

# Package: smoothROctime (via r-universe)

June 1, 2026

**Type** Package

**Title** Smooth Time-Dependent ROC Curve Estimation

**Version** 0.1.1

**Description** Computes smooth estimations for the Cumulative/Dynamic and Incident/Dynamic ROC curves, in presence of right censorship, based on the bivariate kernel density estimation of the joint distribution function of the Marker and Time-to-event variables.

**License** GPL

**Encoding** UTF-8

**Imports** ks

**Suggests** KMsurv,lattice, survival

**RoxygenNote** 7.3.1

**NeedsCompilation** no

**Author** Susana Diaz-Coto [aut, cre]

**Maintainer** Susana Diaz-Coto <susanacoto@gmail.com>

**Repository** <https://susanadc2009.r-universe.dev>

**Date/Publication** 2025-07-31 23:20:02 UTC

**RemoteUrl** <https://github.com/cran/smoothROctime>

**RemoteRef** HEAD

**RemoteSha** ffdce1fdb4444d5d15968f2f07ad5dc46b654ed

## Contents

smoothROctime-package . . . . .	2
funcen . . . . .	3
plot.sROct . . . . .	5
stRoc . . . . .	6

<b>Index</b>	<b>10</b>
--------------	-----------

---

smoothROctime-package *Smooth Time-Dependent ROC Curve Estimation*

---

## Description

Computes smooth estimations for the Cumulative/Dynamic and Incident/Dynamic ROC curves, in presence of right censorship, based on the bivariate kernel density estimation of the joint distribution function of the Marker and Time-to-event variables.

## Details

- `funcen` - Bivariate kernel density estimation of the joint density function of the (*marker, time-to-event*) variable.
- `stRoc` - Smooth estimations for Cumulative/Dynamic and Incident/Dynamic ROC curves.
- `plot.sROct` - Plots of Cumulative/Dynamic and Incident/Dyanmic ROC curve estimations.

## Author(s)

Susana Diaz-Coto [aut, cre]

Maintainer: Susana Diaz-Coto <susanacoto@gmail.com>

## References

- P. Martinez-Cambor and J. C. Pardo-Fernandez. Smooth time-dependent receiver operating characteristic curve estimators. *Statistical Methods in Medical Research*, 27(3):651-674, 2018. doi:[10.1177/0962280217740786](https://doi.org/10.1177/0962280217740786).
- P. Martinez-Cambor, G. F-Bayon, and S. Perez-Fernandez. Cumulative/dynamic ROC curve estimation. *Journal of Statistical Computation and Simulation*, 86(17):3582-3594, 2016. doi:[10.1080/00949655.2016.1175442](https://doi.org/10.1080/00949655.2016.1175442).
- T. Duong. Bandwidth matrices for multivariate kernel density estimation. Ph.D. Thesis, University of Western, Australia, 2004.
- L. Li, T. Green, and B. Hu. A simple method to estimate the time-dependent receiver operating characteristic curve and the area under the curve with right censored data. *Statistical Methods in Medical Research*, 27(8), 2016. doi:[10.1177/0962280216680239](https://doi.org/10.1177/0962280216680239).

## See Also

CRAN package `ks` is used in this package.

## Description

Computes the kernel density estimation of the bivariate vector (*marker, time – to – event*) with the time-to-event variable subject to right censorship, according to the procedure exposed in [doi:10.1177/0962280217740786](https://doi.org/10.1177/0962280217740786).

## Usage

```
funcen(data, H, bw, adj, ...)
```

## Arguments

data	matrix with three columns: time-to-event, censoring status (0=censored/1=uncensored) and marker.
H	2x2 bandwidth matrix when it is specified in an explicit way.
bw	method for computing the bandwidth matrix. Most of the methods included in the kde function can be used: Hpi, Hpi.diag, Hlscv, Hlscv.diag, Hbcv, Hbcv.diag, Hscv, Hscv.diag, Hucv and Hucv.diag. Other considered methods are naive.pdf ( $\text{diag}(N^{(-1/5)}, N^{(-1/5)})^2$ ) and naive.cdf ( $\text{diag}(N^{(-1/3)}, N^{(-1/3)})^2$ ), where N is the sample size.
adj	adjustment parameter for calculating the bandwidth matrix. Default value 1.
...	kde function arguments can also be used for specifying the way in which the kernel density function estimation should be computed.

