

# Package ‘lpint’

May 8, 2026

**Type** Package

**Title** Local Polynomial Estimators of the Intensity Function and Its Derivatives

**Version** 2.1

**Date** 2022-04-12

**Author** Feng Chen <feng.chen@unsw.edu.au>

**Maintainer** Feng Chen <feng.chen@unsw.edu.au>

**Description** Functions to estimate the intensity function and its derivative of a given order of a multiplicative counting process using the local polynomial method.

**License** GPL (>= 2.0)

**LazyLoad** yes

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2022-04-12 03:52:27 UTC

## Contents

lpint-package . . . . .	1
lpint . . . . .	2
lpikint . . . . .	4

<b>Index</b>	<b>7</b>
--------------	----------

---

lpint-package	<i>Local Polynomial Estimators of the Intensity Function of a Counting Process and Its Derivatives</i>
---------------	--

---

## Description

Estimates the intensity function or its derivative of a give a given order using the local polynomial method with automatic bandwidth selection using a rule of thumb plug-in approach.

**Details**

Package: lpint  
 Type: Package  
 Version: 1.0  
 Date: 2012-09-21  
 License: GPL (>=2.0)  
 LazyLoad: yes

**Author(s)**

Feng Chen <feng.chen@unsw.edu.au> Maintainer: Feng Chen <feng.chen@unsw.edu.au>

**References**

- Chen, F. (2011) Maximum local partial likelihood estimators for the counting process intensity function and its derivatives. *Statistica Sinica* 21(1): 107 -128. <http://www3.stat.sinica.edu.tw/statistica/j21n1/J21N14/J21N14.htm>
- Chen, F., Yip, P.S.F., & Lam, K.F. (2011) On the Local Polynomial Estimators of the Counting Process Intensity Function and its Derivatives. *Scandinavian Journal of Statistics* 38(4): 631 - 649. <http://dx.doi.org/10.1111/j.1467-9469.2011.00733.x>
- Chen, F., Higgins, R.M., Yip, P.S.F. & Lam, K.F. (2008) Nonparametric estimation of multiplicative counting process intensity functions with an application to the Beijing SARS epidemic, *Communications in Statistics - Theory and Methods* 37: 294 - 306. <http://www.tandfonline.com/doi/abs/10.1080/03610920701649035>
- Chen, F., Higgins, R.M., Yip, P.S.F. & Lam, K.F. (2008) Local polynomial estimation of Poisson intensities in the presence of reporting delays, *Journal of the Royal Statistical Society Series C (Applied Statistics)* 57(4): 447 - 459. <http://onlinelibrary.wiley.com/doi/10.1111/j.1467-9876.2008.00624.x/full>

---

lpint

*Martingale estimating equation local polynomial estimator of counting process intensity function and its derivatives*

---

**Description**

This local polynomial estimator is based on a biased martingale estimating equation.

**Usage**

```
lpint(jmptimes, jmpsizes = rep(1, length(jmptimes)),
      Y = rep(1,length(jmptimes)), bw = NULL,
      adjust = 1, Tau = max(1, jmptimes), p = nu + 1,
      nu = 0, K = function(x) 3/4 * (1 - x^2) * (x <= 1 & x >= -1),
      n = 101, bw.only=FALSE)
```

**Arguments**

jmptimes	a numeric vector giving the jump times of the counting process
jmpsizes	a numeric vector giving the jump sizes at each jump time. Need to be of the same length as jmptimes
Y	a numeric vector giving the value of the exposure process (or size of the risk set) at each jump times. Need to be of the same length as jmptimes
bw	a numeric constant specifying the bandwidth used in the estimator. If left unspecified the automatic bandwidth selector will be used to calculate one.
adjust	a positive constant giving the adjust factor to be multiplied to the default bandwidth parameter or the supplied bandwidth
Tau	a numeric constant >0 giving the censoring time (when observation of the counting process is terminated)
p	the degree of the local polynomial used in constructing the estimator. Default to 1 plus the degree of the derivative to be estimated
nu	the degree of the derivative of the intensity function to be estimated. Default to 0 for estimation of the intensity itself.
K	the kernel function
n	the number of evenly spaced time points to evaluate the estimator at
bw.only	TRUE or FALSE according as if the rule of thumb bandwidth is the only required output or not

**Value**

either a list containing

x	the vector of times at which the estimator is evaluated
y	the vector giving the values of the estimator at times given in x
se	the vector giving the standard errors of the estimates given in y
bw	the bandwidth actually used in defining the estimator equal the automatically calculated or supplied multiplied by adjust

or a numeric constant equal to the rule of thumb bandwidth estimate

**Author(s)**

Feng Chen <feng.chen@unsw.edu.au.>

**References**

Chen, F., Yip, P.S.F., & Lam, K.F. (2011) On the Local Polynomial Estimators of the Counting Process Intensity Function and its Derivatives. *Scandinavian Journal of Statistics* 38(4): 631 - 649. <http://dx.doi.org/10.1111/j.1467-9469.2011.00733.x>

**See Also**

[lplikint](#)

