Package: NPHazardRate (via r-universe)

October 21, 2024

Encoding UTF-8

Type Package

Title Nonparametric Hazard Rate Estimation

Version 0.1

Date 2018-10-13

Author Dimitrios Bagkavos [aut, cre]

Maintainer Dimitrios Bagkavos <dimitrios.bagkavos@gmail.com>

Depends R (>= 3.5.0)

Imports stats, survival

Description Provides functions and examples for histogram, kernel
(classical, variable bandwidth and transformations based),
discrete and semiparametric hazard rate estimators.

License GPL (>= 2)

NeedsCompilation no

RoxygenNote 6.1.0

Repository CRAN

Date/Publication 2018-11-02 18:10:18 UTC

Contents

`cvfunction` *Cross Validation for Histogram Hazard Rate Estimator*

Description

Implements the cross validation function for determining the optimal number of bins for the histogram hazard rate estimator of Patil and Bagkavos (2012). It is used as input in `HazardHistogram`.

Usage

```
cvfunction(h, xin, xout, cens)
```

Arguments

`h`	Target number of bins.
`xin`	A vector of data points. Missing values not allowed.
`xout`	A vector of grid points at which the histogram will be calculated.
`cens`	A vector of 1s and zeros, 1's indicate uncensored observations, 0's correspond to censored obs.

Details

The least square cross validation criterion, defined in (12), Patil and Bagkavos (2012) is

$$CV(h) = \frac{1}{h}\sum_k \big\{(2f_k^0 - f_k^{0^2})[\bar{F}_k(\bar{F}_k+1)]^{-1} - f_k^{0^2}[\bar{F}_k(\bar{F}_k+1)^2]^{-1}\big\}.$$

Optimization of the criterion is done through a nonlinear optimization function such as `nlminb` as illustrated also in the example of `HazardHistogram`.

Value

Returns the optimal number of bins.

References

Patil and Bagkavos (2012), Histogram for hazard rate estimation, pp. 286-301, Sankhya, B.

See Also

`HazardHistogram`

`DefVarBandRule` *Default adaptive bandwidth rule*

Description

Implements an adaptive variable bandwidth hazard rate rule for use with the `VarBandHazEst` based on the Weibull distribution, with parameters estimated by maximum likelihood

Usage

```
DefVarBandRule(xin, cens)
```

Arguments

`xin`	A vector of data points. Missing values not allowed.
`cens`	A vector of censoring indicators: 1's indicate uncensored observations, 0's correspond to censored obs.

Details

The adaptive AMISE optimal bandwidth for the variable bandwidth hazard rate estimator `VarBandHazEst` is given by

$$h_2 = \left[\frac{R(K) M_2}{8 n \mu_4^2(K) R(g)} \right]^{1/14}$$

where

$$M_2 = \int \frac{\lambda^{3/2}(x)}{1 - F(x)}\, dx$$

and

$$g(x) = \frac{1}{24 \lambda(x)^5} \left(24 \lambda'(x)^4 - 36 \lambda'(x)^2 \lambda''(x)^2 \lambda(x) + 6 \lambda''(x)^2 \lambda^2(x) + 8 \lambda'(x) \lambda'''(x) \lambda^2(x) - \lambda^{(4)}(x) \lambda^3(x) \right)$$

Value

the value of the adaptive bandwidth

References

Bagkavos and Patil (2009), Variable Bandwidths for Nonparametric Hazard Rate Estimation, Communications in Statistics - Theory and Methods, 38:7, 1055-1078

See Also

HazardRateEst, TransHazRateEst, PlugInBand

Examples

```
library(survival)
x<-seq(0, 5,length=100) #design points where the estimate will be calculated

SampleSize <- 100

ti<- rweibull(SampleSize, .6, 1)#draw a random sample from the actual distribution
ui<-rexp(SampleSize, .05)       #draw a random sample from the censoring distribution
cat("\n AMOUNT OF CENSORING: ", length(which(ti>ui))/length(ti)*100, "\n")
x1<-pmin(ti,ui)                 #this is the observed sample
cen<-rep.int(1, SampleSize)     #censoring indicators
cen[which(ti>ui)]<-0            #censored values correspond to zero

h2<-DefVarBandRule(ti, cen)     #Deafult Band. Rule - Weibull Reference
```

DiscretizeData   *Discretize the available data set*

Description

Defines equispaced disjoint intervals based on the range of the sample and calculates empirical hazard rate estimates at each interval center

Usage

```
DiscretizeData(xin, xout)
```

Arguments

xin	A vector of input values
xout	Grid points where the function will be evaluated

Details

The function defines the subinterval length $\Delta = (0.8 \max(X_i) - \min(X_i))/N$ where N is the sample size. Then at each bin (subinterval) center, the empirical hazard rate estimate is calculated by

$$c_i = \frac{f_i}{\Delta(N - F_i + 1)}$$

where f_i is the frequency of observations in the ith bin and $F_i = \sum_{j \le i} f_j$ is the empirical cummulative distribution estimate.

Value

A vector with the values of the function at the designated points xout or the random numbers drawn.

Examples

```
x<-seq(0, 5,length=100) #design points where the estimate will be calculated
SampleSize<-100 #amount of data to be generated
ti<- rweibull(SampleSize, .6, 1) # draw a random sample
ui<-rexp(SampleSize, .2)         # censoring sample
cat("\n AMOUNT OF CENSORING: ", length(which(ti>ui))/length(ti)*100, "\n")
x1<-pmin(ti,ui)                  # observed data
cen<-rep.int(1, SampleSize)      # initialize censoring indicators
cen[which(ti>ui)]<-0             # 0's correspond to censored indicators

a.use<-DiscretizeData(ti, x)     # discretize the data
BinCenters<-a.use$BinCenters     # get the data centers
ci<-a.use$ci                     # get empircal hazard rate estimates
Delta=a.use$Delta                # Binning range
```

`HazardHistogram` *Histogram Hazard Rate Estimator*

Description

Implements the histogram hazard rate estimator of Patil and Bagkavos (2012)

Usage

```
HazardHistogram(xin, xout, cens, bin)
```

Arguments

`xin`	A vector of data points. Missing values not allowed.
`xout`	A vector of grid points at which the histogram will be calculated.
`cens`	A vector of 1s and zeros, 1's indicate uncensored observations, 0's correspond to censored obs.
`bin`	Number of bins to use in construction of the histogram.

Details

The histogram hazard rate estimator is defined in (1), Patil and Bagkavos (2012) by

$$\hat{\lambda}(x) = h_n^{-1} C_{i_{(x)}} = h_n^{-1} f^0_{i_{(x)}} (\bar{F}_{i_{(x)}} + 1)^{-1}.$$

Value

A vector with the values of the histogram estimate at each bin.

