Package ‘survidm’

January 8, 2019

Type Package
Title Inference and Prediction in an Illness-Death Model
Version 1.2.0
Date 2019-01-02
Author Luis Meira-Machado, Marta Sestelo and Gustavo Soutinho
Maintainer Marta Sestelo <sestelo@uvigo.es>
Description Newly developed methods for the estimation of several probabilities in an illness-death model. The package can be used to obtain nonparametric and semiparametric estimates for: transition probabilities, occupation probabilities, cumulative incidence function and the sojourn time distributions. Additionally, it is possible to fit proportional hazards regression models in each transition of the Illness-Death Model. Several auxiliary functions are also provided which can be used for marginal estimation of the survival functions.
Depends R (>= 3.0.0)
Encoding UTF-8
License GPL-3
LazyData true
Imports KernSmooth, np, survival, doParallel, parallel, doRNG, foreach, TPmsm
RoxygenNote 6.0.1
NeedsCompilation yes
Repository CRAN
Date/Publication 2019-01-07 23:00:02 UTC

R topics documented:
survidm-package ......................................................... 2
Beran ........................................................................... 3
bladderIDM ................................................................. 4
CIF ............................................................................ 5
survidm-package

survidm-package

survidm: Inference and Prediction in an Illness-Death Model

Description

Newly developed methods for the estimation of several probabilities in an illness-death model. The package can be used to obtain nonparametric and semiparametric estimates for: transition probabilities, occupation probabilities, cumulative incidence function and the sojourn time distributions. Additionally, it is possible to fit proportional hazards regression models in each transition of the Illness-Death Model. Several auxiliary functions are also provided which can be used for marginal estimation of the survival functions.

Author(s)

Luis Meira-Machado, Marta Sestelo and Gustavo Soutinho.

Maintainer: Marta Sestelo, sestelo@uvigo.es

References


| Description | Estimation of the conditional distribution function of the response, given the covariate under random censoring. |

**Description**

Computes the conditional survival probability $P(T > y|Z = z)$

**Usage**

```
Beran(time, status, covariate, delta, x, y, kernel = "gaussian", bw, lower.tail = FALSE)
```

**Arguments**

- `time`: The survival time of the process.
- `status`: Censoring indicator of the total time of the process; 0 if the total time is censored and 1 otherwise.
- `covariate`: Covariate values for obtaining estimates for the conditional probabilities.
- `delta`: Censoring indicator of the covariate.
- `x`: The first time (or covariate value) for obtaining estimates for the conditional probabilities. If missing, 0 will be used.
- `y`: The total time for obtaining estimates for the conditional probabilities.
- `kernel`: A character string specifying the desired kernel. See details below for possible options. Defaults to "gaussian" where the gaussian density kernel will be used.
- `bw`: A single numeric value to compute a kernel density bandwidth.
- `lower.tail`: logical; if FALSE (default), probabilities are $P(T > y|Z = z)$ otherwise, $P(T <= y|Z = z)$.

**Details**

Possible options for argument window are "gaussian", "epanechnikov", "tricube", "boxcar", "triangular", "quartic" or "cosine".

**Author(s)**

Luis Meira-Machado, Marta Sestelo and Gustavo Soutinho

**References**

Examples

```r
obj <- with(colonIDM, survIDM(time1L, event1L, StimeL, eventL))
obj0 <- obj

#P(T>y|age=45)
library(KernSmooth)
h <- dpik(colonIDM$age)
Beran(time = obj0$Stime, status = obj0$event, covariate = colonIDM$age,
x = 45, y = 730, bw = h)

#P(T<y|age=45)
Beran(time = obj0$Stime, status = obj0$event, covariate = colonIDM$age,
x = 45, y = 730, bw = h, lower.tail = TRUE)
```

### bladderIDM

**Bladder Cancer Recurrences.**

**Description**

bladderIDM is a data frame with 8 variables and 85 observations. Data on recurrences of bladder cancer, used by many people to demonstrate methodology for recurrent event modelling.

**Usage**

data("bladderIDM")

**Format**

A data frame with 85 observations on the following 8 variables. Below a brief description is given for some of these variables.

- **t1** Time to first recurrence/censoring, whichever occurs first.
- **e1** Recurrence/censoring indicator (first recurrence=1) for the first time (t1).
- **t2** Time to second recurrence/censoring, whichever occurs first.
- **e2** Recurrence/censoring indicator (second recurrence=1) for the second time (t2).
- **t3** Time to recurrence/censoring, whichever occurs first.
- **e3** Recurrence/censoring indicator (third recurrence=1) for the third time (t3).
- **t4** Time to fourth recurrence/censoring, whichever occurs first.
- **e4** Recurrence/censoring indicator (fourth recurrence=1) for the fourth time (t4).

**References**

CIF

Nonparametric estimation of the Cumulative Incident Functions in the illness-death model

Description
This function is used to obtain nonparametric estimates of the cumulative incidence probabilities in the illness-death model. They represent the probability of one individual’s being or having been in state j at time t.

