Robin Genuer [ctb]

Date/Publication 2024-04-10 20:00:08 UTC

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

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

RemoteRef HEAD

RemoteSha 7e55a8fa4abad21334d9227d6776f9c65e4e0dac

Contents

idm 2
idmModel 7
intensity 10
Paq1000 12
plot.idm 13
plot.shr 14
predict.idm 16
print.idm 18
print.shr 19
shr 20
sim.idmModel 23
sim.survIC 24
simulateIDM 25
summary.idm 26
summary.shr 27
survIC 28
testdata 29

Index **30**

`idm` *Fit an illness-death model*

Description

Fit an illness-death model using either a semi-parametric approach (penalized likelihood with an approximation of the transition intensity functions by linear combination of M-splines) or a parametric approach (specifying Weibull distributions on the transition intensities). Left-truncated, right-censored, and interval-censored data are allowed. State 0 corresponds to the initial state, state 1 to the transient one, state 2 to the absorbant one. The allowed transitions are: 0 –> 1, 0 –> 2 and 1 –> 2.

Usage

```
idm(
  formula01,
  formula02,
  formula12,
  data,
```

```
  maxiter = 200,
  eps = c(5, 5, 3),
  n.knots = c(7, 7, 7),
  knots = "equidistant",
  CV = FALSE,
  kappa = c(1000000, 500000, 20000),
  method = "Weib",
  conf.int = 0.95,
  print.iter = FALSE,
  subset = NULL,
  na.action = na.fail
)
```

Arguments

`formula01`	A formula specifying a regression model for the `0 --> 1` transition from the initial state to the transient state of the illness-death model. The right hand side of the formula specifies the covariate terms, and the left hand side must be an event history object as returned by the function `Hist`.
`formula02`	A formula specifying a regression model for the `0 --> 2` transition from the initial state to the absorbing state. The left hand side must be equal to the left hand side of `formula01`. If missing it is set to `formula01`.
`formula12`	A formula specifying a regression model for the `1 --> 2` transition from the transient state to the absorbing state. operator is not required. If missing it is set to `formula01`.
`data`	A data frame in which to interpret the variables of `formula01`, `formula02` and `formula12`.
`maxiter`	Maximum number of iterations. The default is 200.
`eps`	A vector of 3 integers >0 used to define the power of three convergence criteria: 1. for the regression parameters, 2. for the likelihood, 3. for the second derivatives. The default is `c(5,5,3)` which is translated into convergence if the respective values change less then 10^{-5} (for regression parameters and likelihood) and 10^{-3} for the second derivatives between two iterations.
`n.knots`	For `method="Splines"` only, a vector of length 3 specifing the number of knots, one for each transition, for the M-splines estimate of the baseline intensities in the order `0 --> 1`, `0 --> 2`, `1 --> 2`. The default is c(7,7,7). When `knots` are specified as a list this argument is ignored. The algorithm needs least 5 knots and at most 20 knots.
`knots`	Argument only active for the penalized likelihood approach `method="Splines"`. There are three ways to control the placement of the knots between the smallest and the largest of all time points: `knots="equidistant"` Knots are placed with same distance on the time scale. `knots="quantiles"` Knots are placed such that the number of observations is roughly the same between knots. **knots=list()** List of 1 or 2 or three vectors. The list elements are the actual placements (timepoints) of the knots for the M-spline. The list may contain

one vector of placements for each transition in the order `0` `-->` `1`, `0` `-->` `2`, `1` `-->` `2`. If only vector is specified the knots are used for all transitions. If only 2 vectors are specified, the knots for the `0` `-->` `1` transition are also used for the `1` `-->` `2` transition.

The algorithm needs at least 5 knots and allows no more than 20 knots.

`CV`	Binary variable equals to 1 when search (by approximated cross validation) of the smoothing parameters `kappa` and 0 otherwise. Argument for the penalized likelihood approach. The default is 0.
`kappa`	Argument only active for the penalized likelihood approach `method="Splines"`. A vector with 3 positive values (smoothing parameters), one for each transition, in the order 0 –> 1, 0 –> 2 and 1 –> 2.. If CV=1 these are used as starting values for a cross validation search to optimize kappa.
`method`	type of estimation method: "Splines" for a penalized likelihood approach with approximation of the transition intensities by M-splines, "Weib" for a parametric approach with a Weibull distribution on the transition intensities. Default is "Weib".
`conf.int`	Level of confidence pointwise confidence intervals of the transition intensities, i.e., a value between 0 and 1, the default is `0.95`. The default is also used when `conf.int=TRUE`. To avoid computation of confidence intervals, set `conf.int` to FALSE or NULL.
`print.iter`	boolean parameter. Equals to `TRUE` to print the likelihood during the iteration process, `FALSE` otherwise. Default is `FALSE`. This option is not running on Windows.
`subset`	expression indicating the subset of the rows of data to be used in the fit. All observations are included by default.
`na.action`	how NAs are treated. The default is first, any na.action attribute of data, second a na.action setting of options, and third 'na.fail' if that is unset. The 'factory-fresh' default is na.omit. Another possible value is NULL.

Details

The estimated parameters are obtained using the robust Marquardt algorithm (Marquardt, 1963) which is a combination between a Newton-Raphson algorithm and a steepest descent algorithm.