References

Patil and Bagkavos (2012), Histogram for hazard rate estimation, pp. 286-301, Sankhya, B.

Examples

```
SampleSize <-400
ti<-rweibull(SampleSize,0.5,0.8)
xout<-seq(0.02, 3.5, length=80)
true.hazard<-dweibull(xout,0.5, 0.8)/(1-pweibull(xout, 0.5, 0.8))
cen<-rep.int(1, SampleSize)
cen[sample(1:SampleSize, SampleSize/10)]<-0

band<-nlminb(start= 2, obj=cvfunction, control = list(iter.max = 100, x.tol = .001)
            ,xin=ti, xout= xout, cens = cen,  lower=.01, upper=max(xout))
bin<- 3.49 * sd(ti)^2 * SampleSize^(-1/3) /50 #Scott 1979 Biometrika default rule
bin<-unlist(band[1])
histest<- HazardHistogram(ti,xout, cen, bin+0.013 )

plot(xout, true.hazard, type="l")
lines(histest[,1], histest[,2], col=2, type="s")
barplot( histest[,2], rep(bin, times=length(histest[,2])))
lines(xout, true.hazard, type="l", lwd=2, col=2)
```

HazardRateEst — *Kernel Hazard Rate Estimation*

Description

Implements the (classical) kernel hazard rate estimator for right censored data defined in Tanner and Wong (1983).

Usage

```
HazardRateEst(xin, xout, kfun, h, ci)
```

Arguments

`xin`	A vector of data points. Missing values not allowed.
`xout`	A vector of grid points at which the estimate will be calculated.
`kfun`	Kernel function to use. Supported kernels: Epanechnikov, Biweight, Gaussian, Rectangular, Triangular, HigherOrder.
`h`	A scalar, the bandwidth to use in the estimate.
`ci`	A vector of censoring indicators: 1's indicate uncensored observations, 0's correspond to censored obs.

Details

The kernel hazard rate estimator of Tanner and Wong (1983) is given by

$$\hat{\lambda}(x;h) = \sum_{i=1}^{n} \frac{K_h(x - X_{(i)})\delta_{(i)}}{n - i + 1}$$

h is determined by a bandwidth rule such as PlugInBand. HazardRateEst is also used as a pilot estimate in the implementation of both the variable bandwidth estimate VarBandHazEst and the transformed hazard rate estimate TransHazRateEst.

Value

A vector with the hazard rate estimates at the designated points xout.

References

Tanner and Wong (1983), The Estimation Of The Hazard Function From Randomly Censored Data By The Kernel Method, Annals of Statistics, 3, pp. 989-993.

See Also

VarBandHazEst, TransHazRateEst, PlugInBand

Examples

```
x<-seq(0, 5,length=100) #design points where the estimate will be calculated
plot(x, HazardRate(x, "weibull", .6, 1),  type="l", xlab = "x",
                   ylab="Hazard rate") #plot true hazard rate function
SampleSize <- 100
ti<- rweibull(SampleSize, .6, 1)  #draw a random sample from the actual distribution
ui<-rexp(SampleSize, .2)  #draw a random sample from the censoring distribution
cat("\n AMOUNT OF CENSORING: ", length(which(ti>ui))/length(ti)*100, "\n")
x1<-pmin(ti,ui)             #this is the observed sample
cen<-rep.int(1, SampleSize) #censoring indicators
cen[which(ti>ui)]<-0        #censored values correspond to zero
huse<-PlugInBand(x1, x,   cen, Biweight)
arg2<-HazardRateEst(x1, x, Epanechnikov, huse, cen) #Calculate the estimate
lines(x, arg2, lty=2)       #draw the result on the graphics device.
```

HRSurv *Estimate of the constant in the optimal AMISE expression*

Description

Calculation of the integrand of the contant term in the AMISE plugin bandwidth rule implemented in PlugInBand.

Usage

```
HRSurv(x, xin, cens, h, kfun)
```

Arguments

xin	A vector of data points
x	The point at which the estimates should be calculated.
cens	Censoring Indicators.
h	bandwidth to use.
kfun	The kernel function to use.

Details

Calculates the term

$$\frac{\lambda_T(x)}{1-F(x)}\,dx$$

which is passed then as argument to the function NP.M.Estimate for numerical integtaion. Currrently the fraction is estimated by

$$\frac{\hat{\lambda}(x;b)}{1-\hat{F}(x)}$$

where $\hat{\lambda}(x;b)$ is implemented by HazardRateEst using bandwidth bw.nrd{xin}. For $1-\hat{F}(x)$ the Kaplan-Meier estimate KMest is used.

Value

A vector with the value of the fraction.

References

Hua, Patil and Bagkavos, An L_1 analysis of a kernel-based hazard rate estimator, Australian and New Zealand J. Statist., (60), 43-64, (2018).

See Also

PlugInBand, NP.M.Estimate

Examples

```
x<-seq(0, 5,length=100) #design points where the estimate will be calculated
SampleSize<-100 #amount of data to be generated
ti<- rweibull(SampleSize, .6, 1) # draw a random sample
ui<-rexp(SampleSize, .2)         # censoring sample
cat("\n AMOUNT OF CENSORING: ", length(which(ti>ui))/length(ti)*100, "\n")
x1<-pmin(ti,ui)                  # observed data
cen<-rep.int(1, SampleSize)      # initialize censoring indicators
cen[which(ti>ui)]<-0             # 0's correspond to censored indicators

HRSurv(x, x1, cen, bw.nrd(x1), Biweight)
```

iHazardRateEst *Kernel Integrated Hazard Rate Estimation*

Description

Implements the integrated kernel hazard rate estimator for right censored data, i.e. a kernel estimate of the cummulative hazard function.

Usage

```
iHazardRateEst(xin, xout, ikfun, h, ci)
```

Arguments

xin	A vector of data points. Missing values not allowed.
xout	A vector of grid points at which the estimates will be calculated.
ikfun	Integrated kernel function to use
h	A scalar, the bandwidth to use in the estimate.
ci	A vector of censoring indicators: 1's indicate uncensored observations, 0's correspond to censored obs.

Details

The function iHazardRateEst implements the cummulative hazard rate estimator $\hat{\Lambda}(x;h_1)$ given by

$$\hat{\Lambda}(x;h_1) = \sum_{i=1}^{n} \frac{k\left\{(x - X_{(i)})h_1^{-1}\right\}\delta_{(i)}}{n-i+1}$$

where

$$k(x) = \int_{-\infty}^{x} K(y)\,dy$$

Note that iHazardRateEst is used in the implementation of the transformed hazard rate estimate TransHazRateEst.

Value

A vector with the cummulative hazard rate estimates at the designated points xout.

