

Package: survivalREC (via r-universe)

September 8, 2024

Type Package

Title Nonparametric Estimation of the Distribution of Gap Times for Recurrent Events

Version 1.1

Date 2023-08-09

Description Provides estimates for the bivariate and trivariate distribution functions and bivariate and trivariate survival functions for censored gap times. Two approaches, using existing methodologies, are considered: (i) the Lin's estimator, which is based on the extension the Kaplan-Meier estimator of the distribution function for the first event time and the Inverse Probability of Censoring Weights for the second time (Lin DY, Sun W, Ying Z (1999) <[doi:10.1093/biomet/86.1.59](https://doi.org/10.1093/biomet/86.1.59)> and (ii) another estimator based on Kaplan-Meier weights (Una-Alvarez J, Meira-Machado L (2008) <https://w3.math.uminho.pt/~lmachado/Biometria_conference.pdf>). The proposed methods are the landmark estimators based on subsampling approach, and the estimator based on weighted cumulative hazard estimator. The package also provides nonparametric estimator conditional to a given continuous covariate. All these methods have been submitted to be published.

License GPL-3

Depends R (>= 3.5.0)

Imports survival, KernSmooth, graphics, stats, utils, methods

RoxygenNote 7.2.1

Encoding UTF-8

LazyLoad yes

LazyData yes

NeedsCompilation yes

Suggests rmarkdown, knitr

VignetteBuilder knitr

Author Gustavo Soutinho [aut, cre]
 (<<https://orcid.org/0000-0002-0559-1327>>), Luis Meira-Machado
 [aut] (<<https://orcid.org/0000-0002-8577-7665>>), Artur Araujo
 [ctb] (<<https://orcid.org/0000-0003-1419-4208>>), Ana Moreira
 [ctb]

Maintainer Gustavo Soutinho <gustavosoutinho@sapo.pt>

Date/Publication 2023-08-09 10:30:05 UTC

Repository <https://gsoutinho.r-universe.dev>

RemoteUrl <https://github.com/cran/survivalREC>

RemoteRef HEAD

RemoteSha 6dfe78177d23090e5605b14505fabce43628575b

Contents

b3	3
b3size	3
b3state	4
b3state2	5
b4	5
b4state	6
Beran	7
bladder3	8
bladder3state	9
bladder4	9
bladder4state	10
bladder5	11
bladder5state	11
IPCWdf	12
KM	14
KMW	15
KMW3df	16
KMWdf	17
LDM3df	18
LDMdf	19
LIN3df	20
LINdf	21
multidf	22
NWW	23
plot.multidf	24
WCH3df	26
WCHdf	27

Index **29**

b3	<i>b3</i>
----	-----------

Description

b3 data set.

Usage

```
data("b3")
```

Format

A data frame with 85 observations on the following 4 variables. Below a brief description is given for each of these variables.

time1 First time or censoring time.

time The total time of the process.

event1 Indicator of the first time; 0 if the first time is censored and 1 otherwise.

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

Examples

```
data(b3)
head(b3)
```

b3size	<i>b3size</i>
--------	---------------

Description

b3size data set.

Usage

```
data("b3size")
```

Format

A data frame with 85 observations on the following 5 variables. Below a brief description is given for each of these variables.

time1 First time or censoring time.

time The total time of the process

event1 Indicator of the first time; 0 if the first time is censored and 1 otherwise.

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

size Values of covariate size.

Examples

```
data(b3size)
```

b3state	<i>b3state</i>
---------	----------------

Description

b3state data set.

Usage

```
data("b3state")
```

Format

A data frame with 85 observations on the following 4 variables. Below a brief description is given for each of these variables.

time1 First time or censoring time.

time The total time of the process.

event1 Indicator of the first time; 0 if the first time is censored and 1 otherwise.

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

Examples

```
data(b3state)
str(b3state)
```

b3state2

b3state2

Description

b3state2 data set

Usage

```
data("b3state2")
```

Format

A data frame with 85 observations on the following 5 variables. Below a brief description is given for each of these variables.

time1 First time or censoring time.

time The total time of the process

event1 Indicator of the first time; 0 if the first time is censored and 1 otherwise.