Usage
CIF(formula, s, data, conf = FALSE, n.boot = 199, conf.level = 0.95, 
z.value, bw = "dpik", window = "gaussian", method.weights = "NW", 
cluster = FALSE, ncores = NULL, presmooth = FALSE)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>formula</td>
<td>A formula object, which must have a survIDM object as the response on the left of the ~ operator and, if desired, a term on the right. The term may be a qualitative or quantitative variable. Without covariates, the right hand side should be ~ 1.</td>
</tr>
<tr>
<td>s</td>
<td>The first time for obtaining estimates for the cumulative incidence functions. If missing, 0 will be used.</td>
</tr>
<tr>
<td>data</td>
<td>A data.frame including at least four columns named <code>time1</code>, <code>event1</code>, <code>stime</code> and <code>event</code>, which correspond to disease free survival time, disease free survival indicator, time to death or censoring, and death indicator, respectively.</td>
</tr>
<tr>
<td>conf</td>
<td>Provides pointwise confidence bands. Defaults to FALSE.</td>
</tr>
<tr>
<td>n.boot</td>
<td>The number of bootstrap replicates to compute the variance of the estimator. Default is 199.</td>
</tr>
<tr>
<td>conf.level</td>
<td>Level of confidence. Defaults to 0.95 (corresponding to 95%).</td>
</tr>
<tr>
<td>z.value</td>
<td>The value of the covariate on the right hand side of formula at which the cumulative incidence probabilities are computed. For quantitative covariates, i.e. of class integer and numeric.</td>
</tr>
<tr>
<td>bw</td>
<td>A single numeric value to compute a kernel density bandwidth. Use &quot;dpik&quot; for the KernSmooth package based selector or &quot;np&quot; for the npudensbw function of the np package.</td>
</tr>
</tbody>
</table>
>window| A character string specifying the desired kernel. See details below for possible options. Defaults to "gaussian" where the gaussian density kernel will be used. |

Examples

```r
data(bladderIDM)
head(bladderIDM)
```
method.weights A character string specifying the desired weights method. Possible options are "NW" for the Nadaraya-Watson weights and "LL" for local linear weights. Defaults to "NW".

cluster A logical value. If TRUE (default), the bootstrap procedure for the confidence intervals is parallelized. Note that there are cases (e.g., a low number of bootstrap repetitions) that R will gain in performance through serial computation. R takes time to distribute tasks across the processors also it will need time for binding them all together later on. Therefore, if the time for distributing and gathering pieces together is greater than the time need for single-thread computing, it does not worth parallelize.

ncores An integer value specifying the number of cores to be used in the parallelized procedure. If NULL (default), the number of cores to be used is equal to the number of cores of the machine - 1.

presmooth A logical value. If TRUE, the presmoothed landmark estimator of the cumulative incidence function is computed.

Details

Possible options for argument window are "gaussian", "epanechnikov", "tricube", "boxcar", "triangular", "quartic" or "cosine".

Value

An object of class "survIDM" and one of the following two classes: "CIF" (Cumulative Incidence Function), and "cifIPCW" (Inverse Probability of Censoring Weighting for the Cumulative Incidence Function). Objects are implemented as a list with elements:

- est data.frame with estimates of the cumulative incidence probabilities.
- CI data.frame with the confidence intervals of the cumulative incidence probabilities.
- conf.level Level of confidence.
- s The first time for obtaining estimates for the cumulative incidence probabilities.
- t The time for obtaining the estimates of cumulative incidence probabilities.
- conf logical; if FALSE (default) the pointwise confidence bands are not given.
- callp The expression of the estimated probability.
- Nlevels The number of levels of the covariate. Provides important information when the covariate at the right hand side of formula is of class factor.
- levels The levels of the qualitative covariate (if it is of class factor) on the right hand side of formula.
- formula A formula object.
- call A call object.

Author(s)

Luis Meira-Machado, Marta Sestelo and Gustavo Soutinho.
References


Examples

```r
# Cumulative Incidence Function (CIF)
res <- CIF(survIDM(time1, event1, Stime, event) ~ 1, data = colonIDM, conf = FALSE)
res
summary(res, time=365*1:7)
plot(res, ylim=c(0, 0.6))

res01 <- CIF(survIDM(time1, event1, Stime, event) ~ 1, data = colonIDM, conf = FALSE, presmooth = TRUE)
res01
summary(res01, time=365*1:7)
plot(res01, ylim=c(0, 0.6))

# CIF for those in State 1 at time s=365, Y(s)=0
res1 <- CIF(survIDM(time1, event1, Stime, event) ~ 1, data = colonIDM, s = 365, conf = FALSE)
summary(res1, time=365*1:7)
plot(res1, ylim=c(0, 0.6))

# Conditional CIF (with a factor)
res2 <- CIF(survIDM(time1, event1, Stime, event) ~ factor(sex), data = colonIDM, s = 365, conf = FALSE)
summary(res2, time=365*1:5)
plot(res2)

res2.1 <- CIF(survIDM(time1, event1, Stime, event) ~ factor(sex), data = colonIDM, s = 365, conf = FALSE, presmooth = TRUE)
summary(res2.1, time=365*1:5)
plot(res2.1)

# Conditional CIF (with continuous covariate)
res3 <- CIF(survIDM(time1, event1, Stime, event) ~ age, data = colonIDM, z.value = 56, conf = FALSE)
summary(res3, time=365*1:6)
plot(res3)
```
colonIDM

Chemotherapy for Stage B/C colon cancer.

Description

These are data from one of the first successful trials of adjuvant chemotherapy for colon cancer. Levamisole is a low-toxicity compound previously used to treat worm infestations in animals; 5-FU is a moderately toxic (as these things go) chemotherapy agent.