References

Tanner and Wong (1983), The Estimation Of The Hazard Function From Randomly Censored Data By The Kernel Method, Annals of Statistics, 3, pp. 989-993.

See Also

VarBandHazEst, TransHazRateEst, PlugInBand

Examples

```
x<-seq(0, 5,length=100) #design points where the estimate will be calculated

SampleSize <- 100
ti<- rweibull(SampleSize, .6, 1)  #draw a random sample from the actual distribution
ui<-rexp(SampleSize, .2)  #draw a random sample from the censoring distribution
cat("\n AMOUNT OF CENSORING: ", length(which(ti>ui))/length(ti)*100, "\n")
x1<-pmin(ti,ui)             #this is the observed sample
cen<-rep.int(1, SampleSize) #censoring indicators
cen[which(ti>ui)]<-0        #censored values correspond to zero
huse<-PlugInBand(x1, x,   cen, Biweight)
arg2<-iHazardRateEst(x1, x, IntEpanechnikov, huse, cen) #Calculate the estimate
```

Kernels *Kernel functions*

Description

Implements various kernel functions, including boundary, integrated and discrete kernels for use in the definition of the nonparametric estimates

Usage

```
Biweight(x, ...)
Epanechnikov(x, ...)
Triangular(x, ...)
Gaussian(x, ...)
HigherOrder(x, ...)
Rectangular(x, ...)
IntBiweight(x)
IntEpanechnikov(x)
IntRectangular(x)
IntTriangular(x)
IntGaussian(x)
SDBiweight(x)
a0(x,h)
a1(x,h)
a2(x,h)
BoundaryBiweight(x, h)
b0(x,h)
b1(x,h)
b2(x,h)
BoundaryEpanechnikov(x, h)
Habbema(xin, x)
```

Arguments

x	A vector of data points where the kernel will be evaluated.
h	A scalar.
xin	Discrete data inputs especially for the Habbema discrete kernel.
...	Further arguments.

Details

Implements the Biweight, Second Derivative Biweight, Epanechnikov, Triangular, Guassian, Rectangular, the Boundary adjusted Biweight and Epanechnikov kernels. It also provides the kernel distribution functions for the Biweight, Epanechnikov, Rectangular, Triangular and Guassian kernels. Additionally it implements the discrete kernel Habbema.

Value

The value of the kernel at x

References

1. Bagkavos and Patil, Local Polynomial Fitting in Failure Rate Estimation, IEEE Transactions on Reliability, 57, (2008),
2. Bagkavos (2011), Annals of the Institute of Statistical Mathematics, 63(5), 1019-1046,

---

KMest — *Kaplan-Meier Estimate*

Description

Custom implementation of the Kaplan Meier estimate. The major difference with existing implementations is that the user can specify exactly the grid points where the estimate is calculated. The implementation corresponds to $1 - \hat{H}(x)$ of Hua, Patil and Bagkavos (2018), and is used mainly for estimation of the censoring distribution.

Usage

```
KMest(xin, cens, xout)
```

Arguments

xin	A vector of data points
xout	The point at which the estimates should be calculated.
cens	Censoring Indicators.

Details

Calculates the well known Kaplan-Meier estimate

$$1-\hat{H}(x)=1, 0 \le x \le X_{(1)}$$

or

$$1-\hat{H}(x)=\prod_{i=1}^{k-1}\left(\frac{n-i+1}{n-i+2}\right)^{1-\delta_{(i)}}, X_{(k-1)} < x \le X_{(k)}, k=2,\ldots,n$$

or

$$1-\hat{H}(x)=\prod_{i=1}^{n}\left(\frac{n-i+1}{n-i+2}\right)^{1-\delta_{(i)}}, X_{(n)} < x.$$

The implementation is mainly for estimating the censoring distribution of the available sample.

Value

A vector with the Kaplan-Meier estimate at xout.

References

Kaplan, E. L., and Paul Meier. Nonparametric Estimation from Incomplete Observations., J. of the American Statist. Association 53, (1958): 457-81.

Examples

```
x<-seq(0, 5,length=100) #design points where the estimate will be calculated
SampleSize<-100 #amount of data to be generated
ti<- rweibull(SampleSize, .6, 1) # draw a random sample
ui<-rexp(SampleSize, .2)         # censoring sample
cat("\n AMOUNT OF CENSORING: ", length(which(ti>ui))/length(ti)*100, "\n")
x1<-pmin(ti,ui)                  # observed data
cen<-rep.int(1, SampleSize)      # initialize censoring indicators
cen[which(ti>ui)]<-0             # 0's correspond to censored indicators

arg1<- KMest(x1, cen, x)
plot(x, arg1, type="l")
```

---

l1-l4, lw, lwF, gx *Weibull hazard rate functionals*

Description

Privides the various hazard rate function derivatives and related functionals with reference to the Weibull function

Usage

```
l1(x,p,l)
l2(x,p,l)
l3(x,p,l)
l4(x,p,l)
lw(x,p,l)
lwF(x,p,l)
gx(x,p,l)
```

Arguments

x	A vector of points at which the hazard rate function will be estimated.
p	MLE estimate of the shape parameter
l	MLE estimate of the scale parameter

Details

Implements the necessary functions for calculating the squared bias term of the variable bandwidth estimate.

Value

A vector with the values of the function at the designated points x.

References

Bagkavos and Patil (2009), Variable Bandwidths for Nonparametric Hazard Rate Estimation, Communications in Statistics - Theory and Methods, 38:7, 1055-1078

---

lambdahat *Discrete non parametric mle hazard rate estimator*

Description

Implementation of the purely nonparametric discrete hazard rate estimator `lambdahat` discussed among others in Patil and Bagkavos (2012). `lambdahat` is also used as the nonparametric component in the implementation of `SemiparamEst`.

Usage

```
lambdahat(xin, cens, xout)
```

Arguments

xin	A vector of data points. Missing values not allowed.
cens	Censoring indicators as a vector of 1s and zeros, 1's indicate uncensored observations, 0's correspond to censored obs.
xout	The grid points where the estimates will be calculated.

Details

The discrete - crude - hazard rate estimator (NPMLE) in Patil and Bagkavos (2012) is given by

$$\hat{\lambda}(t_k) = \frac{n_k^0}{m_k + 1}$$

Value

Returns a vector with the values of the hazard rate estimates at $x = xout$.

References

Patil and Bagkavos (2012), Semiparametric smoothing of discrete failure time data, Biometrical Journal, 54, (2012), 5–19.