Value

`call`	the call that produced the result.
`coef`	regression parameters.
`loglik`	vector containing the log-likelihood without and with covariate.
`cv`	vector containing the convergence criteria.
`niter`	number of iterations.
`converged`	integer equal to 1 when the model converged, 2, 3 or 4 otherwise.
`modelPar`	Weibull parameters.
`N`	number of subjects.

`events1`	number of events 0 –> 1.
`events2`	number of events 0 –> 2 or 0 –> 1 –> 2.
`NC`	vector containing the number of covariates on transitions 0 –> 1, 0 –> 2, 1 –> 2.
`responseTrans`	model response for the 0 –> 1 transition. `Hist` or `Surv` object.
`responseAbs`	model response for the 0 –> 2 transition. `Hist` or `Surv` object.
`time`	times for which transition intensities have been evaluated for plotting. Vector in the Weibull approach. Matrix in the penalized likelihhod approach for which the colums corresponds to the transitions 0 –> 1, 1 –> 2, 0 –> 2.
`intensity01`	matched values of the intensities for transition 0 –> 1.
`lowerIntensity01`	lower confidence intervals for the values of the intensities for transition 0 –> 1.
`upperIntensity01`	upper confidence intervals for the values of the intensities for transition 0 –> 1.
`intensity02`	matched values of the intensities for transition 0 –> 2.
`lowerIntensity02`	lower confidence intervals for the values of the intensities for transition 0 –> 2.
`upperIntensity02`	upper confidence intervals for the values of the intensities for transition 0 –> 2.
`intensity12`	matched values of the intensities for transition 1 –> 2.
`lowerIntensity12`	lower confidence intervals for the values of the intensities for transition 1 –> 2.
`upperIntensity12`	upper confidence intervals for the values of the intensities for transition 1 –> 2.
`RR`	vector of relative risks.
`V`	variance-covariance matrix derived from the Hessian of the log-likelihood if using method="Weib" or, from the Hessian of the penalized log-likelihood if using method="Splines".
`se`	standart errors of the regression parameters.
`Xnames01`	names of covariates on 0 –> 1.
`Xnames02`	names of covariates on 0 –> 2.
`Xnames12`	names of covariates on 1 –> 2.
`knots01`	knots to approximate by M-splines the intensity of the 0 –> 1 transition.
`knots02`	knots to approximate by M-splines the intensity of the 0 –> 2 transition.
`knots12`	knots to approximate by M-splines the intensity of the 1 –> 2 transition.
`nknots01`	number of knots on transition 0 –> 1.
`nknots02`	number of knots on transition 0 –> 2.
`nknots12`	number of knots on transition 1 –> 2.
`theta01`	square root of splines coefficients for transition 0 –> 1.
`theta02`	square root of splines coefficients for transition 0 –> 2.
`theta12`	square root of splines coefficients for transition 1 –> 2.

`CV`	a binary variable equals to 1 when search of the smoothing parameters kappa by approximated cross-validation, 1 otherwise. The default is 0.
`kappa`	vector containing the smoothing parameters for transition 0 –> 1, 0 –> 2, 1 –> 2 used to estimate the model by the penalized likelihood approach.
`CVcrit`	cross validation criteria.
`DoF`	degrees of freedom of the model.
`na.action`	observations deleted if missing values.

Author(s)

R: Celia Touraine <Celia.Touraine@isped.u-bordeaux2.fr> Fortran: Pierre Joly <Pierre.Joly@isped.u-bordeaux2.fr>

References

D. Marquardt (1963). An algorithm for least-squares estimation of nonlinear parameters. *SIAM Journal of Applied Mathematics*, 431-441.

See Also

`print.idm` `summary.idm` `predict.idm`

Examples

```
library(lava)
library(prodlim)
set.seed(17)
d <- simulateIDM(100)
# right censored data
fitRC <- idm(formula01=Hist(time=observed.illtime,event=seen.ill)~X1+X2,
             formula02=Hist(time=observed.lifetime,event=seen.exit)~X1+X2,
             formula12=Hist(time=observed.lifetime,event=seen.exit)~X1+X2,data=d,
             conf.int=FALSE)
fitRC


set.seed(17)
d <- simulateIDM(300)
fitRC.splines <- idm(formula01=Hist(time=observed.illtime,event=seen.ill)~X1+X2,
             formula02=Hist(time=observed.lifetime,event=seen.exit)~X1+X2,
             formula12=Hist(time=observed.lifetime,event=seen.exit)~1,data=d,
             conf.int=FALSE,method="splines")

# interval censored data
fitIC <- idm(formula01=Hist(time=list(L,R),event=seen.ill)~X1+X2,
             formula02=Hist(time=observed.lifetime,event=seen.exit)~X1+X2,
             formula12=Hist(time=observed.lifetime,event=seen.exit)~X1+X2,data=d,
             conf.int=FALSE)
fitIC
```

```
data(Paq1000)

# Illness-death model with certif on the 3 transitions
# Weibull parametrization and likelihood maximization

fit.weib <- idm(formula02=Hist(time=t,event=death,entry=e)~certif,
                formula01=Hist(time=list(l,r),event=dementia)~certif,
                data=Paq1000)

# Illness-death model with certif on transitions 01 and 02
# Splines parametrization and penalized likelihood maximization
fit.splines <-  idm(formula02=Hist(time=t,event=death,entry=e)~certif,
                    formula01=Hist(time=list(l,r),event=dementia)~certif,
                    formula12=~1,
                    method="Splines",
                    data=Paq1000)
fit.weib
summary(fit.splines)
```

`idmModel` *Generate illness-death model objects*

Description

Function to generate an illness-death model for simulation.