Usage

data("colonIDM")

Format

A data frame with 929 observations on the following 15 variables. Below a brief description is given for some of these variables.

- **time1** Time to recurrence/censoring/death, whichever occurs first.
- **event1** Recurrence/censoring indicator (recurrence=1, alive=0).
- **Stime** Time to censoring/death, whichever occurs first.
- **event** Death/censoring indicator (death=1, alive=0).
- **rx** Treatment - Observation, Lev(amisole), Lev(amisole)+5-FU.
- **sex** Sex indicator (male=1, female=0).
- **age** Age in years.
- **obstruct** Obstruction of colon by tumour.
- **perfor** Perforation of colon.
- **adhere** Adherence to nearby organs.
- **nodes** Number of lymph nodes with detectable cancer.
- **differ** Differentiation of tumour (1=well, 2=moderate, 3=poor).
- **extent** Extent of local spread (1=submucosa, 2=muscle, 3=serosa, 4=contiguous structures).
- **surg** Time from surgery to registration (0=short, 1=long).
- **node4** More than 4 positive lymph nodes.

Source

The study is originally described in Laurie (1989). The main report is found in Moertel (1990). This data set is closest to that of the final report in Moertel (1991). A version of the data with less follow-up time was used in the paper by Lin (1994).
coxdm

References


Examples

data(colonIDM)
head(colonIDM)

coxidm(formula, data, semiMarkov = FALSE)

Description

 Fits a Cox proportional hazards regression model for each transition.

Usage

coxidm(formula, data, semiMarkov = FALSE)

Arguments

formula A formula object, which must have a survIDM object as the response on the left of the ~ operator and, if desired, a term on the right. The term may be a qualitative or quantitative variable. Without covariates, the right hand side should be ~ 1.

data A data.frame including at least four columns named time1, event1, Stime and event, which correspond to disease free survival time, disease free survival indicator, time to death or censoring, and death indicator, respectively.

semiMarkov By default, FALSE.
Value

A data frame or a list containing the following components:

- **coef**: Estimated coefficients.
- **exp(coef)**: Exponent of the estimated coefficients.
- **lower**: Lower limit of the confidence interval.
- **upper**: Upper limit of the confidence interval.
- **pvalue**: Obtained p-value testing that the coefficient is equal to zero.

Author(s)

Luis Meira-Machado, Marta Sestelo and Gustavo Soutinho.

Examples

```r
cmm1 <- coxidm(survidm(time1, event1, Stime, event) ~ age, data = colonIDM)
summary(cmm1, conf.level = 0.95)

cmm2 <- coxidm(survidm(time1, event1, Stime, event) ~ rx + sex + age + nodes,
data = colonIDM)
summary(cmm2)

cmm3 <- coxidm(survidm(time1, event1, Stime, event) ~ rx + sex + age + nodes,
data = colonIDM, semiMarkov = TRUE)
summary(cmm3)
```

---

**gbcsIDM**

*German Breast Cancer Study Data.*

Description

gbcsCS is a data frame with 16 variables and 686 observations. Cancer clinical trials are a rich source for examples of applications of methods for the analysis of time to event. Willi Sauerbrei and Patrick Royston have graciously provided us with data obtained from the German Breast Cancer Study Group, which they used to illustrate methods for building prognostic models (Sauerbrei and Royston, 1999). In the main study, a total of 720 patients with primary node positive breast cancer were recruited between July 1984 and December 1989 (see Schmoor, Olschweski and Schumacher M. 1996 and Schumacher et al. (1994)).

Usage

```r
data("gbcsIDM")
```

Format

A data frame with 686 observations on the following 16 variables.


References


Examples

data(gbcsIDM)
head(gbcsIDM)

Km

Kaplan-Meier product-limit estimate of survival

Description

This function provides survival estimates using the product-limit Kaplan-Meier estimator.

Usage

KM(time, status, t)

Arguments

time Survival time of the process.
status Censoring indicator of the survival time of the process; 0 if the survival time is censored and 1 otherwise.
t The time for obtaining survival estimates.

Author(s)

Luis Meira-Machado, Marta Sestelo and Gustavo Soutinho.

References


See Also

PKM
Examples

```r
require(survival)
obj <- with(colonIDM, survIDM(time1, event1, Stime, event))
KM(time = obj$Stime, status = obj$event, t = 1095)

fit <- survfit(Surv(obj$Stime, obj$event) ~ 1, data = obj)
summary(fit, time = 1095)$surv
```

---

**KMW**  
*Kaplan-Meier weights*

---

Description

This function returns a vector with the Kaplan-Meier weights.

Usage

```r
KMW(time, status)
```

Arguments

- `time`  
  Survival time of the process.
- `status`  
  Censoring indicator of the survival time of the process; 0 if the survival time is censored and 1 otherwise.

Value

Vector with Kaplan-Meier weights.

Author(s)

Luis Meira-Machado, Marta Sestelo and Gustavo Soutinho.

References


See Also

PKMW
Examples

```r
obj <- with(colonIDM, survIDM(time1, event1, Stime, event))
kmw <- KMW(time = obj$Stime, status = obj$event)

require(survival)
colon.surv <- survfit(Surv(time, event) ~ 1, obj)
times <- summary(colon.surv)$time
surv <- summary(colon.surv)$surv
nevent <- summary(colon.surv)$n.event
p <- match(obj$Stime, times)
kmw2 <- -diff(c(1, surv))/nevent
kmw2 <- kmw2[p]*obj$event
kmw2[is.na(kmw2)] <- 0
all.equal(kmw, kmw2)
```

---

**Description**

Computes local linear weights based on Kernel smoothing.