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

size Values of covariate size.

Examples

```
data(b3state2)  
str(b3state2)
```

b4

b4

Description

b4 data set.

Usage

```
data("b4")
```

Format

A data frame with 85 observations on the following 6 variables. Below a brief description is given for each of these variables.

time1 First time or censoring time.

time2 Second time.

time The total time of the process

event1 Indicator of the first time; 0 if the first time is censored and 1 otherwise.

event2 Indicator of the second time; 0 if the first time is censored and 1 otherwise.

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

Examples

```
data(b4)
head(b4)
```

b4state	<i>b4state</i>
---------	----------------

Description

b4state data set.

Usage

```
data("b4state")
```

Format

A data frame with 85 observations on the following 6 variables. Below a brief description is given for each of these variables.

time1 First time or censoring time.

time2 Second time.

time The total time of the process

event1 Indicator of the first time; 0 if the first time is censored and 1 otherwise.

event2 Indicator of the second time; 0 if the first time is censored and 1 otherwise.

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

Examples

```
data(b4state)
## maybe str(b4state) ; plot(b4state) ...
```

Beran	<i>Estimation of the conditional distribution function of the response, given the covariate under random censoring.</i>
-------	---

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"

Value

Vector with the estimation of the conditional distribution function of the response, given the covariate under random censoring.

Author(s)

Gustavo Soutinho and Luis Meira-Machado

References

R. Beran. Nonparametric regression with randomly censored survival data. Technical report, University of California, Berkeley, 1981.

Examples

```
data("bladder4state")
b3state<-multidf(gap1=bladder4state$y1, event1=bladder4state$d1,
                gap2=bladder4state$y2, status=bladder4state$d2,
                size=bladder4state$size)

head(b3state[[1]])

##P(T>y|size=3)
library(KernSmooth)

obj0 <- b3state[[1]]

h <- dpik(obj0$size)
Beran(time = obj0$time, status = obj0$status, covariate =obj0$size, x = 3,
y = 50, bw = h)

##P(T<=y|size=3)
Beran(time = obj0$time, status = obj0$status, covariate =obj0$size, x = 3,
y = 50, bw = h,
lower.tail = TRUE)
```

bladder3

bladder3

Description

bladder3-description

Usage

```
data("bladder3")
```

Format

A data frame with 85 observations on the following 6 variables.

t1 First time or censoring time.

d1 Indicator of the first time; 0 if the first time is censored and 1 otherwise.

t2 The total time of the process

d2 Censoring indicator of the survival time of the process; 0 if the total time is censored and 1 otherwise.

rx Values of covariate rx.

size Values of covariate size.

Examples

```
data(bladder3)
str(bladder3)
```

bladder3state	<i>bladder3state</i>
---------------	----------------------

Description

bladder3state data set.

Usage

```
data("bladder3state")
```

Format

A data frame with 85 observations on the following 7 variables.

id Identification number.

y1 First gap time.

d1 Indicator of the first gap time; 0 if the first time is censored and 1 otherwise.

y2 Second gap time.

d2 Censoring indicator of the second gap time; 0 if the total time is censored and 1 otherwise.

rx Values of covariate rx.

size Values of covariate size.

Examples

```
data(bladder3state)
str(bladder3state)
```

bladder4	<i>bladder4</i>
----------	-----------------

Description

bladder4 data set.

Usage

```
data("bladder4")
```

Format

A data frame with 85 observations on the following 8 variables.

- t1 First time or censoring time.
- d1 Indicator of the first time; 0 if the first time is censored and 1 otherwise.
- t2 Second time or censoring time.
- d2 Indicator of the second time; 0 if the first time is censored and 1 otherwise.
- t3 The total time of the process.
- d3 Censoring indicator of the survival time of the process; 0 if the total time is censored and 1 otherwise.
- rx Values of covariate rx.
- size Values of covariate size.