Usage

```
idmModel(
  scale.illtime = 1/100,
  shape.illtime = 1,
  scale.lifetime = 1/100,
  shape.lifetime = 1,
  scale.waittime = 1/100,
  shape.waittime = 1,
  scale.censtime = 1/100,
  shape.censtime = 1,
  n.inspections = 5,
  schedule = 10,
  punctuality = 5
)
```

Arguments

`scale.illtime`	Weilbull scale for latent illness time
`shape.illtime`	Weilbull shape for latent illness time
`scale.lifetime`	Weilbull scale for latent life time
`shape.lifetime`	Weilbull shape for latent life time
`scale.waittime`	Weilbull scale for latent life time
`shape.waittime`	Weilbull shape for latent life time
`scale.censtime`	Weilbull scale for censoring time
`shape.censtime`	Weilbull shape for censoring time
`n.inspections`	Number of inspection times
`schedule`	Mean of the waiting time between adjacent inspections.
`punctuality`	Standard deviation of waiting time between inspections.

Details

Based on the functionality of the lava PACKAGE the function generates a latent variable model (latent illtime, waittime and lifetime) and censoring mechanism (censtime, inspection1,inspection2,...,inspectionK).

The function `sim.idmModel` then simulates right censored lifetimes and interval censored illness times.

Value

A latent variable model object `lvm`

Author(s)

Thomas Alexander Gerds

Examples

```
library(lava)
library(prodlim)
# generate illness-death model based on exponentially
# distributed times
m <- idmModel(scale.illtime=1/70,
              shape.illtime=1.8,
              scale.lifetime=1/50,
              shape.lifetime=0.7,
              scale.waittime=1/30,
              shape.waittime=0.7)
round(sim(m,6),1)

# Estimate the parameters of the Weibull models
# based on the uncensored exact event times
# and the uncensored illstatus.
set.seed(18)
d <- sim(m,100,latent=FALSE)
```

```
d$uncensored.status <- 1
f <- idm(formula01=Hist(time=illtime,event=illstatus)~1,
         formula02=Hist(time=lifetime,event=uncensored.status)~1,
         data=d,
         conf.int=FALSE)
print(f)

# Change the rate of the 0->2 and 0->1 transitions
# also the rate of the 1->2 transition
# and also lower the censoring rate
m <- idmModel(scale.lifetime=1/2000,
              scale.waittime=1/30,
              scale.illtime=1/1000,
              scale.censtime=1/1000)
set.seed(18)
d <- sim(m,50,latent=TRUE)
d$uncensored.status <- 1

f <- idm(formula01=Hist(time=observed.illtime,event=illstatus)~1,
         formula02=Hist(time=observed.lifetime,event=uncensored.status)~1,
         data=d,
         conf.int=FALSE)
print(f)

# Estimate based on the right censored observations
fc <- idm(formula01=Hist(time=illtime,event=seen.ill)~1,
          formula02=Hist(time=observed.lifetime,event=seen.exit)~1,
          data=d,
          conf.int=FALSE)
print(fc)

# Estimate based on interval censored and right censored observations
fi <- idm(formula01=Hist(time=list(L,R),event=seen.ill)~1,
          formula02=Hist(time=observed.lifetime,event=seen.exit)~1,
          data=d,
          conf.int=FALSE)
print(fi)

# Estimation of covariate effects:
# X1, X2, X3
m <- idmModel(shape.waittime=2,
              scale.lifetime=1/2000,
              scale.waittime=1/300,
              scale.illtime=1/10000,
              scale.censtime=1/10000)
distribution(m,"X1") <- binomial.lvm(p=0.3)
distribution(m,"X2") <- normal.lvm(mean=120,sd=20)
distribution(m,"X3") <- normal.lvm(mean=50,sd=20)
regression(m,to="latent.illtime",from="X1") <- 1.7
regression(m,to="latent.illtime",from="X2") <- 0.07
regression(m,to="latent.illtime",from="X3") <- -0.1
regression(m,to="latent.waittime",from="X1") <- 1.8
regression(m,to="latent.lifetime",from="X1") <- 0.7
```

```
set.seed(28)
d <- sim(m,100,latent=TRUE)
head(d)
table(ill=d$seen.ill,death=d$seen.exit)

# Estimation based on uncensored data
d$uncensored.status <- 1
# uncensored data
F1 <- idm(formula01=Hist(time=illtime,event=illstatus)~X1+X2+X3,
          formula02=Hist(time=lifetime,event=uncensored.status)~X1+X2+X3,
          data=d,conf.int=FALSE)
print(F1)

# Estimation based on right censored data
F2 <- idm(formula01=Hist(time=illtime,event=seen.ill)~X1+X2+X3,
          formula02=Hist(time=observed.lifetime,event=seen.exit)~X1+X2+X3,
          data=d,conf.int=FALSE)
print(F2)

# Estimation based on interval censored and right censored data
F3 <- idm(formula01=Hist(time=list(L,R),event=seen.ill)~X1+X2+X3,
          formula02=Hist(time=observed.lifetime,event=seen.exit)~X1+X2+X3,
          data=d,conf.int=FALSE)
print(F3)
cbind(uncensored=F1$coef,right.censored=F2$coef,interval.censored=F3$coef)
```

`intensity` *M-spline estimate of the transition intensity function*

Description

M-spline estimate of the transition intensity function and the cumulative transition intensity function for survival and illness-death models

Usage

```
intensity(times, knots, number.knots, theta, linear.predictor = 0)
```

Arguments

`times`	Time points at which to estimate the intensity function
`knots`	Knots for the M-spline
`number.knots`	Number of knots for the M-splines (and I-splines see details)
`theta`	The coefficients for the linear combination of M-splines (and I-splines see details)
`linear.predictor`	Linear predictor beta*Z. When it is non-zero, transition and cumulative transition are multiplied by `exp(linear.predictor)`. Default is zero.

