Description

Newly developed methods for the estimation of the conditional survival function are implemented in this package. The condSURV package implements nonparametric and semiparametric estimators for these quantities. The package also implements feasible estimation methods for these quantities conditionally on current or past covariate measures. Other related estimators are also implemented in the package. One of these estimators is the Kaplan-Meier estimator typically assumed to estimate the survival function. A modification of the Kaplan-Meier estimator, based on a preliminary estimation (presmoothing) of the censoring probability for the survival time, given the available information is also implemented.

Author(s)

Luis Meira-Machado and Marta Sestelo.

Maintainer: Marta Sestelo, sestelo@uvigo.es

References


Beran

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, $P(T \leq y|Z = z)$.

Details

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

Author(s)

Luis Meira-Machado and Marta Sestelo

References

Examples

```r
obj <- with(colonCS, survCS(time1, event1, Stime, event))

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

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

**bladderCS**  
*Bladder Cancer Recurrences.*

**Description**

bladderCS 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.

**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**


**Examples**

```r
data(bladderCS)
head(bladderCS)
```
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.

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).

References


Examples

```r
data(colonCS)
head(colonCS)
```

gbcsCS

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)).

Format

A data frame with 686 observations on the following 16 variables. Below a brief description is given for some of these variables.

- **rectime** Time to recurrence/censoring, whichever occurs first.
- **censrec** Recurrence/censoring indicator (recurrence=1, alive=0).
- **survtime** Time to censoring/death, whichever occurs first.
- **censdead** Death/censoring indicator (death=1, alive=0).
- **age** Age in years.
- **size** Tumour size.
- **deathdate**
- **diagdateb**
- **estrg_recp**
References


Examples

data(gbcsCS)
head(gbcsCS)

---

**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.
KMW

Kaplan-Meier weights.

Description

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

Usage

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 and Marta Sestelo
References


See Also

PKMW

Examples

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

require(survival)
colon.surv <- survfit(Surv(Stime, 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)
```

LLW

**Local linear weights.**

Description

Computes local linear weights based on Kernel smoothing.

Usage

`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 and Marta Sestelo

See Also

NWW

Examples

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

Nadaraya-Watson weights.

Description

Computes the Nadaraya-Watson weights.

Usage

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".
**PKM**

**Value**

A vector with Nadaraya-Watson weights.

**Author(s)**

Luis Meira-Machado and Marta Sestelo

**See Also**

LLW

**Examples**

```r
epw(covariate = colonCS$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 and Marta Sestelo

**References**


PKMW

See Also

KM

Examples

obj <- with(colonCS, survCS(time1, event1, Stime, event))
PKM(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 and Marta Sestelo

References


See Also

KMW
Examples

```r
obj <- with(colonCS, survCS(time1, event1, Stime, event))
PKMW(time = obj$Stime, status = obj$event)
```

Description

It draws the estimated conditional survival probabilities.

Usage

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

Arguments

- **x**: An object of class "survCS".
- **y**: NULL
- **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.
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 \texttt{par}.

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).

A title for the $x$ axis: see \texttt{title}.

A title for the $y$ axis: see \texttt{title}.

The $y$ limits of the plot.

The $x$ limits of the plot.

Other options.

Value

No value is returned.

Author(s)

Luis Meira-Machado and Marta Sestelo

Examples

\begin{verbatim}
fit1 <- survCOND(survCS(time1, event1, Stime, event) ~ 1, x = 365,
data = colonCS, method = "LDM", conf = TRUE)
plot(fit1, xlab = "Time (days)", ylab = "S(y|365)", ylim = c(0.5, 1))
fit4 <- survCOND(survCS(time1, event1, Stime, event) ~ rx,
                 x = 365, data = colonCS, method = "LDM")
plot(fit4, xlab = "Time (days)", ylab = "S(y|365)", ylim = c(0.5, 1))
\end{verbatim}

\textbf{summary.survCS} \hspace{1cm} \textit{Summarizing fits of \texttt{survCS} class}

Description

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

Usage

\begin{verbatim}
## S3 method for class 'survCS'
summary(object, times = NULL, ...)
\end{verbatim}
Arguments

object  A fitted survCS object as produced by survCOND().
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 conditional survival probabilities.
est  Estimates of the conditional survival probability.
lower 95% CI  The lower conditional survival probabilities of the interval.
upper 95% CI  The upper conditional survival probabilities of the interval.

Author(s)

Luis Meira-Machado and Marta Sestelo

References


Examples

fit <- survCOND(survCS(time1, event1, Stime, event) ~ 1, x = 365, data = colonCS, method = "LDM", conf = TRUE, n.boot = 50, cluster = FALSE)
summary(fit)
summary(fit, times = c(400, 1000, 2900))

---

survCOND  Conditional survival probabilities based on the Kaplan-Meier weights, Landmark approaches and Inverse probability of censoring weighted.

Description

Provides estimates for the conditional survival probabilities based on Kaplan-Meier weighted estimators, the Landmark approaches and Inverse probability of censoring weighted.
Usage