Examples

```
data(bladder4)
```

bladder4state	<i>bladder4state</i>
---------------	----------------------

Description

bladder4state data set.

Usage

```
data("bladder4state")
```

Format

A data frame with 85 observations on the following 9 variables.

- id Identification number.
- y1 First gap time.
- d1 Indicator of the first gap time; 0 if the first time is censored and 1 otherwise.
- y2 Second gap time.
- d2 Censoring indicator of the second gap time; 0 if the total time is censored and 1 otherwise.
- y3 Third gap time.
- d3 Censoring indicator of the third gap time; 0 if the total time is censored and 1 otherwise.
- rx Values of covariate rx.
- size Values of covariate size.

Examples

```
data(bladder4state)
```

bladder5	<i>bladder5</i>
----------	-----------------

Description

bladder5 data set.

Usage

```
data("bladder5")
```

Format

A data frame with 85 observations on the following 10 variables.

t1 First time or censoring time.

d1 Indicator of the first time; 0 if the first time is censored and 1 otherwise.

t2 Second time or censoring time.

d2 Indicator of the second time; 0 if the first time is censored and 1 otherwise.

t3 Third time or censoring time.

d3 Indicator of the third time; 0 if the first time is censored and 1 otherwise.

t4 The total time of the process

d4 Censoring indicator of the survival time of the process; 0 if the total time is censored and 1 otherwise.

rx Values of covariate rx.

size Values of covariate size.

Examples

```
data(bladder5)
```

bladder5state	<i>bladder5state</i>
---------------	----------------------

Description

bladder5state data set.

Usage

```
data("bladder5state")
```

Format

A data frame with 85 observations on the following 11 variables.

id Identification number.
 y1 First gap time.
 d1 Indicator of the first gap time; 0 if the first time is censored and 1 otherwise.
 y2 Second gap time.
 d2 Censoring indicator of the second gap time; 0 if the total time is censored and 1 otherwise.
 y3 Third gap time.
 d3 Censoring indicator of the third gap time; 0 if the total time is censored and 1 otherwise.
 y4 Fourth gap time.
 d4 Censoring indicator of the fourth gap time; 0 if the total time is censored and 1 otherwise.
 rx Values of covariate rx.
 size Values of covariate size.

Examples

```
data(bladder5state)
```

IPCWdf	<i>Inverse probability of censoring weighting estimator for the bivariate distribution function.</i>
--------	--

Description

Provides estimates for the bivariate distribution function based on the Inverse Probability of Censoring Weighting estimator (IPCW).

Usage

```
IPCWdf(object, x, y, covariate, cov.value, bw, window = "gaussian")
```

Arguments

object	An object of class <code>multidf</code> .
x	The first time for obtaining estimates for the bivariate distribution function.
y	The second time for obtaining estimates for the bivariate distribution function.
covariate	Name of the quantitative covariate.
cov.value	The value of the quantitative covariate.
bw	A single numeric value to compute a kernel density bandwidth. Use <code>"dpik"</code> for the KernSmooth package based selector or <code>"np"</code> for the <code>'npudensbw'</code> function of the np package.
window	A character string specifying the desired kernel. See details below for possible options. Defaults to <code>"gaussian"</code> where the gaussian density kernel will be used.

Value

Vector with the IPWC estimates for the bivariate distribution function.

Author(s)

Gustavo Soutinho and Luis Meira-Machado.

Gustavo Soutinho and Luis Meira-Machado

References

de Una-Alvarez, J. and Meira-Machado, L. (2008). A simple estimator of the bivariate distribution function for censored gap times, *Statistics and Probability Letters* 78, 2440-2445.

See Also

[KMWdf](#), [LDMdf](#), [LINDf](#) and [WCHdf](#).