Details

The estimate of the transition intensity function is a linear combination of M-splines and the estimate of the cumulative transition intensity function is a linear combination of I-splines (the integral of a M-spline is called I-spline). The coefficients `theta` are the same for the M-splines and I-splines.

Important: the theta parameters returned by `idm` and `shr` are in fact the square root of the splines coefficients. See examples.

This function is a R-translation of a corresponding Fortran function called `susp`. `susp` is used internally by `idm` and `shr`.

Value

`times`	The time points at which the following estimates are evaluated.
`intensity`	The transition intensity function evaluated at `times`.
`cumulative.intensity`	The cumulative transition intensity function evaluated at `times`
`survival`	The "survival" function, i.e., exp(-cumulative.intensity)

Author(s)

R: Celia Touraine <Celia.Touraine@isped.u-bordeaux2.fr> and Thomas Alexander Gerds <tag@biostat.ku.dk>
Fortran: Pierre Joly <Pierre.Joly@isped.u-bordeaux2.fr>

See Also

`shr`, `idm`

Examples

```
data(testdata)
fit.su <- shr(Hist(time=list(l, r), id) ~ cov,
              data = testdata,method = "Splines",CV = TRUE)
intensity(times = fit.su$time, knots = fit.su$knots,
           number.knots = fit.su$nknots, theta = fit.su$theta^2)


  data(Paq1000)
  fit.idm <-  idm(formula02 = Hist(time = t, event = death, entry = e) ~ certif,
                  formula01 = Hist(time = list(l,r), event = dementia) ~ certif,
                  formula12 = ~ certif, method = "Splines", data = Paq1000)
  # Probability of survival in state 0 at age 80 for a subject with no cep given
  # that he is in state 0 at 70
  su0 <- (intensity(times = 80, knots = fit.idm$knots01,
                   number.knots = fit.idm$nknots01,
                   theta = fit.idm$theta01^2)$survival
         *intensity(times = 80, knots = fit.idm$knots02,
                   number.knots = fit.idm$nknots02,
                   theta = fit.idm$theta02^2)$survival)/
        (intensity(times = 70, knots = fit.idm$knots01,
                   number.knots = fit.idm$nknots01,
```

```
                 theta = fit.idm$theta01^2)$survival
      *intensity(times = 70, knots = fit.idm$knots02,
                 number.knots = fit.idm$nknots02,
                 theta = fit.idm$theta02^2)$survival)
# Same result as:
predict(fit.idm, s = 70, t = 80, conf.int = FALSE) # see first element
```

`Paq1000` *Paquid data set*

Description

Paquid data set composed of 1000 subjects selected randomly from the Paquid data set of 3675 subjects.

Format

A data frame with 1000 rows and the following 8 columns.

dementia dementia status, 0=non-demented, 1=demented

death death status, 0=alive, 1=dead

e age at entry in the study

l for demented subjects: age at the visit before the diagnostic visit; for non-demented subjects: age at the last visit (censoring age)

r for demented subjects: age at the diagnostic visit; for non-demented subjects: age at the last visit (censoring age)

t for dead subjects: age at death; for alive subject: age at the latest news

certif primary school certificate:`0=without certificate`, `1=with certificate`

gender gender: `0=female`, `1=male`

Examples

```
data(Paq1000)
```

`plot.idm` *Plot method for an illness-death model*

Description

Plot estimated baseline transition intensities from an object of class `idm` optionally with confidence limits.

Usage

```
## S3 method for class 'idm'
plot(
  x,
  conf.int = FALSE,
  citype = "shadow",
  add = FALSE,
  axes = TRUE,
  col,
  lwd,
  lty,
  xlim,
  ylim,
  xlab,
  ylab,
  legend = TRUE,
  transition = c("01", "02", "12"),
  ...
)
```

Arguments

`x`	a `idmWeib` class object (output from calling `idm` with the (default) option `intensities`="Weib".
`conf.int`	If TRUE show confidence limits
`citype`	Type of confidence limits, can be "shadow" or "bars"
`add`	If TRUE add to existing plot
`axes`	If TRUE axes are drawn
`col`	Color of the lines
`lwd`	Width of the lines
`lty`	Type of the lines
`xlim`	Limits for x-axis
`ylim`	Limits for y-axis
`xlab`	Label for x-axis
`ylab`	Label for y-axis

`legend`	If TRUE a legend is drawn, which can be further controlled via `SmartControl`.
`transition`	Choose one of the transition intensities: `c("01","02","12")`.
`...`	Passed to `SmartControl`

Value

Print a plot of the baseline transition intensities of an illness-death model estimated using a Weibull approach.