## Details

The matrix of bandwidths can be defined by using `H=matrix()` or automatically selected by the method indicated in `bw`.

Given the matrix of bandwidths, H, the argument `adj` modifies it and the final computed matrix is  $\text{adj}^2 H$ .

If H is missing, the `naive.pdf` method is used for obtaining the kernel density estimation.

Function `funcen` generates, from the original set of data, a collection of pseudodata through an iterative weights allocation process, with two main goals: keep the information from the censored observations represented in the sample and prepare data so they can be used as incoming parameters in the `kde` function included in the `ks` package. A weighted kernel density estimation is therefore finally computed.

There should be at least two uncensored observations for computing the density estimation.

Omitted parameters are considered to be the default ones in the `kde` function.

**Value**

An object of class `kde` is returned. It is a list where the most relevant values are:

<code>x</code>	matrix containing the pseudodata values. It has two columns: marker and time-to-event.
<code>eval.points</code>	list of points where the bivariate kernel estimation is calculated.
<code>estimate</code>	values of the density estimation.
<code>H</code>	bandwidth matrix.
<code>names</code>	variable names.
<code>w</code>	weights calculated by the function and allocated to pseudodata.

**References**

P. Martinez-Cambor and J. C. Pardo-Fernandez. Smooth time-dependent receiver operating characteristic curve estimators. *Statistical Methods in Medical Research*, 27(3):651-674, 2018. doi:[10.1177/0962280217740786](https://doi.org/10.1177/0962280217740786).

T. Duong. Bandwidth matrices for multivariate kernel density estimation. Ph.D. Thesis, University of Western, Australia, 2004.

**Examples**

```
library(smoothROctime)
require(KMsurv)
require(lattice)

data(kidtran)

# Preparing data: a logarithmic transformation of the time-to-event variable is made
DT <- cbind(log(kidtran$time), kidtran$delta, kidtran$age)
n <- length(log(kidtran$time))

# Directly definition of the bandwidth matrix
H <- diag((c(sd(kidtran$age), sd(log(kidtran$time))))*n^(-0.2))^2)

# Kernel density function estimation
density <- funcen(data=DT, H=H)

# Plot graphics
wireframe(density$estimate, row.values=density$eval.points[[1]],
          column.values=density$eval.points[[1]], zlab="density")
contour(x=density$eval.points[[1]],
        y=density$eval.points[[2]],
        z=density$estimate,
        ylim=c(6, 10))
```

---

plot.sROct

*Plots of time-dependent ROC curve estimations*


---

## Description

Plots of both Cumulative and Incident/Dynamic ROC curve estimations, provided by function [stRoc](#).

## Usage

```
## S3 method for class 'sROct'
plot(x, tcr, xlab, ylab, type = "l", lwd = 5, ...)
```

## Arguments

x	object of class <a href="#">sROct</a> generated with <a href="#">stRoc</a> function and containing the estimations of the time-dependent ROC curves for one single point or a vector of points.
tcr	type of time-dependent ROC curve estimation that will be plotted: <ul style="list-style-type: none"> <li>• “C” for Cumulative/Dynamic,</li> <li>• “I” for Incident/Dynamic,</li> <li>• “B” for Both time-dependent ROC curve estimations.</li> </ul>
xlab	a title for the x axis. The default value is "False - Positive Rate".
ylab	a title for the y axis. The default value is "True - Negative Rate".
type	what type of plot is going to be drawn. The default value is "l" and a line will be plotted.
lwd	line width. As a default value "5" is taken.
...	plot function arguments can also be used for customizing the plot.

## Details

Parameter tcr is mandatory with no default values. If a "B" is indicated and the [sROct](#) object placed as x parameter contains only one type of time-dependent ROC curve estimation, an error message will be returned. Another error message will appear in case of placing either "C" or "I" when the [sROct](#) object does not contain the suitable ROC curve estimation.

When one single type of ROC curve estimation is chosen, one graphic will be drawn for each point of time in the [sROct](#) object, having as many independent plots as number of points of time. Graphic parameters like axis labels or line width will be the same for all the plots.