Examples

```
data("bladder4state")

b3state<-multidf(gap1=bladder4state$y1, event1=bladder4state$d1,
                gap2=bladder4state$y2, status=bladder4state$d2,
                size=bladder4state$size)

b3size<-multidf(gap1=bladder3$t1, event1=bladder3$d1,
                gap2=bladder3$t2-bladder3$t1, status=bladder4state$d2,
                size=bladder3$size)

library(KernSmooth)

IPCWdf(object=b3state, x=13, y=15, covariate="size", cov.value=3,
        window = "gaussian")

IPCWdf(object=b3state, x=13, y=15, covariate="size", bw=2, cov.value=3,
        window = "gaussian")

IPCWdf(object=b3size, x=13, y=15, covariate="size", cov.value=3,
        window = "gaussian")

IPCWdf(object=b3size, x=13, y=15, covariate="size", bw=2, cov.value=3,
        window = "gaussian")
```

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

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

Value

Vector with Kaplan-Meier estimate of survival.

Author(s)

Gustavo Soutinho and Luis Meira-Machado

References

E. Kaplan and P. Meier. Nonparametric estimation from incomplete observations. *Journal of the American Statistical Association*, 53:457-481, 1958.

Examples

```
require(survival)
data("bladder4state")

obj2<- multidf(gap1=bladder4state$y1, event1=bladder4state$d1,
              gap2=bladder4state$y2, status=bladder4state$d2,
              size=bladder4state$size)

obj2<-obj2[[1]]
KM(time = obj2$time, status = obj2$status, t = 20)
fit <- survfit(Surv(obj2$time, obj2$status) ~ 1, data = obj2)
summary(fit, time = 20)$surv
```

KMW*Kaplan-Meier weights*

Description

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

Usage

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

Arguments

<code>time</code>	Survival time of the process.
<code>status</code>	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)

Gustavo Soutinho and Luis Meira-Machado

References

E. Kaplan and P. Meier. Nonparametric estimation from incomplete observations. *Journal of the American Statistical Association*, 53:457-481, 1958.

Examples

```
data("bladder4state")

obj<- multidf(gap1=bladder4state$y1, event1=bladder4state$d1,
             gap2=bladder4state$y2, status=bladder4state$d2,
             size=bladder4state$size)

obj2<-obj[[1]]
kmw <- KMW(time = obj2$time, status = obj2$status)
require(survival)
bladder.surv <- survfit(Surv(time, status) ~ 1, obj2)
times <- summary(bladder.surv)$time
surv <- summary(bladder.surv)$surv
nevent <- summary(bladder.surv)$n.event
p <- match(obj2$time, times)
kmw2 <- -diff(c(1, surv))/nevent
kmw2 <- kmw2[p]*obj2$status
kmw2[is.na(kmw2)] <- 0
```

```
all.equal(kmw, kmw2)
```

KMW3df	<i>Kaplan-Meier Weighted estimator for three gap times distribution function.</i>
--------	---

Description

Provides estimates for three gap times distribution function based on Kaplan-Meier Weights (KMW).

Usage

```
KMW3df(object, x, y, z)
```

Arguments

object	An object of class <code>multidf</code> .
x	The first time for obtaining estimates for the trivariate distribution function.
y	The second time for obtaining estimates for the trivariate distribution function.
z	The third time for obtaining estimates for the trivariate distribution function.

Value

Vector with the Kaplan-Meier Weighted estimates for three gapes times distribution function.

Author(s)

Gustavo Soutinho and Luis Meira-Machado

References

de Una-Alvarez J, Meira Machado LF (2008). "A Simple Estimator of the Bivariate Distribution Function for Censored Gap Times", *Statistical and Probability Letters*, 78, 2440-2445.

Davison, A.C. and Hinkley, D.V. (1997) "Bootstrap Methods and Their Application", Chapter 5. Cambridge University Press.

See Also

[LDM3df](#), [LIN3df](#) and [WCH3df](#).