```
survCOND(
    formula,
    x,
    y,
    lower.tail = FALSE,
    method = "LDM",
    presmooth = FALSE,
    conf = TRUE,
    n.boot = 200,
    data,
    conf.level = 0.95,
    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 survCS 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. For a single survival curve the right hand side should be `~ 1`.
- **x**: Time or vector of times for the conditional event(s).
- **y**: The total time for obtaining estimates for the conditional survival probabilities.
- **lower.tail**: Vector of logical values with the same size as `x`. If `x` has dimension one and if `lower.tail = FALSE` (default), probabilities are \( P(T > y | T_1 > x) \) otherwise, \( P(T > y | T_1 = x) \). If the conditional event is 2-dimensional, then, for example, given \( x = c(x_1, x_2) \) and `lower.tail = c(TRUE, FALSE)` must be used to obtain probabilities \( P(T > y | T_1 = x_1, T_2 > x_2) \). Multi-dimensional conditional events are introduced similarly.
- **method**: The method used to compute the conditional survival function. Possible options are "LDM" and "KMW". Defaults to "LDM".
- **presmooth**: A logical value. If TRUE, the presmoothed landmark estimator of the conditional survival function is computed. Only valid for method = "LDM".
- **conf**: Provides pointwise confidence bands. Defaults to TRUE.
- **n.boot**: The number of bootstrap samples. Defaults to 200 samples.
- **data**: A data frame in which to interpret the variables named in the `formula` argument.
- **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 conditional survival 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".

Value

An object of class "survCS" and one of the following four classes: "KMW", "LMD", "PLDM" and "IPCW". Objects are implemented as a list with elements:

est
data.frame with estimates of the conditional probabilities.
estimate
Estimates of the conditional survival probability.
LCI
The lower conditional survival probabilities of the interval.
UCI
The upper conditional survival probabilities of the interval.
conf.level
Level of confidence.
y
The total time for obtaining the estimates of the conditional survival probabilities.
x
The first time for obtaining the estimates of the conditional survival probabilities.
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.
conf
logical; if FALSE (default) the pointwise confidence bands are not given.
callp
The expression of the estimated probability.
levels
The levels of the qualitative covariate (if it is of class factor) on the right hand side of formula.
Author(s)
Luis Meira-Machado and Marta Sestelo

References

Examples

```r
fit <- survCOND(survCS(time1, event1, Stime, event) ~ 1, x = 365, y = 730, data = colonCS, method = "KMW", conf = FALSE)
fit1 <- survCOND(survCS(time1, event1, Stime, event) ~ 1, x = 365, data = colonCS, method = "LDM", conf = FALSE)
fit2 <- survCOND(survCS(time1, event1, Stime, event) ~ 1, x = 365, data = colonCS, method = "LDM", lower.tail = TRUE, conf = FALSE)
fit3 <- survCOND(survCS(time1, event1, Stime, event) ~ 1, x = 365, y = c(730, 1095, 1460), data = colonCS, method = "LDM", presmooth = TRUE, lower.tail = TRUE, conf = TRUE, n.boot = 100, conf.level = 0.95, cluster = FALSE)
fit4 <- survCOND(survCS(time1, event1, Stime, event) ~ rx, x = 365, data = colonCS, method = "LDM", conf = FALSE)
fit5 <- survCOND(survCS(time1, event1, Stime, event) ~ factor(sex), x = 365, data = colonCS, method = "LDM", conf = FALSE)

## Not run:
fit6 <- survCOND(survCS(time1, event1, Stime, event) ~ age, x = 365, y = 730, z.value = 48, data = colonCS, conf = TRUE)
## End(Not run)
```

survCS

Create a survCS object.

Description
Creates a "survCS" object, usually used as a response variable in a model formula.
Usage

```
survCS(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`, `time2`, ..., `Stime` and `event`, where `time1`, `time2`, ..., `Stime` are ordered event times and `event1`, `event2`, ..., `event` their corresponding indicator statuses.

Value

An object of class "survCS" and of class "surv". "survCS" objects are implemented as a single dataframe.

Author(s)

Luis Meira-Machado and Marta Sestelo

Examples

```
with(colonCS, survCS(time1, event1, Stime, event))
```
Index

* datasets
  bladderCS, 4
  colonCS, 5
  gbcsCS, 6

Beran, 2
bladderCS, 4

colonCS, 5
condSURV (condSURV-package), 2
condSURV-package, 2

gbcsCS, 6

KM, 7, 12
KMW, 8, 12

LLW, 9, 11

NWW, 10, 10

par, 13, 14
PKM, 8, 11
PKMW, 9, 12
plot.survCS, 13
print.survCS (summary.survCS), 14

summary.survCS, 14
survCOND, 15
survCS, 18

title, 14