

# Package ‘SmoothHazard’

September 19, 2014

**Title** Fitting illness-death model for interval-censored data

**Version** 1.2.3

**Author** Celia Touraine, Pierre Joly, Thomas A. Gerds

**Description** SmoothHazard is a package for fitting illness-death (and survival) model with possibly interval-censored data for transition toward transient state. Left-truncated and right-censored data are also allowed. Proportional intensities regression models allow for covariates on each of the 3 transitions. 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. Estimation of parameters is conducted by likelihood maximization in the parametric approach or penalized likelihood maximization in the semi-parametric approach.

**Depends** R (>= 1.9.1)

**Imports** lava, mvtnorm, prodlim

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

**License** GPL (>= 2)

**NeedsCompilation** yes

**Repository** CRAN

**Date/Publication** 2014-09-19 00:58:57

## R topics documented:

idm	2
idmModel	6
lifexpect	9
Paq1000	10
plot.idm	11
plot.shr	12
predict.idm	14

print.idm . . . . .	15
print.shr . . . . .	16
shr . . . . .	17
sim.idmModel . . . . .	20
simulateIDM . . . . .	21
summary.idm . . . . .	21
summary.shr . . . . .	22
testdata . . . . .	23

<b>Index</b>	<b>25</b>
--------------	-----------

---

idm	<i>Fit an illness-death model</i>
-----	-----------------------------------

---

## 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 absorbing 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 = TRUE,
    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 than $10^{-5}$ (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 specifying 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).
knots	List of length 3 containing the placements (timepoints) of the knots for the M-spline of the three transitions.
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	a vector of length 3. If CV=FALSE, smoothing parameters for the transition 0 -> 1, 0 -> 2 and 1 -> 2. If CV=TRUE, initial values of the smoothing parameters for the cross validation search. Argument for the penalized likelihood approach.
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	Boolean parameter. Equals to TRUE to calculate pointwise confidence intervals for the transition intensities curves, FALSE otherwise. Default is TRUE.
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 likelihood approach for which the columns 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.
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 <a href="#">kappa</a> 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](#)

**Examples**

```
library(lava)
library(prodlm)
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
# 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

## Not run:

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)
```

```

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)

## to print
fit.weib

## to summary
summary(fit.splines)

## End(Not run)

```

---

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	Weibull scale for latent illness time
shape.illtime	Weibull shape for latent illness time
scale.lifetime	Weibull scale for latent life time
shape.lifetime	Weibull shape for latent life time
scale.waittime	Weibull scale for latent life time
shape.waittime	Weibull shape for latent life time
scale.censtime	Weibull scale for censoring time
shape.censtime	Weibull 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**

```
## Not run:
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)

## End(Not run)

```

lifexpect

*Predictions of life expectancies from an illness-death.***Description**

Predict life expectancies from an object of class `idm`. Life expectancies are calculated at time  $s$  for a subject who has the covariates values  $Z01$ ,  $Z02$ ,  $Z12$ . Confidence intervals are calculated.

**Usage**

```
lifexpect(x, s, Z01, Z02, Z12, nsim = 1000, CI = TRUE, ...)
```

**Arguments**

<code>x</code>	an object as returned by a call to the <code>idm</code> function.
<code>s</code>	time at prediction.
<code>Z01</code>	vector for the values of the covariates on the transition $0 \rightarrow 1$ (in the same order as the covariates within the call. The default values are all 0.
<code>Z02</code>	vector for the values of the covariates on the transition $0 \rightarrow 2$ (in the same order as the covariates within the call. The default values are all 0.
<code>Z12</code>	vector for the values of the covariates on the transition $1 \rightarrow 2$ (in the same order as the covariates within the call. The default values are all 0.
<code>nsim</code>	number of simulations for the confidence intervals calculations. The default is 1000.
<code>CI</code>	boolean: with (TRUE) or without (WRONG) confidence intervals for the life expectancies. The default is TRUE.
<code>...</code>	others arguments.

**Value**

a list containing:

<code>life.in.0.expectancy</code>	life expectancy in state 0 and confidence interval.
<code>life.expectancy.nondis</code>	life expectancy of a non-diseased subject and confidence interval.
<code>life.expectancy.dis</code>	life expectancy of a diseased subject and confidence interval.