Examples

```

data("bladder5state")
b4state<-multidf(gap1=bladder5state$y1, event1=bladder4state$d1,
                gap2=bladder5state$y2, event2=bladder4state$d2,
                gap3=bladder5state$y3, status=bladder4state$d3)

head(b4state)[[1]]

KMW3df(b4state, x=13, y=20, z=40)

b4<-multidf(gap1=bladder4$t1, event1=bladder4$d1,
            gap2=bladder4$t2-bladder4$t1, event2=bladder4$d2,
            gap3=bladder4$t3-bladder4$t2, status=bladder4state$d3)

KMW3df(b4, x=13, y=20, z=40)

```

KMWdf	<i>Kaplan-Meier Weighted estimator for the bivariate distribution function.</i>
-------	---

Description

Provides estimates for the bivariate distribution function based on Kaplan-Meier Weights (KMW).

Usage

```
KMWdf(object, x, y)
```

Arguments

object	An object of class multidf.
x	The first time for obtaining estimates for the bivariate distribution function.
y	The second time for obtaining estimates for the bivariate distribution function.

Value

Vector with the Kaplan-Meier weights estimates for the bivariate distribution function.

Author(s)

Gustavo Soutinho and Luis Meira-Machado

References

de Una-Alvarez J, Meira Machado LF (2008). "A Simple Estimator of the Bivariate Distribution Function for Censored Gap Times", *Statistical and Probability Letters*, 78, 2440-2445.

Davison, A.C. and Hinkley, D.V. (1997) "Bootstrap Methods and Their Application", Chapter 5. Cambridge University Press.

See Also

[IPCwdf](#), [LDMdf](#), [LINdf](#) and [WCHdf](#).

Examples

```
data("bladder4state")

b3state<-multidf(gap1=bladder4state$y1, event1=bladder4state$d1,
                gap2=bladder4state$y2, status=bladder4state$d2,
                size=bladder4state$size)

KMWdf(b3state, x=13, y=20)
```

LDM3df

Landmark estimator for three gap times distribution function.

Description

Provides estimates for three gap times distribution function based on landmarking. The extension of the landmark estimator (LDM) to three gap times is a consequence of Bayes' theorem.

Usage

```
LDM3df(object, x, y, z)
```

Arguments

object	An object of class multidf.
x	The first time for obtaining estimates for the trivariate distribution function.
y	The second time for obtaining estimates for the trivariate distribution function.
z	The third time for obtaining estimates for the trivariate distribution function.

Value

Vector with the Landmark estimates for three gap times distribution function.

Author(s)

Gustavo Soutinho and Luis Meira-Machado

References

van Houwelingen, H.C. (2007). Dynamic prediction by landmarking in event history analysis, *Scandinavian Journal of Statistics*, 34, 70-85.

Kaplan, E. and Meier, P. (1958). Nonparametric Estimation from Incomplete Observations, *Journal of the American Statistical Association* 53(282), 457-481.

See Also

[LDM3df](#), [LIN3df](#) and [WCH3df](#).

Examples

```
data("bladder5state")
b4state<-multidf(gap1=bladder5state$y1, event1=bladder4state$d1,
                gap2=bladder5state$y2, event2=bladder4state$d2,
                gap3=bladder5state$y3, status=bladder4state$d3)

head(b4state)[[1]]

LDM3df(b4state, x=13, y=20, z=40)

b4<-multidf(gap1=bladder4$t1, event1=bladder4$d1,
            gap2=bladder4$t2-bladder4$t1, event2=bladder4$d2,
            gap3=bladder4$t3-bladder4$t2, status=bladder4state$d3)

LDM3df(b4, x=13, y=20, z=40)
```

LDMdf

Landmark estimator for the bivariate distribution function

Description

Provides estimates for the bivariate distribution function based on Bayes' theorem and Kaplan-Meier survival function. This approach is also named as landmarking.

Usage

```
LDMdf(object, x, y)
```

Arguments

object	An object of class multidf.
x	The first time for obtaining estimates for the bivariate distribution function.
y	The second time for obtaining estimates for the bivariate distribution function.

Value

Vector with the Landmark estimates for the bivariate distribution function.