In case of choosing both time-dependent ROC curve estimations, they will be plotted in a single graphic for each point of time in [sROct](#) object. As before, we will have as many independent plots as points of time and the graphic parameters will be the same in all plots.

## Examples

```

library(smoothROctime)
require(survival)

# Monoclonal Gammopathy of Undetermined Significance dataset
data(mgus)

# Time-to-event
time <- ifelse(is.na(mgus$pctime), mgus$futime,mgus$pctime)

# Status
status <- ifelse(is.na(mgus$pctime), 0, 1)

# Preparing data
DT <-as.data.frame(cbind(log(time), status, mgus$alb))
colnames(DT) <- c("futime", "pcm", "alb")
dta <- na.omit(cbind(DT$futime, DT$pcm, -DT$alb))

# Point of Time
t10 <- log(10*365.25) # ten years in logarithm scale

# Cumulative/Dynamic and Incident dynamic ROC curve estimations at t=10 years
rcu <- stRoc(data=dta, t=t10, tcr="B", meth = "1", verbose=TRUE)

# Plots of both ROC curve estimations
plot(rcu, tcr="B", frame=FALSE)

```

---

stRoc

*Smooth Time-dependent ROC curve estimations*


---

## Description

Provides smooth estimations of **Cumulative/Dynamic (C/D)** and **Incident/Dynamic (I/D)** ROC curves in presence of righth censorship and the corresponding Areas Under the Curves (AUCs), at a single point of time or a vector of points.

- The function computes two different procedures to obtain smooth estimations of the **C/D ROC curve**. Both are based on the kernel density estimation of the joint distribution function of the marker and time-to-event variables, provided by `funcen` function. The first method, to which we will refer as **smooth method**, is carried out according to the methodology proposed in [doi:10.1177/0962280217740786](https://doi.org/10.1177/0962280217740786). The second one uses this estimation of the joint density function of the variables marker and time-to-event for computing the weights or probabilities allocated to censored observations (undefined individuals) in [doi:10.1080/00949655.2016.1175442](https://doi.org/10.1080/00949655.2016.1175442) and [doi:10.1177/0962280216680239](https://doi.org/10.1177/0962280216680239). It will be referred as **p-kernel method**.
- In case of the **I/D ROC curve**, a smooth approximation procedure (**smooth method**) is computed based as well on the kernel density estimation of the joint distribution function of the marker and time-to-event variables proposed in [doi:10.1177/0962280217740786](https://doi.org/10.1177/0962280217740786)

**Usage**

```
stRoc(data, t, H, bw, adj, tcr, meth, ...)
```

**Arguments**

data	matrix of data values with three columns: time-to-event, censoring status (0=censored/1=uncensored) and marker.
t	point of time or vector of points where the time-dependent ROC curve is estimated.
H	2x2 bandwidth matrix.
bw	procedure for computing the bandwidth matrix. Most of the methods included at the kde function can be used: Hpi, Hpi.diag, Hlscv, Hlscv.diag, Hbcv, Hbcv.diag, Hscv, Hscv.diag, Hucv and Hucv.diag. Other considered methods are naive.pdf ( $\text{diag}(N^{-1/5}, N^{-1/5})^2$ ) and naive.cdf ( $\text{diag}(N^{-1/3}, N^{-1/3})^2$ ), where N is the sample size.
adj	adjustment parameter for calculating the bandwidth matrix. Default value 1.
tcr	type of time-dependent ROC curve estimation that will be estimated: <ul style="list-style-type: none"> <li>• “C” for <b>C</b>umulative/Dynamic,</li> <li>• “I” for <b>I</b>ncident/Dynamic,</li> <li>• “B” for <b>B</b>oth time-dependent ROC curve estimations.</li> </ul>
meth	method for computing the estimation of the C/D ROC curve. The suitable values are: <ul style="list-style-type: none"> <li>• “1” for the smooth method,</li> <li>• “2” for the p-kernel method.</li> </ul> <p>As default value the smooth method is taken.</p>
...	kde function arguments can be used for estimating the bivariate kernel density function.