See Also

`print.idm`,`summary.idm`,`idm`,

`idm`

Examples

```
library(lava)
library(prodlim)
m <- idmModel(scale.lifetime=1/10,scale.illtime=1/8)
distribution(m,"X") <- binomial.lvm()
regression(m,latent.lifetime~X) <- 0.7
set.seed(30)
d <- sim(m,100)
fit.weib <- idm(formula02=Hist(observed.lifetime,event=seen.exit)~1,
formula01=Hist(time=list(L,R),event=seen.ill)~1,data=d,conf.int=FALSE)
plot(fit.weib)


## FIXME: the limits for the 01 transition are a bit wide!?
## with bootstrap confidence limits
fit.weib <- idm(formula02=Hist(observed.lifetime,event=seen.exit)~1,
formula01=Hist(time=list(L,R),event=seen.ill)~1,data=d,conf.int=TRUE)
plot(fit.weib)
```

`plot.shr` *Plot method for a survival model.*

Description

Plot estimated baseline survival function from an object of class `shr`. Pointwise confidence limits are available.

Usage

```
## S3 method for class 'shr'
plot(
  x,
  type = "shr",
  add = FALSE,
  newdata = NULL,
  cause = NULL,
  col,
  lty,
  lwd,
  ylim,
  xlim,
  xlab = "Time",
  ylab,
  legend = TRUE,
  confint = TRUE,
  timeOrigin = 0,
  axes = TRUE,
  percent = TRUE,
  ...
)
```

Arguments

`x`	a `shrWeib` or a `shrSplines` class object (output from calling `shr` function).
`type`	type of function to plot. The default is "shr".
`add`	boolean.
`newdata`	newdata.
`cause`	cause.
`col`	col.
`lty`	lty.
`lwd`	lwd.
`ylim`	ylim.
`xlim`	xlim.
`xlab`	xlab.
`ylab`	ylab.
`legend`	legend.
`confint`	confint.
`timeOrigin`	timeOrigin.
`axes`	axes.
`percent`	percent.
`...`	other graphical parameters.

Value

Print a plot of a suvival model.

Author(s)

R: Celia Touraine <Celia.Touraine@isped.u-bordeaux2.fr> Fortran: Pierre Joly <Pierre.Joly@isped.u-bordeaux2.fr>

See Also

`plot.shr`

Examples

```
# Weibull survival model
library(prodlim)
data(testdata)
fit.su <- shr(Hist(time=list(l,r),id)~cov,data=testdata)

# pointwise confidence limits
plot(fit.su)

# no pointwise confidence limits
plot(fit.su,confint=FALSE)
```

`predict.idm` *Predictions for an illness-death model using either a penalized likelihood approach or a Weibull parametrization.*

Description

Predict transition probabilities and cumulative probabilities from an object of class `idmSplines` with confidence intervals are calculated.

Usage

```
## S3 method for class 'idm'
predict(
  object,
  s,
  t,
  newdata,
  nsim = 200,
  seed = 21,
  conf.int = 0.95,
  lifeExpect = FALSE,
```

```
  maxtime,
  ...
)
```

Arguments

`object`	an `idm` class objects returned by a call to the `idm` function
`s`	time point at which prediction is made.
`t`	time horizon for prediction.
`newdata`	A data frame with covariate values for prediction.
`nsim`	number of simulations for the confidence intervals calculations. The default is 200.
`seed`	Seed passed to `set.seed` for Monte Carlo simulation of confidence intervals.
`conf.int`	Level of confidence, i.e., a value between 0 and 1, the default is `0.95`. The default is also used when `conf.int=TRUE`. To avoid computation of confidence intervals, set `conf.int` to FALSE or NULL.
`lifeExpect`	Logical. If `TRUE` compute life expectancies, i.e., `t=Inf`.
`maxtime`	The upper limit of integration for calculations of life expectancies from Weibull parametrizations.
`...`	other parameters.

Value

a list containing the following predictions with pointwise confidence intervals:

`p00`	the transition probability p_{00}.
`p01`	the transition probability p_{01}.
`p11`	the transition probability p_{11}.
`p12`	the transition probability p_{12}.
`p02_0`	the probability of direct transition from state 0 to state 2.
`p02_1`	the probability of transition from state 0 to state 2 via state 1.
`p02`	transition probability p_{02}. Note that `p02=p_02_0+p02_1`.
`F01`	the lifetime risk of disease. `F01=p01+p02_1`.
`F0.`	the probability of exit from state 0. `F0.=p02_0+p01+p02_1`.

Author(s)

R: Celia Touraine <Celia.Touraine@isped.u-bordeaux2.fr> and Thomas Alexander Gerds <tag@biostat.ku.dk>
Fortran: Pierre Joly <Pierre.Joly@isped.u-bordeaux2.fr>