Author(s)

Gustavo Soutinho and Luis Meira-Machado

References

van Houwelingen, H.C. (2007). Dynamic prediction by landmarking in event history analysis, *Scandinavian Journal of Statistics*, 34, 70-85.

Kaplan, E. and Meier, P. (1958). Nonparametric Estimation from Incomplete Observations, *Journal of the American Statistical Association* 53(282), 457-481.

See Also

[IPCwdf](#), [KMwdf](#), [LINDf](#) and [WCHdf](#).

Examples

```
b3state<-multidf(gap1=bladder4state$y1, event1=bladder4state$d1,
                gap2=bladder4state$y2, status=bladder4state$d2,
                size=bladder4state$size)
```

```
LDMdf(b3state, x=13, y=20)
```

LIN3df

Lin's estimator for three gap times distribution function.

Description

Provides estimates for three gap times distribution function based on the extension the Lin's estimator.

Usage

```
LIN3df(object, x, y, z)
```

Arguments

object	An object of class <code>multidf</code> .
x	The first time for obtaining estimates for the trivariate distribution function.
y	The second time for obtaining estimates for the trivariate distribution function.
z	The third time for obtaining estimates for the trivariate distribution function.

Value

Vector with the Lin's estimates for three gapes times distribution function.

Author(s)

Gustavo Soutinho and Luis Meira-Machado

References

Lin, D. Y., Sun, W. and Ying, Z. (1999). Nonparametric estimation of the gap time distributions for serial events with censored data, *Biometrika* 86, 59-70.

See Also

[LDM3df](#), [KMW3df](#) and [WCH3df](#).

Examples

```
data("bladder5state")
b4state<-multidf(gap1=bladder5state$y1, event1=bladder4state$d1,
                gap2=bladder5state$y2, event2=bladder4state$d2,
                gap3=bladder5state$y3, status=bladder4state$d3)

head(b4state)[[1]]

LIN3df(b4state, x=13, y=20, z=40)

b4<-multidf(gap1=bladder4$t1, event1=bladder4$d1,
            gap2=bladder4$t2-bladder4$t1, event2=bladder4$d2,
            gap3=bladder4$t3-bladder4$t2, status=bladder4state$d3)

LIN3df(b4, x=13, y=20, z=40)
```

 LINdf

Lin's estimator for the bivariate distribution function.

Description

Provides estimates for the bivariate distribution function based on the extension the Kaplan-Meier estimator of the distribution function for the first event time and the Inverse Probability of Censoring Weights for the second time.

Usage

```
LINdf(object, x, y)
```

Arguments

object	An object of class <code>multidf</code> .
x	The first time for obtaining estimates for the bivariate distribution function.
y	The second time for obtaining estimates for the bivariate distribution function.

Value

Vector with the Lin's estimates for the bivariate distribution function.

Author(s)

Gustavo Soutinho and Luis Meira-Machado

References

Lin, D. Y., Sun, W. and Ying, Z. (1999). Nonparametric estimation of the gap time distributions for serial events with censored data, *Biometrika* 86, 59-70.

See Also

[IPCwdf](#), [LDMdf](#), [KMwdf](#) and [WCHdf](#).

Examples

```
b3state<-multidf(gap1=bladder4state$y1, event1=bladder4state$d1,
                gap2=bladder4state$y2, status=bladder4state$d2,
                size=bladder4state$size)
```

```
LINdf(b3, x=13, y=20)
```

multidf

Create a multidf object

Description

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

Usage

```
multidf(gap1, gap2, gap3=NULL, event1, status, event2=NULL, ...)
```

Arguments

gap1	First gap time.
gap2	Second gap time.
gap3	Third gap time. By default is NULL.
event1	Indicator of the first time; 0 if the first time is censored and 1 otherwise.
status	Censoring indicator of the survival time of the process; 0 if the total time is censored and 1 otherwise. For instance, for three gap times, status is given by the indicator of the third time.
event2	Indicator of the second time; 0 if the first time is censored and 1 otherwise. By default is NULL.
...	Other options. Additional arguments, such as covariates, can also be included in the data set.