**Author(s)**

C. Touraine

**See Also**[idm](#)**Examples**

```
## Not run:
library(lava)
library(prodlim)
set.seed(17)
d <- simulateIDM(100)
table(d$seen.ill,d$seen.exit)
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)
try(lifexpect(fitIC,s=10),silent=TRUE)

data(Paq1000)

fit <- idm(formula02=prodlim::Hist(time=t,event=death,entry=e)~certif,
           formula01=prodlim::Hist(time=list(l,r),event=dementia)~certif,
           method="Splines",
           data=Paq1000,
           conf.int=FALSE)

pred <- lifexpect(fit,s=70,t=80,Z01=c(1),Z02=c(1),Z12=c(1))

## End(Not run)
```

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=with certificate, 1=without certificate  
**gender** gender: 0=female, 1=male

### Examples

```
data(Paq1000)
```

---

plot.idm	<i>Plot method for an illness-death model</i>
----------	---

---

### 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

<code>x</code>	a <code>idmWeib</code> class object (output from calling <code>idm</code> with the (default) option <code>intensities="Weib"</code> ).
<code>conf.int</code>	If TRUE show confidence limits
<code>citype</code>	Type of confidence limits, can be "shadow" or "bars"
<code>add</code>	If TRUE add to existing plot
<code>axes</code>	If TRUE axes are drawn
<code>col</code>	Color of the lines
<code>lwd</code>	Width of the lines
<code>lty</code>	Type of the lines
<code>xlim</code>	Limits for x-axis
<code>ylim</code>	Limits for y-axis
<code>xlab</code>	Label for x-axis
<code>ylab</code>	Label for y-axis
<code>legend</code>	If TRUE a legend is drawn, which can be further controlled via <a href="#">SmartControl</a> .
<code>transition</code>	Choose one of the transition intensities: <code>c("01", "02", "12")</code> .
<code>...</code>	Passed to <a href="#">SmartControl</a>

**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](#)

**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)

## Not run:
## 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)

## End(Not run)
```

---

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 <a href="#">shr</a> 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 survival 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	<i>Predictions for an illness-death model using either a penalized likelihood approach or a Weibull parametrization.</i>
-------------	--

---

### 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, Z01, Z02, Z12, nsim = 2000, CI = TRUE,
  ...)
```

### Arguments

<code>object</code>	an <code>idm</code> class objects returned by a call to the <code>idm</code> function
<code>s</code>	time at prediction.
<code>t</code>	time for prediction.
<code>Z01</code>	vector for the values of the covariates on the transition 0 → 1 (in the same order as the covariates within the call. The default values are all 0.
<code>Z02</code>	vector for the values of the covariates on the transition 0 → 2 (in the same order as the covariates within the call. The default values are all 0.
<code>Z12</code>	vector for the values of the covariates on the transition 1 → 2 (in the same order as the covariates within the call. The default values are all 0.
<code>nsim</code>	number of simulations for the confidence intervals calculations. The default is 2000.
<code>CI</code>	boolean: with (TRUE) or without (WRONG) confidence intervals for the predicted values. The default is TRUE.
<code>...</code>	other parameters.

### Value

a list containing the following predictions with pointwise confidence intervals:

<code>p00</code>	the transition probability $p_{00}$ .
<code>p01</code>	the transition probability $p_{01}$ .
<code>p11</code>	the transition probability $p_{11}$ .
<code>p12</code>	the transition probability $p_{12}$ .
<code>p02_0</code>	the probability of direct transition from state 0 to state 2.
<code>p02_1</code>	the probability of transition from state 0 to state 2 via state 1.
<code>p02</code>	transition probability $p_{02}$ . Note that $p_{02}=p_{02\_0}+p_{02\_1}$ .
<code>F01</code>	the lifetime risk of disease. $F_{01}=p_{01}+p_{02\_1}$ .
<code>F0.</code>	the probability of exit from state 0. $F_{0.}=p_{02\_0}+p_{01}+p_{02\_1}$ .

**Author(s)**

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

**See Also**

[idm](#)

**Examples**

```
## Not run:
data(Paq1000)
library(prodlim)
fit <- idm(formula02=Hist(time=t,event=death,entry=e)~certif,
formula01=Hist(time=list(l,r),event=dementia)~certif,data=Paq1000)

pred <- predict(fit,s=70,t=80,Z01=c(1),Z02=c(1),Z12=c(1))
pred

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)

pred <- predict(fit.splines,s=70,t=80,Z01=c(1),Z02=c(1))
pred

## End(Not run)
```

---

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 <code>idm</code> function with <code>intensities="Weib"</code> .
conf.int	Confidence level.
digits	Number of digits to print.
pvalDigits	Number of digits to print for p-values.
eps	Passed to <code>format.pval</code> .
...	Not used.