See Also

`idm`

Examples

```
set.seed(100)
d=simulateIDM(n = 200)
fit <- idm(formula01=Hist(time=list(L,R),event=seen.ill)~X1+X2+X3,
               formula02=Hist(time=observed.lifetime,event=seen.exit)~X1+X2+X3,
               data=d,conf.int=FALSE)
predict(fit,s=0,t=80,conf.int=FALSE,lifeExpect=FALSE)
predict(fit,s=0,t=80,nsim=4,conf.int=TRUE,lifeExpect=FALSE)
predict(fit,s=0,t=80,nsim=4,conf.int=FALSE,lifeExpect=TRUE)

data(Paq1000)
library(prodlim)
fit.paq <- idm(formula02=Hist(time=t,event=death,entry=e)~certif,
formula01=Hist(time=list(l,r),event=dementia)~certif,data=Paq1000)

predict(fit.paq,s=70,t=80,newdata=data.frame(certif=1))
predict(fit.paq,s=70,lifeExpect=TRUE,newdata=data.frame(certif=1))

fit.splines <-  idm(formula02=Hist(time=t,event=death,entry=e)~certif,
  formula01=Hist(time=list(l,r),event=dementia)~certif,
                formula12=~1,
                method="Splines",
  data=Paq1000)

predict(fit.splines,s=70,t=80,newdata=data.frame(certif=1))
predict(fit.splines,s=70,t=80,lifeExpect=TRUE,newdata=data.frame(certif=1),nsim=20)
```

`print.idm` *Print method for* `idm` *objects*

Description

Print a summary of a fitted illness-death model

Usage

```
## S3 method for class 'idm'
print(x, conf.int = 0.95, digits = 4, pvalDigits = 4, eps = 0.0001, ...)
```

Arguments

`x`	Class `idm` object, i.e. the result of a call to the `idm` function with `intensities`="Weib".
`conf.int`	The level of confidence for the hazard ratios. The default is `0.95`.
`digits`	Number of digits to print.

`pvalDigits`	Number of digits to print for p-values.
`eps`	Passed to `format.pval`.
`...`	Not used.

Value

No return value.

Author(s)

Celia Touraine <Celia.Touraine@isped.u-bordeaux2.fr>, Thomas A. Gerds <tag@biostat.ku.dk>

See Also

`summary.idm`, `plot.idm`

Examples

```
data(Paq1000)
library(prodlim)
fit.splines <-  idm(formula02=Hist(time=t,event=death,entry=e)~certif,
  formula01=Hist(time=list(l,r),event=dementia)~certif,
                formula12=~1,
                method="Splines",
  data=Paq1000)
print(fit.splines)
```

`print.shr` *Print method for* `shrSplines` *objects*

Description

Print a summary of a fitted illness-death model using the penalized likelihood approach.

Usage

```
## S3 method for class 'shr'
print(x, conf.int = 0.95, digits = 4, pvalDigits = 4, eps = 0.0001, ...)
```

Arguments

`x`	a `shr` object, i.e., the result of a call to the `shr` function with `hazard`="Splines".
`conf.int`	The level of confidence for the hazard ratios. The default is `0.95`.
`digits`	number of digits to print.
`pvalDigits`	number of digits to print for p-values.
`eps`	convergence criterion used for p-values.
`...`	other unusued arguments.

Value

No return value.

Author(s)

R: Celia Touraine <Celia.Touraine@isped.u-bordeaux2.fr> Fortran: Pierre Joly <Pierre.Joly@isped.u-bordeaux2.fr>

See Also

`summary.shr`, `plot.shr`

Examples

```
# a penalized survival model
library(prodlim)
data(testdata)
fit.su <- shr(Hist(time=list(l,r),id)~cov,data=testdata,method="Splines")
print(fit.su)
```

---

shr *Fit a survival model*

Description

Fit a survival model using either a semi-parametric approach (penalized likelihood with an approximation of the hazard function by linear combination of M-splines) or a parametric approach (specifying a Weibull distribution on the hazard function). Left-truncated, right-censored, and interval-censored data are allowed.

Usage

```
shr(
  formula,
  data,
  eps = c(5, 5, 3),
  n.knots = 7,
  knots = "equidistant",
  CV = FALSE,
  kappa = 10000,
  conf.int = 0.95,
  maxiter = 200,
  method = "Weib",
  print.iter = FALSE,
  na.action = na.omit
)
```

Arguments

`formula`	a formula object with the response on the left hand side and the terms on the right hand side. The response must be a survival object or Hist object as returned by the `'Surv'` or `'Hist'` function.
`data`	a data frame in which to interpret the variables named in the `formula`.
`eps`	a vector of length 3 for the convergence criteria (criterion for parameters, criterion for likelihood, criterion for second derivatives). The default is `c(5,5,3)` and corresponds to criteria equals to `10^{-5}`, `10^{-5}` and `10^{-3}`.
`n.knots`	Argument only active for the penalized likelihood approach `method="splines"`. Number of knots for the splines to use to approximate the hazard function. The default is 7. If `knots` are given as a vector this argument is ignored. The algorithm needs least 5 knots and at most 20 knots.
`knots`	Argument only active for the penalized likelihood approach `method="splines"`. There are three ways to control the placement of the knots between the smallest and the largest of all time points: `knots="equidistant"` Knots are placed with same distance on the time scale. `knots="quantiles"` Knots are placed such that the number of observations is roughly the same between knots. **knots=list()** List of length 3. The list elements are the actual placements (timepoints) of the knots for the M-spline. The algorithm reuqires at least 5 knots and allows no more than 20 knots.
`CV`	binary variable equals to 1 when search (by approximated cross validation) of the smoothing parameter kappa and 0 otherwise. Argument for the penalized likelihood approach. The default is 0.
`kappa`	Argument only active for the penalized likelihood approach `method="splines"`. A positive number (smoothing parameter) If CV=1 the value is used as a starting value for a cross validation search to optimize `kappa`.
`conf.int`	Level of confidence pointwise confidence intervals of the survival and hazard functions, i.e., a value between 0 and 1, the default is `0.95`. The default is also used when `conf.int=TRUE`. To avoid computation of confidence intervals, set `conf.int` to FALSE or NULL.
`maxiter`	maximum number of iterations. The default is 200.
`method`	type of estimation method: "Splines" for a penalized likelihood approach with approximation of the hazard function by M-splines, "Weib" for a parametric approach with a Weibull distribution on the hazard function. Default is `"Weib"`.
`print.iter`	boolean parameter. Equals to `TRUE` to print the likelihood during the iteration process, `FALSE` otherwise. Default is `FALSE`. This option is not running on Windows.
`na.action`	how NAs are treated. The default is first, any na.action attribute of data, second a na.action setting of options, and third 'na.fail' if that is unset. The 'factory-fresh' default is na.omit. Another possible value is NULL.