Details

Arguments in this function must be introduced in the following order: `gap1`, `event1`, `gap2` and `status`, where `gap1` and `gap2` are ordered event times and `event1` and `status` their corresponding indicator statuses. Other arguments can be also added. These should consider intermediate times and corresponding censoring indicators or covariates.

Value

An object of class "multidf". "multidf" objects are implemented as a single data frame.

Author(s)

Gustavo Soutinho and Luis Meira-Machado

Examples

```
library(survivalREC)
data("bladder4state")

b3state<-multidf(gap1=bladder4state$y1, event1=bladder4state$d1,
                gap2=bladder4state$y2, status=bladder4state$d2,
                size=bladder4state$size)

head(b3state[[1]])

class(b3state)

b4state<-multidf(gap1=bladder4state$y1, event1=bladder4state$d1,
                gap2=bladder4state$y2, event2=bladder4state$d2,
                gap3=bladder4state$y3, status=bladder4state$d3,
                size=bladder4state$size)

head(b4state[[1]])
```

NWW

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".

Value

A vector with Nadaraya-Watson weights.

Author(s)

Gustavo Soutinho and Luis Meira-Machado

Examples

```
b3state<-multidf(gap1=bladder4state$y1, event1=bladder4state$d1,
                gap2=bladder4state$y2, status=bladder4state$d2,
                size=bladder4state$size)

obj0 <- b3state[[1]]

NWW(covariate = obj0$size, x=3, kernel = "gaussian", bw = 3)
```

plot.multidf

Plot methods for a multidf object

Description

Provides the plots for the bivariate distribution function and marginal distribution of the second time.

Usage

```
## S3 method for class 'multidf'
plot(x, t1, method = "KMW", type = "s", ...)
```


Arguments

x	An object of class multidf.
t1	Value of the first gap time.
method	A character string specifying which estimator to fit. Possible values are "KMW", "LIN", "WCH" and "LANDMARK".
type	The type of plot that should be drawn. See details par for possible options. Defaults to "s" for the draw be stair steps.
...	Other options.

Value

No value is returned.

Author(s)

Gustavo Soutinho and Luis Meira-Machado

References

de Una-Alvarez, J. and Meira-Machado, L. (2008). A simple estimator of the bivariate distribution function for censored gap times, *Statistics and Probability Letters* 78, 2440-2445.

Davison, A.C. and Hinkley, D.V. (1997) "Bootstrap Methods and Their Application", Chapter 5. Cambridge University Press.

van Houwelingen, H.C. (2007). Dynamic prediction by landmarking in event history analysis, *Scandinavian Journal of Statistics*, 34, 70-85. Kaplan, E. and Meier, P. (1958). Nonparametric Estimation from Incomplete Observations, *Journal of the American Statistical Association* 53(282), 457-481.

See Also

[KMWdf](#), [LDMdf](#), [LINdf](#) and [WCHdf](#).

Examples

```
data("bladder4state")

b3state<-multidf(gap1=bladder4state$y1, event1=bladder4state$d1,
                gap2=bladder4state$y2, status=bladder4state$d2,
                size=bladder4state$size)

head(b3state[[1]])

KMWdf(b3state,x=13,y=20)
LDMdf(b3state,x=13,y=20)
LINdf(b3state,x=13,y=20)
WCHdf(b3state,x=13,y=20)

plot(x=b3state, t1=3, method="KMW", type = "s")
```

```
plot(x=b3state, t1=3, method="LIN", type = "s")
plot(x=b3state, t1=3, method="WCH", type = "s")
plot(x=b3state, t1=3, method="LANDMARK", type = "s")
```

WCH3df	<i>Weighted cumulative hazard estimator for three gap times distribution function.</i>
--------	--

Description

Provides estimates for three gap times distribution function based on Weighted cumulative hazard estimator (WCH).