**Usage**

```r
LLW(x, kernel = "gaussian", bw, t1)
```

**Arguments**

- `x` Covariate values for obtaining estimates for the conditional probabilities. If missing, unconditioned probabilities will be computed.
- `kernel` A character string specifying the desired kernel. See details below for possible options. Defaults to "gaussian" where the gaussian density kernel will be used.
- `bw` A single numeric value to compute a kernel density bandwidth.
- `t1` Covariate value to compute the weight at.

**Details**

Possible options for argument window are "gaussian", "epanechnikov", "tricube", "boxcar", "triangular", "quartic" or "cosine".

**Value**

A vector with local linear weights.

**Author(s)**

Luis Meira-Machado, Marta Sestelo and Gustavo Soutinho.
See Also

Examples

LLW(x = colonIDM$age, bw = 3, t1 = 60)

markov.test

This function is used to test the markov assumption in the illness-death model.

Description

The markov assumption may be tested including the sojourn time in the initial state, "times1", and other covariates in the Cox model.

Usage

markov.test(formula, s, nm.method = "LM", data)

Arguments

formula A formula object, which must have a survIDM.

s The first time for obtaining a graphical test of markovianity by comparison of the estimates for transition probabilities.

nm.method The non-markov method used to compute the transition probabilities. Defaults to "LM".

data A data.frame including at least four columns named time1, event1, Stime and event, which correspond to disease free survival time, disease free survival indicator, time to death or censoring, and death indicator, respectively.

Details

The markov assumption may be tested including the sojourn time in the initial state, "times1", and other covariates in the Cox model. A graphical test for Markovianity is also available.

Value

cox.markov.test An object of class coxph representing the fit. See coxph.object for details.

TEstimates Dataframe with estimates of the transition probabilities for Aalen-Johansen estimator (markovian) and for non-markov estimator. Confidence intervals for the transition probability from State 1 to State 2 are also available.

nm.method The non-markov method used to compute the transition probabilities.

s The first time for obtaining a graphical test of markovianity by comparison of the estimates for transition probabilities.

call A call object.
Author(s)
Luis Meira-Machado, Marta Sestelo and Gustavo Soutinho.

References

Examples

```r
mk <- markov.test(survIDM(time1, event1, Stime, event) ~ 1, s = 365*4, nm.method = "LM", data = colonIDM)
mk$cox.markov.test
mk$TPEstimates
mk$nm.method
plot(mk)
```

```r
nevents(dataidm, state.names=NULL)
```

Description
Given a dataset of class "survIDM", this function counts the number of observed transitions in the multi-state model.

Usage
```r
nevents(dataidm, state.names=NULL)
```

Arguments
dataidm A dataframe including at least four columns named time1, event1, Stime and event, which correspond to disease free survival time, disease free survival indicator, time to death or censoring, and death indicator, respectively.

state.names Names for the transition states. If NULL (default), transition states are named by "healthy", "illness" and "death".

Details
The columns of the dataset needs to have the format of class "survIDM", which holds the transition matrix of the multi-state model.
Author(s)

Luis Meira-Machado, Marta Sestelo and Gustavo Soutinho.

References


Examples

```r
nevents(colonIDM)

nevents(colonIDM, c('State1', 'State2', 'State3'))
```

Description

Computes the Nadaraya-Watson weights.

Usage

```r
nww(covariate, x, kernel = "gaussian", bw)
```

Arguments

- `covariate`: Covariate values for obtaining weights.
- `x`: Covariate value to compute the weight at.
- `kernel`: A character string specifying the desired kernel. See details below for possible options. Defaults to "gaussian" where the gaussian density kernel will be used.
- `bw`: A single numeric value to compute a kernel density bandwidth.

Details

Possible options for argument window are "gaussian", "epanechnikov", "tricube", "boxcar", "triangular", "quartic" or "cosine".

Value

A vector with Nadaraya-Watson weights.
**PKM**

**Author(s)**

Luis Meira-Machado, Marta Sestelo and Gustavo Soutinho.

**See Also**

LLW

**Examples**

```r
nww(covariate = colonIDM$age, x=40, kernel = "gaussian", bw = 3)
```

| PKM | Presmoothed Kaplan-Meier product-limit estimate of survival |

**Description**

This function provides survival estimates using the presmoothed product-limit Kaplan-Meier estimator.

**Usage**

```r
PKM(time, status, t)
```

**Arguments**

- **time**: Survival time of the process.
- **status**: Censoring indicator of the survival time of the process; 0 if the survival time is censored and 1 otherwise.
- **t**: The time for obtaining survival estimates.

**Author(s)**

Luis Meira-Machado, Marta Sestelo and Gustavo Soutinho.

**References**


**See Also**

KM
Examples

obj <- with(colonIDM, survIDM(time1, event1, Stime, event))
Pkmw(time = obj$Stime, status = obj$event, t = 1095)

---

**PKMW**  
*Presmoothed Kaplan-Meier weights*

**Description**

This function returns a vector with the presmoothed Kaplan-Meier weights.