See Also

SemiparamEst

Examples

```
options(echo=FALSE)
xin<-c(7,34,42,63,64, 74, 83, 84, 91, 108, 112,129, 133,133,139,140,140,146,
      149,154,157,160,160,165,173,176,185, 218,225,241, 248,273,277,279,297,
      319,405,417,420,440, 523,523,583, 594, 1101, 1116, 1146, 1226, 1349,
      1412, 1417)
cens<-c(1,1,1,1,1,0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,0,1,1,1,1,1,1,
        0,1,0,1,1,1,1,1,0,1,1,1,0,1)
xin<-xin/30.438    #Adjust the data
storage.mode(xin)<-"integer"  # turn the data to integers
xout<-seq(1,47, by=1)  # define the grid points to evaluate the estimate
arg<-TutzPritscher(xin,cens,xout)   #Discrete kernel estimate
plot(xout, arg, type="l", ylim=c(0, .35), lty=2,  col=6) # plot the estimate
argSM<-lambdahat(xin, cens, xout)  #crude nonparametric estimate
lines(xout, argSM, lty=3, col=5) # plot the crude estimate
```

LLHRPlugInBand *Simple Plug in badnwidth selector*

Description

Provides the asymptotic MISE optimal plug-in bandwidth for the local linear hazard rate estimator LocLinEst, defined in (4), Bagkavos (2011). This is the binned data version of the PlugInBand AMISE optimal bandwidth rule.

Usage

```
LLHRPlugInBand(BinCenters, h, kfun, Delta, xin, xout, IntKfun, ci, cens)
```

Arguments

`BinCenters`	A vector of data points, the centers of the bins resulting from the discretization of the data.
`h`	Bandwidth for the estimate of the distribution function.
`kfun`	A kernel function.
`Delta`	A scalar. The length of the bins.
`xin`	A vector of data points
`xout`	The point at which the estimates should be calculated.
`IntKfun`	The integrated kernel function.
`ci`	Crude hazard rate estimates.
`cens`	Censoring Indicators.

Details

The bandwidth selector requires binned data, i.e. data in the form (x_i, y_i) where x_i are the bin centers and y_i are empirircal hazard rate estimates at each x_i. This is achieved via the `DiscretizeData` function. As it can be seen from (4) in Bagkavos (2011), the bandwidth selector also requires an estimate of the functional

$$\int \left\{\lambda^{(2)}(x)\right\}^2 dx$$

which is readily implemented in `PlugInBand`. It also requires an estimate of the constant

$$\int \frac{\lambda(x)}{1-F(x)}\, dx$$

For this reason additionally the plug in bandwidth rule is also used, as it is implemented in the `bw.nrd` distribution function default bandwidth rule of Swanepoel and Van Graan (2005). The constants $R(K)$ and $\mu_2^2(K)$ are deterministic and specific to the kernel used in the implementation hence can be claculated precisely.

Value

A scalar with the value of the suggested bandwidth.

References

Bagkavos (2011), Annals of the Institute of Statistical Mathematics, 63(5), 1019-1046.

See Also

`PlugInBand`

Examples

```
x<-seq(0, 5,length=100) #design points where the estimate will be calculated
SampleSize<-100 #amount of data to be generated
ti<- rweibull(SampleSize, .6, 1) # draw a random sample
ui<-rexp(SampleSize, .2)         # censoring sample
cat("\n AMOUNT OF CENSORING: ", length(which(ti>ui))/length(ti)*100, "\n")
x1<-pmin(ti,ui)                  # observed data
cen<-rep.int(1, SampleSize)      # initialize censoring indicators
cen[which(ti>ui)]<-0             # 0's correspond to censored indicators

a.use<-DiscretizeData(ti, x)     # discretize the data
BinCenters<-a.use$BinCenters     # get the data centers
ci<-a.use$ci                     # get empircal hazard rate estimates
Delta=a.use$Delta                # Binning range
h2<-bw.nrd(ti)                   # Bandwidth to use in constant est. of the plug in rule
h.use<-h2                        # the first element is the band to use

huse1<- LLHRPlugInBand(BinCenters,h.use,Epanechnikov,Delta,ti,x,IntEpanechnikov,ci,cen)
huse1
```

`LocLinEst` *Local Linear Hazard Rate Estimator*

Description

Implements the local linear kernel hazard rate estimate of Bagkavos and Patil (2008) and Bagkavos (2011). The estimate assumes binned data (fixed design), of the form (x_i, y_i) where x_i are the bin centers and y_i are empirircal hazard rate estimates at each x_i. These are calculated via the `DiscretizeData` function. The estimate then smooths the empircal hazard rate estimates and achieves automatic boundary adjustments through approrpiately defined kernel weights. The user is able to supply their own bandwidth values through the h argument.

Currently only the `LLHRPlugInBand` bandwidth selector is provided which itself it depends on the `bw.nrd` distribution function default bandwidth rule of Swanepoel and Van Graan (2005) for the constant estimate.

- TO DO: In future implementations the EBBS (empirical bias bandwidth) and AIC based bandwidth methods (see Bagkavos (2011)) will be added to the package

Usage

```
LocLinEst(BinCenters, xout, h, kfun, ci)
```

Arguments

`BinCenters`	A vector with the bin centers of the discretized data.
`xout`	A vector of points at which the hazard rate function will be estimated.

h	A scalar, the bandwidth to use in the estimate.
kfun	Kernel function to use. Supported kernels: Epanechnikov, Biweight, Gaussian, Rectangular, Triangular
ci	Empirical hazard rate estimates.

Details

The estimate in both Bagkavos and Patil (2008) and Bagkavos (2011) is given by

$$\hat{\lambda}_L(x) = \frac{T_{n,1}(x)S_{n,1}(x) - T_{n,0}(x)S_{n,2}(x)}{S_{n,1}(x)S_{n,1}(x) - S_{n,0}(x)S_{n,2}(x)}.$$

The difference between the censored and the uncensored cased is only on the calculation of the empirical hazard rate estimates.

Value

A vector with the values of the function at the designated points xout.