**Author(s)**

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

**See Also**

[summary.idm](#), [plot.idm](#)

**Examples**

```
## Not run:
data(Paq1000)
library(prodlim)
fit.splines <- idm(formula02=Hist(time=t,event=death,entry=t0)~certif,
formula01=Hist(time=list(l,r),event=dementia)~certif,
                 formula12=~1,
                 method="Splines",
data=Paq1000)
print(fit.splines)

## End(Not run)
```

---

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 <a href="#">shr</a> function with hazard="Splines".
conf.int	confiance level.
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.

**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**

```
## Not run:  
# a penalized survival model  
library(prodlim)  
data(testdata)  
fit.su <- shr(Hist(time=list(1,r),id)~cov,data=testdata,method="Splines")  
print(fit.su)  
  
## End(Not run)
```

---

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), nknots = 7, CV = FALSE,  
    kappa = 10000, conf.int = TRUE, maxiter = 200, method = "Weib",  
    print.iter = FALSE, na.action = na.omit)
```

**Arguments**

formula	a formula object with the response on the left of a $\sim$ operator, and the terms on the right. 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 $10^{-3}$ .
nknots	number of knots for the splines to use to approximate the hazard function. Argument for the penalized likelihood approach. The default is 7.
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	if CV=FALSE, smoothing parameter; if CV=TRUE, initial value of the smoothing parameters for the cross validation search. Argument for the penalized likelihood approach.
conf.int	Boolean parameter. Equals to TRUE to calculate pointwise confidence intervals for the survival or hazard curves, FALSE otherwise. Default is TRUE.
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	standart errors.
knots	knots to approximate by M-splines the hazard function.
nknots	number of knots.
CV	a binary variable equals to 1 when search of the smoothing parameter <a href="#">kappa</a> 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@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

[shr](#), [print.shr](#), [summary.shr](#), [print.shr](#),

### Examples

```
# Weibull survival model
library(prodlim)
data(testdata)
fit.su <- shr(Hist(time=list(1,r),id)~cov,data=testdata,)

## to print
fit.su

## summary
summary(fit.su)
```

---

`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**

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

**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)
```

---

simulateIDM	<i>Sample illness-death model data</i>
-------------	--

---

**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

**See Also**

idmModel sim.idmModel

**Examples**

```
simulateIDM
simulateIDM(100)
```

---

summary.idm	<i>Summary of a fitted illness-death model</i>
-------------	--

---

**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 <code>idm</code> function with <code>intensities="Splines"</code> .
conf.int	confiance level.
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.

**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**

```
## Not run:
library(prodlm)
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)

## End(Not run)
```

---

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 <a href="#">shr</a> function.
conf.int	The confidence level.
digits	number of digits to print.
pvalDigits	number of digits to print for p-values.
eps	convergence criterion used for p-values.
...	other unused arguments.

**Author(s)**

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

**See Also**

[shr](#), [print.shr](#), [plot.shr](#)

**Examples**

```
## Not run:
# a penalized survival model
data(testdata)
library(proplim)
fit.su <- shr(Hist(time=list(1,r),id)~cov,data=testdata,method="Splines")
summary(fit.su)

# Weibull survival model
data(testdata)
fit.su <- shr(Hist(time=list(1,r),id)~cov,data=testdata)
summary(fit.su)

## End(Not run)
```

---

testdata	<i>Data set for survival models: right-censored and interval-censored data.</i>
----------	---

---

**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

## \*Topic **datasets**

Paq1000, 10  
testdata, 23

## \*Topic **ilness-death**

idm, 2

## \*Topic **methods**

lifexpect, 9  
plot.idm, 11  
plot.shr, 12  
predict.idm, 14  
print.idm, 15  
print.shr, 16  
shr, 17  
summary.idm, 21  
summary.shr, 22

idm, 2, 9, 10, 12, 14–16, 22

idmModel, 6

kappa, 5, 19

lifexpect, 9

Paq1000, 10

plot.idm, 11, 16, 22

plot.shr, 12, 13, 17, 23

predict.idm, 14

print.idm, 5, 12, 15, 22

print.shr, 16, 19, 23

shr, 13, 17, 17, 19, 23

sim.idmModel, 7, 20

simulateIDM, 21

SmartControl, 11

summary.idm, 5, 12, 16, 21

summary.shr, 17, 19, 22

testdata, 23