Details

The estimated parameters are obtained using the robust Marquardt algorithm (Marquardt, 1963) which is a combination between a Newton-Raphson algorithm and a steepest descent algorithm.

Value

call

coef regression parameters.

loglik vector containing the log-likelihood without and with covariate.

modelPar Weibull parameters.

N number of subjects.

NC number of covariates.

nevents number of events.

modelResponse model response: `Hist` or `Surv` object.

converged integer equal to 1 when the model converged, 2, 3 or 4 otherwise.

time times for which survival and hazard functions have been evaluated for plotting.

hazard matched values of the hazard function.

lowerHazard lower confidence limits for hazard function.

upperHazard upper confidence limits for hazard function.

surv matched values of the survival function.

lowerSurv lower confidence limits for survival function.

upperSurv upper confidence limits for survival function.

RR vector of relative risks.

V variance-covariance matrix.

se standard errors.

knots knots of the M-splines estimate of the hazard function.

nknots number of knots.

CV a binary variable equals to 1 when search of the smoothing parameter kappa by approximated cross-validation, 1 otherwise. The default is 0.

niter number of iterations.

cv vector containing the convergence criteria.

na.action observations deleted if missing values.

Author(s)

R: Celia Touraine `<celia.touraine@icm.unicancer.fr>` Fortran: Pierre Joly `<Pierre.Joly@isped.u-bordeaux2.fr>`

References

D. Marquardt (1963). An algorithm for least-squares estimation of nonlinear parameters. *SIAM Journal of Applied Mathematics*, 431-441.

See Also

`shr`, `print.shr`, `summary.shr`, `print.shr`,

Examples

```
# Weibull survival model
library(prodlim)
data(testdata)
fit.su <- shr(Hist(time=list(l,r),id)~cov,data=testdata)
fit.su
summary(fit.su)

shr.spline <- shr(Hist(time=list(l,r),id)~cov,data=testdata,method="splines",n.knots=6)
shr.spline
shr.spline.q <- shr(Hist(time=list(l,r),id)~cov,data=testdata,
                    method="splines",n.knots=6,knots="quantiles")
plot(shr.spline.q)

## manual placement of knots
shr.spline.man <- shr(Hist(time=list(l,r),id)~cov,data=testdata,method="splines",knots=seq(0,7,1))
```

`sim.idmModel` *Simulate illness-death model data*

Description

Function to simulate illness-death model data

Usage

```
## S3 method for class 'idmModel'
sim(
  x,
  n,
  illness.known.at.death = TRUE,
  compliance = 1,
  latent = FALSE,
  keep.inspectiontimes = FALSE,
  ...
)
```

Arguments

`x`	An `idmModel` object as obtained with `idmModel`
`n`	Number of observations
`illness.known.at.death`	Affects the value of variable seen.ill
`compliance`	Probability of missing an inspection time.
`latent`	if TRUE keep the latent event times
`keep.inspectiontimes`	if `TRUE` keep the inspection times.
`...`	Extra arguments given to `sim`

Details

Based on the functionality of the lava PACKAGE

Value

A data set with interval censored observations from an illness-death model

Author(s)

Thomas Alexander Gerds

Examples

```
example(idmModel)
help(idmModel)
```

`sim.survIC` *Simulate interval censored survival data*

Description

Function to simulate interval censored survival data

Usage

```
## S3 method for class 'survIC'
sim(x, n, compliance = 1, latent = TRUE, keep.inspectiontimes = FALSE, ...)
```

Arguments

`x`	An `survIC` object as obtained with `survIC`
`n`	Number of observations
`compliance`	Probability of missing an inspection time.
`latent`	if TRUE keep the latent event times
`keep.inspectiontimes`	if `TRUE` keep the inspection times.
`...`	Extra arguments given to `sim`

Details

Based on the functionality of the lava PACKAGE

Value

A data set with interval censored observations

Author(s)

Thomas Alexander Gerds

Examples

```
library(lava)
example(survIC)
help(survIC)
ol <- survIC()
dat.ol <- sim(ol,10)
```

`simulateIDM` *Sample illness-death model data*

Description

Simulate data from an illness-death model with interval censored event times and covariates

Usage

```
simulateIDM(n = 100)
```

Arguments

`n`	number of observations

Details

Simulate data from an illness-death model with interval censored event times and covariates for the purpose of illustrating the help pages of the SmoothHazard package. See the body of the function for details, i.e., evaluate simulateIDM

Value

Object with class `data.frame` which contains the simulated data.