References

1. Bagkavos and Patil, Local Polynomial Fitting in Failure Rate Estimation, IEEE Transactions on Reliability, 57, (2008),
2. Bagkavos (2011), Annals of the Institute of Statistical Mathematics, 63(5), 1019-1046,

See Also

HazardRateEst, LLHRPlugInBand

Examples

```
x<-seq(0.05, 5,length=80) #grid points to calculate the estimates
plot(x, HazardRate(x,"weibull", .6, 1),type="l", xlab = "x",ylab="Hazard rate")

SampleSize = 100                #select sample size
ti<- rweibull(SampleSize, .6, 1) # draw a random sample
ui<-rexp(SampleSize, .2)        # censoring sample
cat("\n AMOUNT OF CENSORING: ", length(which(ti>ui))/length(ti)*100, "\n")
x1<-pmin(ti,ui)                 # observed data
cen<-rep.int(1, SampleSize)     # initialize censoring indicators
cen[which(ti>ui)]<-0            # 0's correspond to censored indicators

a.use<-DiscretizeData(ti, x)    # discretize the data
BinCenters<-a.use$BinCenters    # get the data centers
ci<-a.use$ci                    # get empircal hazard rate estimates
Delta=a.use$Delta               # Binning range
h2<-bw.nrd(ti)                  # Bandwidth to use in constant est. of the plug in rule
h.use<-h2                       # the first element is the band to use

# Calcaculate the plug-in bandwidth:
huse1<- LLHRPlugInBand(BinCenters,h.use,Epanechnikov,Delta,ti,x,IntEpanechnikov,ci, cen)
```

```
arg2<-HazardRateEst(x1,x,Epanechnikov, huse1, cen)      # Tanner-Wong Estimate
lines(x, arg2, lty=2) # draw the Tanner-Wong   estimate # Draw TW estimate
arg5<-HazardRateEst(x1,x,BoundaryBiweight,huse1,cen)    # Boundary adjusted TW est
lines(x, arg5, lty=2, col=4) # draw the variable bandwidth # Draw the estimate
arg6<-LocLinEst(BinCenters ,x, huse1, Epanechnikov, ci) # Local linear est.
lines(x, arg6, lty=5, col=5)                             # Draw the estimate
legend("topright", c("Tanner-Wong",  "TW - Boundary Corrected", "Local Linear"),
          lty=c(2,2, 5), col=c(1,4, 5)) # add legend
```

`NP.M.Estimate` *Estimate of bandwidth constant*

Description

Calculation of the contant term in the AMISE plugin bandwidth rule `PlugInBand`.

Usage

```
NP.M.Estimate(xin, cens, xout)
```

Arguments

`xin`	A vector of data points
`xout`	The point at which the estimates should be calculated.
`cens`	Censoring Indicators.

Details

Approximates the term

$$M = \int_0^T \frac{\lambda_T(x)}{1 - F(x)}\, dx$$

which is needed in the optimal AMISE bandwidth expression of `PlugInBand`. The integrand

$$\frac{\lambda_T(x)}{1 - F(x)}\, dx$$

is calculated by `HRSurv` and integration is performed via the extended Simpson's numerical integration rule (`SimpsonInt`).

Value

A scalar with the value of the constant.

References

Hua, Patil and Bagkavos, An L_1 analysis of a kernel-based hazard rate estimator, Australian and New Zealand J. Statist., (60), 43-64, (2018).

nsf, Tm, CparamCalculation, power.matrix, base, SmoothedEstimate
Auxiliary functions for discrete hazard rate estimators

Description

Auxiliary functions for discrete semiparametric and kernel smooth hazard rate estimation

Usage

```
nsf(xin, cens, xout)
Tm(tk, xout, distribution, par1, par2)
CparamCalculation(gamparam, VehHazard)
power.matrix(M, n)
base(m, b)
SmoothedEstimate(NonParEst, VehHazard, gammapar, SCproduct, Cpar)
```

Arguments

`xin`	A vector of data points. Missing values not allowed.
`cens`	A vector of 1s and zeros, 1's indicate uncensored observations, 0's correspond to censored obs.
`xout`	The points where the estimate should be calculated.
`tk`	desing points for the NPMLE estimate.
`distribution`	which distribution to use?
`par1`	distribution parameter 1
`par2`	distribution parameter 2
`gamparam`	gamma parameter
`M`	a matrix to be raised to a power
`n`	the power the matrix will be raised at
`m`	express m as a power of b
`b`	express m as a power of b
`NonParEst`	The crude nonparametric hazard rate estimate.
`VehHazard`	Vehicle hazard rate
`gammapar`	gamma parameter
`SCproduct`	SC product, the result of DetermineSCprod
`Cpar`	C parameter, the result of CparamCalculation.

Details

Auxiliary functions for discrete hazard rate estimators. The function `nsf` is used for the kernel smooth estimate `TutzPritscher`.

- Tm used to calculate $\max(t_k; 1-\sum_{l=0}^{k} \eta_l > \epsilon), \epsilon > 0$ in the implementation of the semiparametric estimate
- CparamCalculationreturns the C smoothing parameter calculated as $C = \gamma / \max_{k \geq 0}(\lambda(t_{k-1}) + \lambda(t_k) + \lambda(t_{k+1}))$
- DetermineSCprodthis finds $SC = \gamma((n+1)\hat{B}_1)^{-1}\hat{V}_1$ n = number of obs, gammapar = sum of vehicle haz at xout (computed elsewhere)

Value

A vector with the values of the hazard rate estimates.

References

1. Patil and Bagkavos (2012), Semiparametric smoothing of discrete failure time data, Biometrical Journal, 54, (2012), 5–19.
2. Tutz, G. and Pritscher, L. Nonparametric Estimation of Discrete Hazard Functions, Lifetime Data Anal, 2, 291-308 (1996)

`PlugInBand` *Simple Plug in badnwidth selector*

Description

Provides the asymptotic MISE optimal plug-in bandwidth for the hazard rate estimator `HazardRateEst`, see Hua, Patil and Bagkavos (2018). The bandwidth is also suitable for use as a pilot bandwidth in `TransHazRateEst` and `VarBandHazEst`.

Usage

```
PlugInBand(xin, xout,   cens, kfun )
```

Arguments

`xin`	A vector of data points
`xout`	The point at which the estimates should be calculated.
`cens`	Censoring Indicators.
`kfun`	A kernel function.

Details

The asymptotic MISE optimal plug-in bandwidth selector for HazardRateEst is defined by

$$h_{opt} = \left[\frac{R(K)}{nR(\lambda_T'')\mu_{2,K}^2} \int \frac{\lambda_T(x)}{1-F(x)}\, dx \right]^{1/5}$$

see (9) in Hua, Patil and Bagkavos (2018). The estimate of $R(\lambda_T'')$ to be used in h_{opt} is

$$R(\hat{\lambda}_T'') = \int_0^{\xi} \left(\hat{\lambda}_T''(x|\hat{b}_n^*) \right)^2 dx.$$

Also,

$$\int_0^T \frac{\lambda_T(x)}{1-F(x)}\, dx$$

is estimated by applying the extended Simpson's numerical integration rule, SimpsonInt, on

$$\frac{\hat{\lambda}_T(x|\hat{b}_n^*)}{1-F(x)}$$

where $1 - F(x)$ is estimated by KMest. The estimation is implemented in the NP.M.Estimate function.