## Examples

```
##simulate a Poisson process on [0,1] with given intensity
int <- function(x)100*(1+0.5*cos(2*pi*x))
censor <- 1
set.seed(2)
N <- rpois(1,150*censor);
jtms <- runif(N,0,censor);
jtms <- jtms[as.logical(mapply(rbinom,n=1,size=1,prob=int(jtms)/150))];

##estimate the intensity
intest <- lpint(jtms,Tau=censor)
##plot and compare
plot(intest,xlab="time",ylab="intensity",type="l",lty=1)
curve(int,add=TRUE,lty=2)

## Example estimating the hazard function from right censored data:
## First simulate the (not directly observable) life times and censoring
## times:
lt <- rweibull(500,2.5,3); ct <- rlnorm(500,1,0.5)
## Now the censored times and censorship indicators delta (the
## observables):
ot <- pmin(lt,ct); dlt <- as.numeric(lt <= ct);
## Estimate the hazard rate based on the censored observations:
jtms <- sort(ot[dlt==1]);
Y <- sapply(jtms,function(x)sum(ot>=x));
haz.est <- lpint(jtms,Y=Y);
## plot the estimated hazard function:
matplot(haz.est$x,
        pmax(haz.est$y+outer(haz.est$se,c(-1,0,1)*qnorm(0.975)),0),
        type="l",lty=c(2,1,2),
        xlab="t",ylab="h(t)",
        col=1);
## add the truth:
haz <- function(x)dweibull(x,2.5,3)/pweibull(x,2.5,3,lower.tail=FALSE)
curve(haz, add=TRUE,col=2)
```

---

lplikint

---

*Partial likelihood based local polynomial estimators of the counting process intensity function and its derivatives*


---

## Description

This local polynomial estimator is based on the (localized) partial likelihood

## Usage

```
lplikint(jmptimes, jmpsizes = rep(1, length(jmptimes)),
        Y = rep(1,length(jmptimes)),
        K = function(x) 3/4 * (1 - x^2) * (x <= 1 & x >= -1),
```

```
bw, adjust = 1, nu = 0, p = 1, Tau = 1, n = 101,
tseq = seq(from = 0, to = Tau, length = n), tol = 1e-05,
maxit = 100, us = 10, gd = 5)
```

### Arguments

jmptimes	a numeric vector giving the jump times of the counting process
jmpsizes	a numeric vector giving the jump sizes at each jump time. Need to be of the same length as jmptimes
Y	a numeric vector giving the value of the exposure process (or size of the risk set) at each jump times. Need to be of the same length as jmptimes
K	the kernel function
bw	a numeric constant specifying the bandwidth used in the estimator. If left unspecified the automatic bandwidth selector will be used to calculate one.
adjust	a positive constant giving the adjust factor to be multiplied to the default bandwidth parameter or the supplied bandwidth
nu	the degree of the derivative of the intensity function to be estimated. Default to 0 for estimation of the intensity itself.
p	the degree of the local polynomial used in constructing the estimator. Default to 1 plus the degree of the derivative to be estimated
Tau	a numeric constant >0 giving the censoring time (when observation of the counting process is terminated)
n	the number of evenly spaced time points to evaluate the estimator at. Not used when tseq is provided.
tseq	the time sequence at which to evaluate the estimator
tol	the parameter error tolerance used to stop the iterations in optimizing the local likelihood
maxit	maximum number of iterations allowed in the optimization used in a single estimation point
us	a numeric constants used together with gd to grid search for a decent start value in solving the local score equation. The starting value is 0 except on its first dimension, which was chosen so that the starting value is the minimizer of the L <sup>1</sup> norm of the score function among the values: average intensity X us <sup>(-gd:gd)</sup>
gd	a numeric constant used together with us to search for a decent start value in solving the local score equation

### Details

The estimator is based on solving the local score equation using the Newton-Raphson method and extract the appropriate dimension.

**Value**

a list containing	
x	the vector of times at which the estimator is evaluated
y	the vector giving the values of the estimator at times given in x
se	the vector giving the standard errors of the estimates given in y
bw	the bandwidth actually used in defining the estimator equal the automatically calculated or supplied multiplied by adjust
fun	the intensity (or derivative) estimator as a function of the estimation point, which can be called to evaluate the estimator at points not included in tseq

**Author(s)**

Feng Chen <feng.chen@unsw.edu.au.>

**References**

Chen, F. (2011) Maximum local partial likelihood estimators for the counting process intensity function and its derivatives. *Statistica Sinica* 21(1): 107 -128. <http://www3.stat.sinica.edu.tw/statistica/j21n1/J21N14/J21N14.htm>

**See Also**

[lpint](#)

**Examples**

```
##simulate a Poisson process on [0,1] with given intensity
int <- function(x)100*(1+0.5*cos(2*pi*x))
censor <- 1
set.seed(2)
N <- rpois(1,150*censor);
jtms <- runif(N,0,censor);
jtms <- jtms[as.logical(mapply(rbinom,n=1,size=1,prob=int(jtms)/150))];

##estimate the intensity
intest <- lplikint(jtms,bw=0.15,Tau=censor)
#plot and compare
plot(intest,xlab="time",ylab="intensity",type="l",lty=1)
curve(int,add=TRUE,lty=2)
```

# Index

- \* **package**
    - lpint-package, 1
  - \* **smooth**
    - lpint, 2
    - lplikint, 4
  - \* **survival**
    - lpint, 2
    - lplikint, 4
- lpint, 2, 6  
lpint-package, 1  
lplikint, 3, 4