**Details**

Function `funcen` is called from each execution of function `stRoc`, in order to compute the kernel density estimation of the joint distribution of the (Marker, Time-to-event) variable, therefore, the input parameters in `funcen` are input parameters as well in `stRoc` and the same considerations apply.

The matrix of bandwidths can be defined by using `H=matrix()` or automatically selected by the method indicated in `bw`.

Given the matrix of bandwidths, H, the argument `adj` modifies it and the final matrix is  $\text{adj}^2 H$ .

If H is missing, the naive.pdf method is used.

If `tcr` is missing the C/D ROC curve estimation will be computed with the method indicated in `meth`.

If no value has been placed in `meth` the smooth method will be used. The I/D ROC curve estimation will be always computed with the smooth method.

**Value**

An object of class `sROct` is returned. It is a list with the following values:

<code>th</code>	considered thresholds for the marker.
<code>FP</code>	false-positive rate calculated at each point in <code>th</code> .
<code>TP</code>	true-positive rate estimated at each point in <code>th</code> .
<code>p</code>	points where the time-dependent ROC curve is evaluated.
<code>R</code>	time-dependent ROC curve values computed at <code>p</code> .
<code>t</code>	time/s at which each time-dependent ROC curve estimation is computed. Each point of time will appear as many times as the length of the vector of points <code>p</code> .
<code>auc</code>	area under the corresponding time-dependent ROC curve estimation. As in the previous case, each value appears as many times as the length of the vector of points <code>p</code> .
<code>tcr</code>	type of time-dependent ROC curve estimation computed, <ul style="list-style-type: none"> <li>• “C” - Cumulative/Dynamic.</li> <li>• “I” - Incident/Dynamic.</li> </ul> <p>For each computed time-dependent ROC curve estimation this value is repeated as many times as the length of <code>p</code>.</p>
<code>Pi</code>	probabilities calculated for the individuals in the sample if the p-kernel method has been used for the estimation of the C/D ROC curve. This element is a matrix with the following columns: <ul style="list-style-type: none"> <li>• <code>time</code> - single point of time at which the estimation each the C/D ROC curve has been computed.</li> <li>• <code>obvt</code> - observed times for the individuals in the sample.</li> <li>• <code>p</code> - estimations of the probabilities computed and allocated to each subject.</li> </ul>

**References**

- P. Martinez-Cambor and J. C. Pardo-Fernandez. Smooth time-dependent receiver operating characteristic curve estimators. *Statistical Methods in Medical Research*, 27(3):651-674, 2018. doi:10.1177/0962280217740786.
- P. Martinez-Cambor, G. F-Bayn, and S. Perez-Fernandez. Cumulative/dynamic ROC curve estimation. *Journal of Statistical Computation and Simulation*, 86(17):3582-3594, 2016. doi:10.1080/00949655.2016.1175442.
- L. Li, T. Green, and B. Hu. A simple method to estimate the time-dependent receiver operating characteristic curve and the area under the curve with right censored data. *Statistical Methods in Medical Research*, 27(8), 2016. doi:10.1177/0962280216680239.
- T. Duong. Bandwidth matrices for multivariate kernel density estimation. Ph.D. Thesis, University of Western, Australia, 2004.

**Examples**

```
library(smoothROctime)
require(KMsurv)

data(kidtran)

# Preparing data: a logarithmic transformation of the time-to-event variable is made
DT <- cbind(log(kidtran$time),kidtran$delta,kidtran$age)

# Point of Time
t5 <- log(5*365.25) # five years in logarithm scale

# Cumulative/dynamic ROC curve estimation
rcd <- stRoc(data=DT, t=t5, bw="Hpi", tcr="C", meth=2)

# Plot graphic
plot(rcd$p, rcd$ROC, type="l", lwd=5, main="C/D ROC",xlab="FPR",ylab="TPR")
lines(c(0,1),c(0,1),lty=2,col="gray")
```

# Index

\* **smoothROctime**

smoothROctime-package, [2](#)

funcen, [2](#), [3](#), [6](#), [7](#)

plot.sROct, [2](#), [5](#)

smoothROctime (smoothROctime-package), [2](#)

smoothROctime-package, [2](#)

sROct, [5](#)

stRoc, [2](#), [5](#), [6](#)