Currently b_n^* is estimated by bw.nrd. However according to (11) in Hua, Patil and Bagkavos (2018)., in future versions this package will support

$$b_n^* = \left\{ \frac{5R(K'')}{n\mu_{2,K}^2 R(\lambda_T^{(4)})} \int \frac{\lambda_T(x)}{1-F(x)}\, dx \right\}^{1/9}.$$

where

$$R(\hat{\lambda}_T^{(4)}) = \frac{(\hat{a}(\hat{a}-1)(\hat{a}-2)(\hat{a}-3)(\hat{a}-4))^2}{(2\hat{a}-9)\hat{b}^{2\hat{a}}} (\xi^{2\hat{a}-9} - {p_\alpha}^{2\hat{a}-9}), \hat{a} \neq 9/2$$

and $\hat{M}$ is already estimated by NP.M.Estimate as expalined above (it will be much more stable than using a Weibull reference model).

Value

A scalar with the value of the suggested bandwidth.

References

Hua, Patil and Bagkavos, An L_1 analysis of a kernel-based hazard rate estimator, Australian and New Zealand J. Statist., (60), 43-64, (2018).

See Also

HazardRateEst, LLHRPlugInBand

Examples

```
x<-seq(0, 5,length=100) #design points where the estimate will be calculated
SampleSize<-100 #amount of data to be generated
ti<- rweibull(SampleSize, .6, 1) # draw a random sample
ui<-rexp(SampleSize, .2)         # censoring sample
cat("\n AMOUNT OF CENSORING: ", length(which(ti>ui))/length(ti)*100, "\n")
x1<-pmin(ti,ui)                  # observed data
cen<-rep.int(1, SampleSize)      # initialize censoring indicators
cen[which(ti>ui)]<-0             # 0's correspond to censored indicators

huse1<- PlugInBand(x1, x, cen, Biweight)
huse1
```

RdistSwitch, PdfSwitch, CdfSwitch, HazardRate

User driven input for random number generation and pdf, survival and hazard rate function calculation

---

Description

Auxiliary functions that help automate the process of random number generation or pdf, survival function or hazard rate functions

Usage

```
RdistSwitch(dist, SampleSize, par1, par2)
PdfSwitch(xout, dist, par1, par2)
CdfSwitch(xout, dist, par1, par2)
HazardRate(xout, dist, par1, par2)
```

Arguments

`dist`	A string. Corresponds to one of weibull, lognorm, chisquare, exponential, binomial, geometric, poisson, negativebinomial, uniform
`SampleSize`	The size of the random sample to be drawn
`xout`	Grid points where the function will be evaluated
`par1`	parameter 1 of the distirbution
`par2`	parameter 2 of the distirbution

Details

Implements random number generation and density, survival and hazard rate estimates for several distributions. These functions are mainly used when simulating the mean square error etc from known distributions.

Value

A vector with the values of the function at the designated points xout or the random numbers drawn.

SDHazardRateEst *Kernel Second Derivative Hazard Rate Estimation*

Description

Implements the kernel estimate of the second derivative of the hazard rate for right censored data defined - based on the estimate of Tanner and Wong (1983). The implementation is based on the second derivative of the Biweight Kernel.

Usage

```
SDHazardRateEst(xin, xout, h, ci)
```

Arguments

xin	A vector of data points. Missing values not allowed.
xout	A vector of grid points at which the estimates will be calculated.
h	A scalar, the bandwidth to use in the estimate.
ci	A vector of censoring indicators: 1's indicate uncensored observations, 0's correspond to censored obs.

Details

The function `SDHazardRateEst` implements the kernel estimate of the second derivative of the hazard rate estimator, given by

$$\hat{\lambda}_2(x;h) = \sum_{i=1}^{n} \frac{K_h''(x - X_{(i)})\delta_{(i)}}{n - i + 1}$$

where K is taken to be the `Biweight` kernel. The function is used for estimation of the functional $R(\lambda'')$ in `PlugInBand` so a default bandwidth rule is used for h provided in (16), Hua, Patil and Bagkavos (2018).

Value

A vector with the second derivative of the hazard rate at the designated points xout.

References

1. Tanner and Wong (1983), The Estimation Of The Hazard Function From Randomly Censored Data By The Kernel Method, Annals of Statistics, 3, pp. 989-993.
2. Hua, Patil and Bagkavos, An L_1 analysis of a kernel-based hazard rate estimator, Australian and New Zealand J. Statist., (60), 43-64, (2018).

SemiparamEst *Discrete hazard rate estimator*

Description

Implements the semiparametric hazard rate estimator for discrete data developed in Patil and Bagkavos (2012). The estimate is obtained by semiparametric smoothing of the (nonsmooth) nonparametric maximum likelihood estimator, which is achieved by repeated multiplication of a Markov chain transition-type matrix. This matrix is constructed with basis a parametric discrete hazard rate model (vehicle model).

Usage

```
SemiparamEst(xin, cens, xout, Xdistr, Udistr, vehicledistr, Xpar1=1, Xpar2=0.5,
              Upar1=1, Upar2=0.5, vdparam1=1, vdparam2=0.5)
```

Arguments

`xin`	A vector of data points. Missing values not allowed.
`cens`	Censoring indicators as a vector of 1s and zeros, 1's indicate uncensored observations, 0's correspond to censored obs.
`xout`	Design points where the estimate will be calculated.
`Xdistr`	The distribution where the data are coming from, currently ignored
`Udistr`	Censoring distribution, currently ignored
`vehicledistr`	String specifying the vehicle hazard rate (the assumed parametric model)
`Xpar1`	Parameter 1 for the X distr, currently ignored
`Xpar2`	Parameter 2 for the X distr, currently ignored
`Upar1`	Parameter 1 for the Cens. distr., currently ignored
`Upar2`	Parameter 2 for the Cens. distr., currently ignored
`vdparam1`	Parameter 1 for the vehicle hazard rate.
`vdparam2`	Parameter 2 for the vehicle hazard rate.

Details

The semiparmaetric estimator implemented is defined in (1) in Patil and Bagkavos (2012) by

$$\tilde{\lambda} = \hat{\lambda}\Gamma^S$$

where S determines the number of repetions and hence the amount of smoothing applied to the estimate. For $S = 0$ the semiparametric estimate equals the nonparmaetric estimate `lambdahat`. On the other hand, if the true unknown underlying probability model is known (up to an unknown constant or constants) then, the greater the S, the closer the semiparmaetric estimate to the vehicle hazard rate model.

- TO DO: The extension to hazard rate estimation with covariates will be added in a future release.
- TO DO: Also, the data driven estimation of the parameter S will be also added in a future release; this will inlcude the SC product and C and γ parameter calculations.

Value

A vector with the values of the discrete hazard rate estimate, calculated at $x = xout$.