**Usage**

`PKMW(time, status)`

**Arguments**

- `time`: Survival time of the process.
- `status`: Censoring indicator of the survival time of the process; 0 if the survival time is censored and 1 otherwise.

**Value**

Vector with presmoothed Kaplan-Meier weights.

**Author(s)**

Luis Meira-Machado, Marta Sestelo and Gustavo Soutinho.

**References**


**See Also**

`kmw`

**Examples**

obj <- with(colonIDM, survIDM(time1, event1, Stime, event))
PKMW(time = obj$Stime, status = obj$event)
Description

It draws the estimated probabilities.

Usage

```r
## S3 method for class 'survIDM'
plot(x = object, y = NULL, trans = "all",
     func = "distribution", conf = NULL,
     type = NULL, conftype = NULL, col = 1:6, confcol = 1:6, lty = 1, conflty = 2,
     xlab = "Time (years)", ylab = NULL, ylim = NULL, xlim = NULL, ...) #
```

Arguments

- **x**: An object of class "survIDM".
- **y**: NULL
- **trans**: The transition probabilities plotted. It is used only when the object is of class "AJ", "LIDA", "LM", "PLM", "LMAJ", "PLMAJ", "PMAJ" and "tpIPCW". Possible options are "all" (default), "00", "01", "02", "11" or "12".
- **func**: It is used only when the object is of class "soj" or "sojIPCW". The type of curve to be drawn ("distribution" or "survival"). Default to "distribution".
- **conf**: Draw the confidence intervals into the plot. By default it is NULL, they are drawn if the "surv" object contains them.
- **type**: The type of plot that should be drawn. See details `par` for possible options. Defaults to "s" for the draw be stair steps.
- **conftype**: The type of plot that should be drawn for confidence intervals. See details `par` for possible options. Defaults to "s" for the draw be stair steps.
- **col**: Vector of colors. Colors are used cyclically.
- **confcol**: Vector of colors for the confidence intervals. Colors are used cyclically.
- **lty**: The line type. Line types can either be specified as an integer (0 = blank, 1 = solid (default), 2 = dashed, 3 = dotted, 4 = dotdash, 5 = longdash, 6 = twodash). See details in `par`.
- **conflty**: The line type for confidence intervals. Line types can either be specified as an integer (0 = blank, 1 = solid (default), 2 = dashed, 3 = dotted, 4 = dotdash, 5 = longdash, 6 = twodash).
- **xlab**: A title for the x axis: see `title`.
- **ylab**: A title for the y axis: see `title`.
- **ylim**: The y limits of the plot.
- **xlim**: The x limits of the plot.
- **...**: Other options.
Value

No value is returned.

Author(s)

Luis Meira-Machado, Marta Sestelo and Gustavo Soutinho.

Examples

```r
res <- tprob(survidm(time1, event1, Stime, event) ~ 1, s = 0, 
method = "AJ", conf = FALSE, data = colonIDM)

plot(res)
plot(res, trans = "02")

res1 <- tprob(survidm(time1, event1, Stime, event) ~ factor(sex), s = 365, 
method = "AJ", conf = FALSE, data = colonIDM)

plot(res1, trans="02", ylim=c(0,0.5))

res2 <- CIF(survIDM(time1, event1, Stime, event) ~ age, data = colonIDM, 
z.value = 56, conf = FALSE)

plot(res2)

res3 <- sojourn(survidm(time1, event1, Stime, event) ~ factor(sex), 
data = colonIDM, conf = FALSE, conf.level = 0.95)

plot(res3)
```

sojourn Nonparametric estimation of the Sojourn time distributions in the illness-death model

Description

This function is used to obtain nonparametric estimates of the sojourn probabilities in the illness-death model.

Usage

```r
sojourn(formula, data, conf = FALSE, n.boot = 199, conf.level = 0.95, 
z.value, bw = "dpik", window = "gaussian", method.weights = "NW", 
method = "LM", presmooth = FALSE, cluster = FALSE, ncores = NULL)
```
Arguments

**formula**  
A formula object, which must have a `survIDM` object as the response on the left of the `~` operator and, if desired, a term on the right. The term may be a qualitative or quantitative variable. Without covariates, the right hand side should be `~ 1`.

**data**  
A data.frame including at least four columns named `time1`, `event1`, `Stime` and `event`, which correspond to disease free survival time, disease free survival indicator, time to death or censoring, and death indicator, respectively.

**conf**  
Provides pointwise confidence bands. Defaults to `FALSE`.

**n.boot**  
The number of bootstrap replicates to compute the variance of the non-Markovian estimator. Default is 199.

**conf.level**  
Level of confidence. Defaults to 0.95 (corresponding to 95%).

**z.value**  
The value of the covariate on the right hand side of formula at which the sojourn probabilities are computed. For quantitative covariates, i.e. of class integer and numeric.

**bw**  
A single numeric value to compute a kernel density bandwidth. Use "dpik" for the KernSmooth package based selector or "np" for the `npudensbw` function of the np package.

**window**  
A character string specifying the desired kernel. See details below for possible options. Defaults to "gaussian" where the gaussian density kernel will be used.

**method.weights**  
A character string specifying the desired weights method. Possible options are "NW" for the Nadaraya-Watson weights and "LL" for local linear weights. Defaults to "NW".

**method**  
The method used to compute the sojourn estimates. Possible options are "LM" and "Satten-Datta". Defaults to "LM".