Usage

```
WCH3df(object, x, y, z)
```

Arguments

object	An object of class multidf.
x	The first time for obtaining estimates for the three gap times distribution function.
y	The second time for obtaining estimates for the three gap times distribution function.
z	The third time for obtaining estimates for the three gap times distribution function.

Value

Vector with the Weighted cumulative hazard estimates for three gap times distribution function.

Author(s)

Gustavo Soutinho and Luis Meira-Machado

References

Wang, M.C. and Wells, M.T. (1998). Nonparametric Estimation of successive duration times under dependent censoring, *Biometrika* 85, 561-572.

See Also

[KMW3df](#), [LIN3df](#) and [LDM3df](#).

Examples

```

data("bladder5state")

b4state<-multidf(gap1=bladder5state$y1, event1=bladder4state$d1,
                gap2=bladder5state$y2, event2=bladder4state$d2,
                gap3=bladder5state$y3, status=bladder4state$d3)

head(b4state)[[1]]

WCH3df(b4state, x=13, y=20, z=40)

b4<-multidf(gap1=bladder4$t1, event1=bladder4$d1,
            gap2=bladder4$t2-bladder4$t1, event2=bladder4$d2,
            gap3=bladder4$t3-bladder4$t2, status=bladder4state$d3)

WCH3df(b4, x=13, y=20, z=40)

```

WCHdf	<i>Weighted cumulative hazard estimator for the bivariate distribution function</i>
-------	---

Description

Provides estimates for the bivariate distribution function based on Weighted cumulative hazard estimator (WCH).

Usage

```
WCHdf(object, x, y)
```

Arguments

object	An object of class multidf.
x	The first time for obtaining estimates for the bivariate distribution function.
y	The second time for obtaining estimates for the bivariate distribution function.

Value

Vector with the Weighted cumulative hazard estimates for the bivariate distribution function.

Author(s)

Gustavo Soutinho and Luis Meira-Machado

References

Wang, M.C. and Wells, M.T. (1998). Nonparametric Estimation of successive duration times under dependent censoring, *Biometrika* 85, 561-572.

See Also

[IPCwdf](#), [KMwdf](#), [LINDf](#) and [LDMdf](#).

Examples

```
data("bladder3")

b3<-multidf(gap1=bladder3$t1, event1=bladder3$d1,
            gap2=bladder3$t2-bladder3$t1, status=bladder4state$d2)

head(b3[[1]])
WCHdf(b3, x=13, y=20)
```

Index

* datasets

- b3, [3](#)
- b3size, [3](#)
- b3state, [4](#)
- b3state2, [5](#)
- b4, [5](#)
- b4state, [6](#)
- bladder3, [8](#)
- bladder3state, [9](#)
- bladder4, [9](#)
- bladder4state, [10](#)
- bladder5, [11](#)
- bladder5state, [11](#)

- b3, [3](#)
- b3size, [3](#)
- b3state, [4](#)
- b3state2, [5](#)
- b4, [5](#)
- b4state, [6](#)
- Beran, [7](#)
- bladder3, [8](#)
- bladder3state, [9](#)
- bladder4, [9](#)
- bladder4state, [10](#)
- bladder5, [11](#)
- bladder5state, [11](#)

IPCwdf, [12](#), [18](#), [20](#), [22](#), [28](#)

- KM, [14](#)
- KMW, [15](#)
- KMW3df, [16](#), [21](#), [26](#)
- KMWdf, [13](#), [17](#), [20](#), [22](#), [25](#), [28](#)

- LDM3df, [16](#), [18](#), [19](#), [21](#), [26](#)
- LDMdf, [13](#), [18](#), [19](#), [22](#), [25](#), [28](#)
- LIN3df, [16](#), [19](#), [20](#), [26](#)
- LINdf, [13](#), [18](#), [20](#), [21](#), [25](#), [28](#)

multidf, [22](#)

NWW, [23](#)

- par, [25](#)
- plot.multidf, [24](#)

WCH3df, [16](#), [19](#), [21](#), [26](#)

WCHdf, [13](#), [18](#), [20](#), [22](#), [25](#), [27](#)