References

Patil and Bagkavos (2012), Semiparametric smoothing of discrete failure time data, Biometrical Journal, 54, (2012), 5-19

See Also

lambdahat, TutzPritscher

Examples

```
options(echo=FALSE)
xin<-c(7,34,42,63,64, 74, 83, 84, 91, 108, 112,129, 133,133,139,140,140,146,
      149,154,157,160,160,165,173,176,185, 218,225,241, 248,273,277,279,297,
      319,405,417,420,440, 523,523,583, 594, 1101, 1116, 1146, 1226, 1349,
      1412, 1417) #head and neck data set
cens<-c(1,1,1,1,1,0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,
        1,0,1,1,1,1,1,1,0,1,0,1,1,1,1,1,0,1,1,1,0,1) #censoring indicators
xin<-xin/30.438 #mean adjust the data
storage.mode(xin)<-"integer"  # turn the data to integers
xout<-seq(1,47, by=1) #design points where to calculate the estimate
arg<-TutzPritscher(xin,cens,xout) #Kernel smooth estimate
plot(xout, arg, type="l", ylim=c(0, .35), lty=2,  col=6)
argSM<-SemiparamEst(xin, cens, xout, "geometric", "uniform",
                    "geometric", 0.2, .6, 0, 90, .25, .9) #semipar. est.
lines(xout, argSM[,2], lty=3, col=5) #add tilde lambda to the plot
```

SimpsonInt *Simpson numerical integration*

Description

Implements Simpson's extended numerical integration rule

Usage

```
SimpsonInt(xin, h)
```

Arguments

`xin`	A vector of data points
`h`	grid length

Details

The extended numerical integration rule is given by

$$\int_0^{x_{2n}} f(x)\,dx = \frac{h}{3}(f(x_0)+4\{f(x_1)+\ldots f(x_{2n-1})\}+2\{f(x_2)+f(x_4)+\ldots f(x_{2n-2})\}+f(x_{2n}))-R_n$$

Value

returns the approximate integral value

References

Weisstein, Eric W. "Simpson's Rule." From MathWorld–A Wolfram Web Resource

---

`sn.i, tn.i` *Local kernel weights*

Description

Implements the local kernel weights which are used in the implementation of `LocLinEst` and the second derivative estimate used in `PlugInBand`.

Usage

```
sn.0(xin, xout, h, kfun)
sn.1(xin, xout, h, kfun)
sn.2(xin, xout, h, kfun)
sn.3(xin, xout, h, kfun)
sn.4(xin, xout, h, kfun)
sn.5(xin, xout, h, kfun)
sn.6(xin, xout, h, kfun)
tn.0(xin, xout, h, kfun, Y)
tn.1(xin, xout, h, kfun, Y)
tn.2(xin, xout, h, kfun, Y)
tn.3(xin, xout, h, kfun, Y)
```

Arguments

`xin`	A vector of data points, typicaly these are the bin centers. Missing values not allowed.
`xout`	A vector of data points where the estimate will be evaluated.
`h`	A scalar. The bandwidth to use.
`kfun`	The kernel function to use.
`Y`	Empirical hazard rate estimates.

Details

The functions calculate the quantities

$$S_{n,l}(x) = \sum_{i=1}^{n} K\left(\frac{x_i - x}{h}\right)(x_i - x)^l, l = 0, \ldots, 6$$

and

$$T_{n,l}(x) = \sum_{i=1}^{n} K\left(\frac{x_i - x}{h}\right)(x_i - x)^l Y_i, l = 0, \ldots, 3$$

These qunatities are used to adjust the hazard rate estimate and its second derivative in the boundary.

Value

The weight of the functional at x

References

1. Bagkavos and Patil, Local Polynomial Fitting in Failure Rate Estimation, IEEE Transactions on Reliability, 57, (2008),
2. Bagkavos (2011), Annals of the Institute of Statistical Mathematics, 63(5), 1019-1046.

`TransHazRateEst` *Transformation Based Hazard Rate Estimator*

Description

Implements the transformated kernel hazard rate estimator of Bagkavos (2008). The estimate is expected to have less bias compared to the ordinary kernel estimate `HazardRateEst`. The estimate results by first transforming the data to a sample from the exponential distribution through the integrated hazard rate function, estimated by `iHazardRateEst` and uses the result as input to the classical kernel hazard rate estimate `HazardRateEst`. An inverse transform turn the estimate to a hazard rate estimate of the original sample. See section "Details" below.

Usage

```
TransHazRateEst(xin, xout, kfun, ikfun, h1, h2, ci)
```

Arguments

`xin`	A vector of data points. Missing values not allowed.
`xout`	A vector of points at which the hazard rate function will be estimated.
`kfun`	Kernel function to use. Supported kernels: Epanechnikov, Biweight, Gaussian, Rectangular, Triangular, HigherOrder.
`ikfun`	An integrated kernel function to use. Supported kernels: Epanechnikov, Biweight, Gaussian, Rectangular, Triangular, HigherOrder.

h1	A scalar, pilot bandwidth.
h2	A scalar, transformed kernel bandwidth.
ci	A vector of censoring indicators: 1's indicate uncensored observations, 0's correspond to censored obs.

Details

The transformed kernel hazard rate estimate of Bagkavos (2008) is given by

$$\hat{\lambda}_t(x; h_1, h_2) = \sum_{i=1}^{n} \frac{K_{h_2}\left\{(\hat{\Lambda}(x; h_1) - \hat{\Lambda}(X_{(i)}; h_1))\right\}\delta_{(i)}}{n-i+1} \hat{\lambda}(x; h_1).$$

The estimate uses the classical kernel hazard rate estimate $\lambda(x; h_1)$ implemented in `HazardRateEst` and its integrated version

$$\hat{\Lambda}(x; h_1) = \sum_{i=1}^{n} \frac{k\left\{(x - X_{(i)})h_1^{-1}\right\}\delta_{(i)}}{n-i+1}$$

where $k(x) = \int_{-\infty}^{x} K(y)\,dy$ implemented in `iHazardRateEst`. The pilot bandwidth h_1 is determined by an optimal bandwidth rule such as `PlugInBand`.

- TO DO: Insert a rule for the adaptive bandwidth h_2.

Value

A vector with the values of the function at the designated points xout.