**presmooth**  
- A logical value. If `TRUE`, the presmoothed landmark estimator of the sojourn function is computed. Only valid for `method = "LM"`.

**cluster**  
A logical value. If `TRUE` (default), the bootstrap procedure for the confidence intervals is parallelized. Note that there are cases (e.g., a low number of bootstrap repetitions) that R will gain in performance through serial computation. R takes time to distribute tasks across the processors also it will need time for binding them all together later on. Therefore, if the time for distributing and gathering pieces together is greater than the time need for single-thread computing, it does not worth parallelize.

**ncores**  
An integer value specifying the number of cores to be used in the parallelized procedure. If `NULL` (default), the number of cores to be used is equal to the number of cores of the machine - 1.

Details

Possible options for argument `window` are "gaussian", "epanechnikov", "tricube", "boxcar", "triangular", "quartic" or "cosine".
Value

An object of class "survIDM" and one of the following two classes: "soj" (Sojourn Time Distribution), and "sojIPCW" (Inverse Probability of Censoring Weighting for the Sojourn Time Distribution). Objects are implemented as a list with elements:

- **est** data.frame with estimates of the sojourn probabilities.
- **CI** data.frame with the confidence intervals of the sojourn probabilities.
- **conf.level** Level of confidence.
- **t** The time for obtaining the estimates of sojourn probabilities.
- **conf** logical; if FALSE (default) the pointwise confidence bands are not given.
- **callp** The expression of the estimated probability.
- **Nlevels** The number of levels of the covariate. Provides important information when the covariate at the right hand side of formula is of class factor.
- **levels** The levels of the qualitative covariate (if it is of class factor) on the right hand side of formula.
- **formula** A formula object.
- **call** A call object.

Author(s)

Luis Meira-Machado, Marta Sestelo and Gustavo Soutinho.

References


Examples

```r
res <- sojourn(survIDM(time1, event1, Stime, event) ~ 1, data = colonIDM, conf = FALSE, conf.level = 0.95)
res
summary(res, time=365*1:6)
plot(res)

res1 <- sojourn(survIDM(time1, event1, Stime, event) ~ 1, data = colonIDM, conf = FALSE, conf.level = 0.95, method = "LM", presmooth = TRUE)
res1
summary(res1, time=365*1:6)
plot(res1)
```

# not run:
# res2 <- sojourn(survIDM(time1, event1, Stime, event) ~ 1, data = colonIDM, conf = FALSE, conf.level = 0.95, method = "Satten-Datta")
# res2
# with a factor
res3 <- sojourn(survIDM(time1, event1, Stime, event) - factor(sex),
data = colonIDM, conf = FALSE, conf.level = 0.95)
summary(res3, time=365*1:6)
plot(res3)

# with a qualitative covariate
res4 <- sojourn(survIDM(time1, event1, Stime, event) - age, data = colonIDM,
z.value = 56, conf = FALSE, conf.level = 0.95)
summary(res4, time=365*1:6)
plot(res4)

summary.cmm

Summarizing fits of cmm class

Description

Produces a summary of a fitted coxidm model (proportional hazards regression model in each transition of the Illness-Death Model.

Usage

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

Arguments

- **object**
  A fitted cmm object as produced by coxidm().

- **conf.level**
  Level for computation of the confidence intervals. If set to FALSE no confidence intervals are printed.

- **...**
  Other options.

Value

A data frame or a list containing the following components:

- **coef**
  Estimated coefficients.

- **exp(coef)**
  Exponent of the estimated coefficients.

- **lower 0.95**
  Lower limit of the confidence interval.

- **upper 0.95**
  Upper limit of the confidence interval.

- **pvalue**
  Obtained pvalue testing that the coefficient is equals to zero.
Author(s)
Luis Meira-Machado, Marta Sestelo and Gustavo Soutinho.

Examples

```r
cmm1 <- coxidm(survIDM(time1, event1, stime, event) ~ age, data = colonIDM)
summary(cmm1, conf.level = 0.95)

cmm2 <- coxidm(survIDM(time1, event1, stime, event) ~ rx + sex + age + nodes, data = colonIDM)
summary(cmm2)

cmm3 <- coxidm(survIDM(time1, event1, stime, event) ~ rx + sex + age + nodes, data = colonIDM, semiMarkov = TRUE)
summary(cmm3)
```

### summary.survIDM

#### Summarizing fits of survIDM class

Description

Returns a data.frame or list containing the estimates of the probabilities, its confidence limits and other information.

Usage

```r
## S3 method for class 'survIDM'
summary(object, times = NULL, ...)
```

Arguments

- **object** A fitted survIDM object as produced by `tprob()`, `CIF()` or `sojourn()`.
- **times** Vector of times; the returned data frame will contain 1 row for each time. Missing values are not allowed.
- **...** For future methods.

Value

A data frame or a list containing the following components:

- **y** The total time for obtaining the estimates of the probabilities.
- **est** Estimates of the probability.
- **lower 95% CI** The lower probabilities of the interval.
- **upper 95% CI** The upper probabilities of the interval.
survIDM

Author(s)
Luis Meira-Machado, Marta Sestelo and Gustavo Soutinho.