See Also

idmModel sim.idmModel

Examples

```
# simulateIDM
simulateIDM(100)
```

`summary.idm` *Summary of a fitted illness-death model*

Description

Summarize the event history data of an illness-death regression model and show regression coefficients for transition intensities

Usage

```
## S3 method for class 'idm'
summary(object, conf.int = 0.95, digits = 4, pvalDigits = 4, eps = 0.0001, ...)
```

Arguments

`object`	a `idmSplines` object, i.e., the result of a call to the `idm` function with `intensities=`"Splines".
`conf.int`	The level of confidence for the hazard ratios. The default is `0.95`.
`digits`	number of digits to print.
`pvalDigits`	number of digits to print for p-values.
`eps`	convergence criterion used for p-values.
`...`	other unusued arguments.

Value

No return value.

Author(s)

R: Celia Touraine <Celia.Touraine@isped.u-bordeaux2.fr> Fortran: Pierre Joly <Pierre.Joly@isped.u-bordeaux2.fr>

See Also

`idm`, `print.idm`, `plot.idm`

Examples

```
library(prodlim)
data(Paq1000)
fit.splines <-  idm(formula02=Hist(time=t,event=death,entry=e)~certif,
  formula01=Hist(time=list(l,r),event=dementia)~certif,
                formula12=~1,
                method="Splines",
  data=Paq1000)
summary(fit.splines)
```

`summary.shr` *Summary of a fitted survival model using a penalized likelihood approach*

Description

Print a short summary of a fitted illness-death model using the penalized likelihood approach.

Usage

```
## S3 method for class 'shr'
summary(object, conf.int = 0.95, digits = 4, pvalDigits = 4, eps = 0.0001, ...)
```

Arguments

`object`	a `shr` object, i.e., the result of a call to the `shr` function.
`conf.int`	The level of confidence for the hazard ratios. The default is `0.95`.
`digits`	number of digits to print.
`pvalDigits`	number of digits to print for p-values.
`eps`	convergence criterion used for p-values.
`...`	other unusued arguments.

Value

No return value.

Author(s)

Celia Touraine <Celia.Touraine@isped.u-bordeaux2.fr>

See Also

`shr`, `print.shr`, `plot.shr`

Examples

```
# a penalized survival model
data(testdata)
library(prodlim)
fit.su <- shr(Hist(time=list(l,r),id)~cov,data=testdata,method="Splines")
summary(fit.su)

# Weibull survival model
data(testdata)
fit.su <- shr(Hist(time=list(l,r),id)~cov,data=testdata)
summary(fit.su)
```

`survIC` *Generate survival model objects*

Description

Function to generate a latent variable model for interval censored survival times.

Usage

```
survIC(
  scale.time = 1/100,
  shape.time = 1,
  n.inspections = 5,
  schedule = 10,
  punctuality = 5
)
```

Arguments

`scale.time`	Weilbull scale for latent time
`shape.time`	Weilbull shape for latent time
`n.inspections`	Number of inspection times
`schedule`	Mean of the waiting time between adjacent inspections.
`punctuality`	Standard deviation of waiting time between inspections.

Details

Based on the functionality of the lava PACKAGE the function generates a latent variable model with a latent time and a censoring mechanism (censtime, inspection1,inspection2,...,inspectionK).

The function `sim.survIC` then simulates interval censored times.

Value

A latent variable model object `lvm`

Author(s)

Thomas Alexander Gerds

Examples

```
library(lava)
library(prodlim)
# generate survival model based on exponentially
# distributed times
m <- survIC(scale.time=1/50, shape.time=0.7)
round(sim(m,6),1)
```

```
# Estimate the parameters of the Weibull models
# based on the uncensored exact event times
# and the uncensored illstatus.
set.seed(18)
d <- sim(m,100,latent=FALSE)
d$uncensored.status <- 1
f <- shr(Hist(time=list(L,R),event=uncensored.status)~1,
         data=d,
         conf.int=FALSE)
print(f)
```

`testdata` *Data set for survival models: right-censored and interval-censored data.*

Description

A simulated data frame for survival models composed of right-censored and interval-censored data.

Format

A data frame with 936 observations on the following 4 variables.

l for diseased subjects: left endpoint of censoring interval; for non-diseased subjects: right censoring time

r for diseased subjects: right endpoint of censoring interval; for non-diseased subjects: right censoring time for the disease event

id disease status

cov covariate

Examples

```
data(testdata)
head(testdata)
```

Index

∗ **datasets**
    Paq1000, 12
    testdata, 29
∗ **illness-death**
    idm, 2
∗ **methods**
    plot.idm, 13
    plot.shr, 14
    predict.idm, 16
    print.idm, 18
    print.shr, 19
    shr, 20
    summary.idm, 26
    summary.shr, 27

idm, 2, *11*, *14*, *17*, *18*, *26*
idmModel, 7
intensity, 10

kappa, *6*, *22*

Paq1000, 12
plot.idm, 13, *19*, *26*
plot.shr, 14, *16*, *20*, *27*
predict.idm, *6*, 16
print.idm, *6*, *14*, 18, *26*
print.shr, 19, *22*, *27*

shr, *11*, *15*, *19*, 20, *22*, *27*
sim.idmModel, *8*, 23
sim.survIC, 24, *28*
simulateIDM, 25
SmartControl, *14*
summary.idm, *6*, *14*, *19*, 26
summary.shr, *20*, *22*, 27
survIC, 28
survModelIC (survIC), 28

testdata, 29