References

Bagkavos (2008), Transformations in hazard rate estimation, J. Nonparam. Statist., 20, 721-738

See Also

`VarBandHazEst`, `HazardRateEst`, `PlugInBand`

Examples

```
x<-seq(0, 5,length=100) #design points where the estimate will be calculated
plot(x, HazardRate(x, "weibull", .6, 1),  type="l",
                   xlab = "x", ylab="Hazard rate") #plot true hazard rate function
SampleSize <- 100
mat<-matrix(nrow=SampleSize, ncol=20)
for(i in 1:20)
{ #Calculate the average of 20 estimates and draw on the screen
 ti<- rweibull(SampleSize, .6, 1)  #draw a random sample from the actual distribution
 ui<-rexp(SampleSize, .05)         #draw a random sample from the censoring distribution
 cat("\n AMOUNT OF CENSORING: ", length(which(ti>ui))/length(ti)*100, "\n")
 x1<-pmin(ti,ui)                   #this is the observed sample
 cen<-rep.int(1, SampleSize)       #censoring indicators
 cen[which(ti>ui)]<-0              #censored values correspond to zero
```

```
 h2<-DefVarBandRule(ti, cen)     #Deafult Band. Rule - Weibull Reference
 huse1<- PlugInBand(x1, x,   cen, Biweight) #
 mat[,i]<-TransHazRateEst(x1,x,Epanechnikov,IntEpanechnikov,huse1,h2,cen)
}
lines(x, rowMeans(mat) , lty=2) #draw the average transformed estimate
```

`TutzPritscher` *Discrete non parametric kernel hazard rate estimator*

Description

Implementation of the kernel discrete hazard rate estimator of Tutz and Pritscher (1996) based on the discrete `Habbema` kernel. The estimate is used for comparison with the semiparametric estimate deveoped in Tutz and Pritscher (1996).

Usage

```
TutzPritscher(xin, cens, xout)
```

Arguments

`xin`	A vector of data points. Missing values not allowed.
`cens`	Censoring indicators as a vector of 1s and zeros, 1's indicate uncensored observations, 0's correspond to censored obs.
`xout`	The grid points where the estimates will be calculated.

Details

The discrete kernel estimate of Tutz and Pritscher (1996) is defined by

$$\hat{\lambda}(t_m|v) = \sum_{s=1}^{q} \sum_{i=1}^{m_s} w_m\left((t,x),(s,x_{is})\right)\tilde{\lambda}(s|x_{is})$$

where w_m is the discrete Habbema kernel.

Value

Returns a vector with the values of the hazard rate estimates at $x = xout$.

References

Tutz, G. and Pritscher, L. Nonparametric Estimation of Discrete Hazard Functions, Lifetime Data Anal, 2, 291-308 (1996)

See Also

`SemiparamEst`

Examples

```
options(echo=FALSE)
xin<-c(7,34,42,63,64, 74, 83, 84, 91, 108, 112,129, 133,133,139,140,140,146,
      149,154,157,160,160,165,173,176,185, 218,225,241, 248,273,277,279,297,
      319,405,417,420,440, 523,523,583, 594, 1101, 1116, 1146, 1226, 1349,
      1412, 1417)
cens<-c(1,1,1,1,1,0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,0,1,1,1,1,1,1,
        0,1,0,1,1,1,1,1,0,1,1,1,0,1)
xin<-xin/30.438    #Adjust the data
storage.mode(xin)<-"integer"  # turn the data to integers
xout<-seq(1,47, by=1)  # define the grid points to evaluate the estimate
arg<-TutzPritscher(xin,cens,xout)   #Discrete kernel estimate
plot(xout, arg, type="l", ylim=c(0, .35), lty=2,  col=6) # plot the estimate
argSM<-lambdahat(xin, cens, xout)  #crude nonparametric estimate
lines(xout, argSM, lty=3, col=5) # plot the crude estimate
```

VarBandHazEst   *Variable Bandwidth Hazard Rate Estimator*

Description

Implements the adaptive variable bandwidth hazard rate estimator of Bagkavos and Patil (2009). The estimate itself is an extension of the classical kernel hazard rate estimator of Tanner and Wong (1983) implemented in `HazardRateEst`. The difference is that instead of h, the variable bandwidth estimate uses bandwidth $h\lambda(X_i)^{-1/2}$. This particular choice cancels the second order term in the bias expansion of the hazard rate estimate and thus it is expected to result in a more precise estimation compared to `HazardRateEst`.

Usage

```
VarBandHazEst(xin, xout, kfun, h1, h2, ci)
```

Arguments

`xin`	A vector of data points. Missing values not allowed.
`xout`	A vector of points at which the hazard rate function will be estimated.
`kfun`	Kernel function to use. Supported kernels: Epanechnikov, Biweight, Gaussian, Rectangular, Triangular, HigherOrder
`h1`	A scalar, pilot bandwidth.
`h2`	A scalar, variable kernel (adaptive) bandwidth.
`ci`	A vector of censoring indicators: 1's indicate uncensored observations, 0's correspond to censored obs.

Details

Implements the adaptive variable bandwidth hazard rate estimator of Bagkavos and Patil (2009), Comm. Statist. Theory and Methods.

$$\hat{\lambda}_v(x; h_1, h_2) = \sum_{i=1}^{n} \hat{\lambda}^{-1/2}(x; h_1) \frac{K_{h_2}\left\{(x - X_{(i)})\hat{\lambda}^{-1/2}(x; h_1)\right\}\delta_{(i)}}{n - i + 1}$$

The pilot bandwidth h_1 is determined by an optimal bandwidth rule such as `PlugInBand`. and used as input to the pilot kernel estimate, implemented by `HazardRateEst`.

- TO DO: Insert a rule for the adaptive bandwidth h_2.

Value

A vector with the values of the function at the designated points xout.

References

Bagkavos and Patil (2009), Variable Bandwidths for Nonparametric Hazard Rate Estimation, Communications in Statistics - Theory and Methods, 38:7, 1055-1078

See Also

`HazardRateEst`, `TransHazRateEst`, `PlugInBand`

Examples

```
x<-seq(0, 5,length=100) #design points where the estimate will be calculated
plot(x, HazardRate(x, "weibull", .6, 1),  type="l",
     xlab = "x", ylab="Hazard rate") #plot true hazard rate function
SampleSize <- 100
mat<-matrix(nrow=SampleSize, ncol=20)
for(i in 1:20)
{
  ti<- rweibull(SampleSize, .6, 1)#draw a random sample from the actual distribution
  ui<-rexp(SampleSize, .05)       #draw a random sample from the censoring distribution
  cat("\n AMOUNT OF CENSORING: ", length(which(ti>ui))/length(ti)*100, "\n")
  x1<-pmin(ti,ui)                 #this is the observed sample
  cen<-rep.int(1, SampleSize)     #censoring indicators
  cen[which(ti>ui)]<-0            #censored values correspond to zero

  h2<-DefVarBandRule(ti, cen)     #Deafult Band. Rule - Weibull Reference
  huse1<-  PlugInBand(x1, x,   cen, Biweight)
  mat[,i]<- VarBandHazEst(x1, x, Epanechnikov, huse1,h2, cen) #Var. bandwidth est.
}
lines(x, rowMeans(mat) , lty=2)   #draw the average  vb estimate
```

Index