Examples

```r
fit <- tprob(survIDM(time1, event1, Stime, event) ~ 1, s = 365,
method = "AJ", conf = TRUE, conf.level = 0.95,
conf.type = "linear", n.boot = 50, data = colonIDM)

summary(fit)
summary(fit, times = c(400, 1000, 2900))
```

survIDM

Create a survIDM object

Description

Creates a "survIDM" object, usually used as a response variable in a model formula.

Usage

```r
survIDM(time1, event1, Stime, event, ...)
```

Arguments

- **time1**: First time or censoring time.
- **event1**: Indicator of the first time; 0 if the first time is censored and 1 otherwise.
- **Stime**: The total time of the process.
- **event**: Censoring indicator of the survival time of the process; 0 if the total time is censored and 1 otherwise.
- **...**: Other options.

Details

Arguments in this function must be introduced in the following order: time1, event1, Stime and event, where time1 and Stime are ordered event times and event1 and event their corresponding indicator statuses.

Value

An object of class "survIDM". "survIDM" objects are implemented as a single dataframe.

Author(s)
Luis Meira-Machado, Marta Sestelo and Gustavo Soutinho.
Examples

with(colonIDM, survIDM(time1, event1, Stime, event))

```
tprob Nonparametric estimation of transition probabilities in the illness-death model
```

Description

This function is used to obtain nonparametric estimates of the transition probabilities in the illness-death model.

Usage

```
tprob(formula, s, method = "AJ", conf = FALSE, conf.level = 0.95, conf.type = "linear", n.boot = 199, data, z.value, bw = "dpik", window = "gaussian", method.weights = "NW", cluster = FALSE, ncores = NULL, na.rm = TRUE)
```

Arguments

- **formula**: A formula object, which must have a survIDM object as the response on the left of the ~ operator and, if desired, a term on the right. The term may be a qualitative or quantitative variable. Without covariates, the right hand side should be ~ 1.
- **s**: The first time for obtaining estimates for the transition probabilities. If missing, 0 will be used.
- **method**: The method used to compute the transition probabilities. Possible options are "AJ", "LIDA", "LM", "PLM", "LMAJ", "PLMAJ", "PAJ", "IPCW", and "breslow". Defaults to "AJ". The "IPCW" method is recommended to obtain conditional transition probabilities (i.e., with a quantitative term on the right hand side of formula).
- **conf**: Provides pointwise confidence bands. Defaults to FALSE.
- **conf.level**: Level of confidence. Defaults to 0.95 (corresponding to 95%).
- **conf.type**: Method to compute the confidence intervals. Transformation applied to compute confidence intervals. Possible choices are "linear", "log", "log-log" and "bootstrap". Default is "linear".
- **n.boot**: The number of bootstrap replicates to compute the variance of the non-Markovian estimator. Default is 199.
- **data**: A data.frame including at least four columns named time1, event1, Stime and event, which correspond to disease free survival time, disease free survival indicator, time to death or censoring, and death indicator, respectively.
- **z.value**: The value of the covariate on the right hand side of formula at which the transition probabilities are computed. For quantitative covariates, i.e. of class integer and numeric.
bw A single numeric value to compute a kernel density bandwidth. Use "dpik" for the KernSmooth package based selector or "np" for the 'npudensbw' function of the np package.

window A character string specifying the desired kernel. See details below for possible options. Defaults to "gaussian" where the gaussian density kernel will be used.

method.weights A character string specifying the desired weights method. Possible options are "NW" for the Nadaraya-Watson weights and "LL" for local linear weights. Defaults to "NW".

cluster A logical value. If TRUE (default), the bootstrap procedure for the confidence intervals is parallelized. Note that there are cases (e.g., a low number of bootstrap repetitions) that R will gain in performance through serial computation. R takes time to distribute tasks across the processors also it will need time for binding them all together later on. Therefore, if the time for distributing and gathering pieces together is greater than the time need for single-thread computing, it does not worth parallelize.

ncores An integer value specifying the number of cores to be used in the parallelized procedure. If NULL (default), the number of cores to be used is equal to the number of cores of the machine - 1.

na.rm A logical value indicating whether NA values should be stripped in the computation.

Details Possible options for argument window are "gaussian", "epanechnikov", "tricube", "boxcar", "triangular", "quartic" or "cosine". Possible methods are:

- AJ Aalen-Johansen estimator
- PAJ Presmoothed Aalen-Johansen estimator
- LIDA LIDA estimator
- LM Landmark approach estimator
- PLM Presmoothed Landmark approach estimator
- LMAJ Landmark approach Aalen-Johansen estimator
- PLDAJ Presmoothed Landmark approach Aalen-Johansen estimator
- tpIPCW Inverse Probability of Censoring Weighting for Transition Probabilities
- tpBreslow Breslow method

Value An object of class "survIDM" and one of the following five classes: "AJ", "LIDA", "LM", "PLM", "LMAJ", "PLMAJ", "PAJ", "tpIPCW" and "tpBreslow". Objects are implemented as a list with elements:

- est data.frame with estimates of the transition probabilities.
- CI data.frame with the confidence intervals of the transition probabilities.
conf.level  Level of confidence.
s  The first time for obtaining estimates for the transition probabilities.
t  The time for obtaining the estimates of transition probabilities.
conf  logical; if FALSE (default) the pointwise confidence bands are not given.
conf.type  Type of the confidence interval.
callp  The expression of the estimated probability.
Nlevels  The number of levels of the covariate. Provides important information when the
  covariate at the right hand side of formula is of class factor.
levels  The levels of the qualitative covariate (if it is of class factor) on the right hand
  side of formula.
formula  A formula object.
call  A call object.

Author(s)
Luis Meira-Machado, Marta Sestelo and Gustavo Soutinho.

References

Examples
# Aalen-Johansen
res <- tprob(survIDM(time1, event1, Stime, event) ~ 1, s = 0, method = "AJ",
conf = FALSE, data = colonIDM)
summary(res, time=365*1:6)
plot(res)
# Transition Probabilities Pij(t)-Pij(365,t)
# LIDA
res1 <- tprob(survIDM(time1, event1, Stime, event) ~ 1, s = 365,
method = "LIDA", conf = FALSE, data = colonIDM)
summary(res1, time = 365*1:6)
plot(res1)
plot(res1, trans = "01", ylim = c(0,0.15))
# Landmark (LM)
res2 <- tprob(survIDM(time1, event1, Stime, event) ~ 1, s = 365,
method = "LM", conf = FALSE, data = colonIDM)
summary(res2, time = 365*1:6)
plot(res2)
# Presmoothed LM
res3 <- tprob(survIDM(time1, event1, Stime, event) ~ 1, s = 365,
  method = "PLM", conf = FALSE, data = colonIDM)
summary(res3, time = 365*1:6)
plot(res3)
# Conditional transition probabilities
# With factor
res4 <- tprob(survIDM(time1, event1, Stime, event) ~ factor(sex), s = 365,
  method = "AJ", conf = FALSE, data = colonIDM)
summary(res4, time = 365*1:6)
plot(res4, trans = "02", ylim = c(0,0.5))
res5 <- tprob(survIDM(time1, event1, Stime, event) ~ rx, s = 365,
  method = "breslow", z.value = 'Lev', conf = FALSE, data = colonIDM)
summary(res5, time = 365*1:6)
plot(res5, trans = "02", ylim = c(0,0.5))
## Not run: # with continuous covariate (IPCW and Breslow Method)
res6 <- tprob(survIDM(time1, event1, Stime, event) ~ age, s = 365,
  method = "IPCW", z.value = 48, conf = FALSE, data = colonIDM,
  bw = "dpik", window = "gaussian", method.weights = "NW")
summary(res6, time = 365*1:6)
plot(res6)
#res7 <- tprob(survIDM(time1, event1, Stime, event) ~ age, s =365,
  # method = "breslow", z.value = 60, conf = FALSE,
  # data = colonIDM)
summary(res7, time = 365*1:6)
plot(res7)
res8 <- tprob(survIDM(time1, event1, Stime, event) ~ age, s =365,
  method = "breslow", conf.type = 'bootstrap', n.boot = 2,
  z.value = 60, conf = TRUE, data = colonIDM)
summary(res8, time = 365*1:6)
plot(res8)
res9 <- tprob(survIDM(time1, event1, Stime, event) ~ rx, s =365,
  method = "breslow", conf.type='bootstrap',
  conf = TRUE, data =colonIDM)
summary(res9, time = 365*1:6)
plot(res9, trans = "02", ylim = c(0,0.5))
# more than a covariate (Breslow Method)
res10 <- tprob(survIDM(time1, event1, Stime, event) ~ nodes + factor(rx),
  s = 365, method = "breslow", conf = TRUE, data =colonIDM)
summary(res10,t = 365*1:5)
plot(res10)
res11 <- tprob(survIDM(time1, event1, Stime, event) ~ nodes + factor(rx),
  s = 365, method = "breslow", z.value = c(10,'Obs'), conf = TRUE,
  data = colonIDM)
summary(res11,t = 365*1:5)
plot(res11)
# more than a covariate for Non Linear Models (Breslow Method)
res12 <- tprob(survIDM(time1, event1, Stime, event) ~ pspline(age) + nodes +
  factor(rx), s =365, method = "breslow", conf = TRUE, data = colonIDM)
summary(res12,t = 365*1:5)
plot(res12)
# Confidence intervals
res13 <- tprob(survIDM(time1, event1, Stime, event) ~ 1, s = 365,
  method = "AJ", conf = TRUE, n.boot = 5, conf.level = 0.95,
conf.type = "log", data = colonIDM)
summary(res13, time=365*1:7)
plot(res13)
res14 <- tprob(survIDM(time1, event1, Stime, event) ~ pspline(age) + nodes +
  factor(rx), s =365, method = "breslow", conf.type = 'bootstrap', conf = TRUE,
  conf.level = 0.95, data = colonIDM)
summary(res14, t = 365*1:5)
plot(res14)

## End(Not run)
Index

*Topic **datasets**
  - bladderIDM, 4
  - colonIDM, 8
  - gbcsIDM, 10

Beran, 3
bladderIDM, 4

CIF, 5
colonIDM, 8
coxidm, 9

gbcsIDM, 10
KM, 11, 17
KMW, 12, 18

LLW, 13, 17
markov.test, 14

nevents, 15
NW, 14, 16

par, 19
PKM, 11, 17
PKMW, 12, 18
plot.survIDM, 19
print.cmm(summary.cmm), 23
print.survIDM(summary.survIDM), 24

sojourn, 20
summary.cmm, 23
summary.survIDM, 24
survIDM, 25
survidm(survidm-package), 2
survidm-package, 2

title, 19
tprob, 26