Package 'muhaz'

February 20, 2015

Description A package for producing a smooth estimate of the hazard
function for censored data.

Version 1.2.6

Author S original by Kenneth Hess, <khess@odin.mdacc.tmc.edu> R port
by R. Gentleman

Title Hazard Function Estimation in Survival Analysis

Depends R(>= 2.3)

Imports survival

License GPL

Maintainer David Winsemius <dwinsemius@comcast.net>

Repository CRAN

Date/Publication 2014-08-09 19:19:51

NeedsCompilation yes

R topics documented:

kphaz.fit	*kphaz.fit*

Description

Calculates Kaplan-Meier type hazard estimates.

Usage

```
kphaz.fit(time,status,strata,q=1,method="nelson")
```

Arguments

time	A vector of time values; all values must be greater than or equal to zero. Missing values (NAs) are not allowed.
status	A vector of status values. The values are 0 for censored or 1 for uncensored (dead). Missing values (NAs) are not allowed. Must have the same length as time.
strata	An optional vector that will be used to divide the subjects into disjoint groups. Each group generates a hazard curve. If missing, all subjects are assumed to be in the same strata. Missing values (NAs) are allowed.
q	Number of failure times combined for estimatingthe hazard at their midpoint. Default is 1.
method	Type of hazard estimation made. Must be one of "nelson" or "product-limit". The default is "nelson".

Details

Let

$$t[1] < t[2] < \cdots < t[m]$$

denote the m "distinct" death times.

1. Estimate the cumulative hazard, H[t[j]], and the variance of the cumulative hazard, Var(H[t[j]]), at each of the m distinct death times according to the method selected.

a. For the "nelson" method:

$$H[t[j]] = sum(t[i] <= t[j])status[i]/(n - i + 1)$$

$$Var(H[t[j]]) = \sum (t[i] <= t[j])status[i]/((n - i + 1)^2)$$

b. For the "product-limit" metod:

$$H[t[j]] = sum(t[i] <= t[j]) - log(1 - status[i]/(n - i + 1))$$

$$Var(H[t[j]]) = sum(t[i] <= t[j])status[i]/((n - i + 1) * (n - i))$$

2. For k=1,...,(m-q), define the hazard estimate and variance at time[k] = (t[q+j]+t[j])/2 to be

$$haz[time[k]] = (H[t[q+j]] - H[t[j]])/(t[q+j] - t[j])$$

$$var[time[k]] = (Var(H[t[q+j]]) - Var(H[t[j]]))/(t[q+j] - t[j])^2$$

Note that if the final time is a death time rather than a censoring time, the "product-limit" estimate will be Inf for the final hazard and variance estimates.

Value

A list representing the results of the hazard estimation, with the following components:

time	A vector containing the times at which hazard estimations were made.
haz	A vector containing the hazard estimate at each time.
var	A vector containing variance estimates for each hazard estimate.
strata	A vector which divides the hazard estimate into disjoint groups. This vector is returned only if 'strata' is defined when 'kphaz.fit' is called.

References

Jarjoura, David (1988). Smoothing Hazard Rates with Cubic Splines. *Commun. Statist. -Simula.* **17(2)**, 377-392.

See Also

[kphaz.plot](#)

Examples

```
time <- 1:10
status <- rep(1,10)
kphaz.fit(time,status)
```

---

kphaz.plot	*kphaz.plot*

Description

Plots a Kaplan-Meier type hazard estimate.

Usage

```
kphaz.plot(fit, ...)
```

Arguments

fit	A list representing the results of a call to "kphaz.fit".
...	Any legal argument for the plot function.

Side Effects

A plot with multiple hazard curves. One for each unique strata with 1 or more point.

See Also

kphaz.fit

Examples

```
# Use "kphaz.fit" to generate a hazard estimate
data(ovarian, package="survival")
attach(ovarian)
kpfit <- kphaz.fit(futime, fustat)
# Use "kphaz.plot" to plot the estimate
kphaz.plot(kpfit)
```

---

muhaz   *Estimate hazard function from right-censored data.*

Description

Estimates the hazard function from right-censored data using kernel-based methods. Options include three types of bandwidth functions, three types of boundary correction, and four shapes for the kernel function. Uses the global and local bandwidth selection algorithms and the boundary kernel formulations described in Mueller and Wang (1994). The nearest neighbor bandwidth formulation is based on that described in Gefeller and Dette (1992). The statistical properties of many of these estimators are reported and compared in Hess et al (1999). Based on the HADES program developed by H.G. Mueller. Returns an object of class 'muhaz.' NOTE: For comparison to the smoothed hazard function estimates, we have also made available a set of functions based on piecewise exponential estimation. These estimates are similar in concept to the histogram estimator of the density function. They give a feel for the features of the data without the manipulations involved in smoothing. They also help to confirm that muhaz is generating realistic estimates of the underlying hazard function. These functions are called: pehaz, plot.pehaz, lines.pehaz, print.pehaz.

Usage

```
muhaz(times, delta, subset, min.time, max.time, bw.grid, bw.pilot,
      bw.smooth, bw.method="local", b.cor="both", n.min.grid=51,
      n.est.grid=101, kern="epanechnikov")
```

Arguments

times	Vector of survival times. Does not need to be sorted.
delta	Vector indicating censoring 0 - censored (alive) 1 - uncensored (dead) If delta is missing, all the observations are assumed uncensored.
subset	Logical vector, indicating the observations used in analysis. T - observation is used F - observation is not used If missing, all the observations will be used.

min.time	Left bound of the time domain used in analysis. If missing, min.time is considered 0.
max.time	Right bound of the time domain used in analysis. If missing, max.time is the time at which ten patients remain at risk.
bw.grid	Bandwidth grid used in the MSE minimization. If bw.method="global" and bw.grid has one component only, no MSE minimization is performed. The hazard estimates are computed for the value of bw.grid. If bw.grid is missing, then a bandwidth grid of 21 components is built, having as bounds:

$$[0.2 * bw.pilot, 20 * bw.pilot]$$

bw.pilot	Pilot bandwidth used in the MSE minimization. If missing, the default value is the one recommended by Mueller and Wang (1994):

$$bw.pilot = (max.time - min.time)/(8 * nz^0.2)$$

where nz is the number of uncensored observations

bw.smooth	Bandwidth used in smoothing the local bandwidths. Not used if bw.method="global" If missing:

$$bw.smooth = 5 * bw.pilot$$

bw.method	Algorithm to be used. Possible values are: "global" - same bandwidth for all grid points. The optimal bandwidth is obtained by minimizing the IMSE. "local" - different bandwidths at each grid point. The optimal bandwidth at a grid point is obtained by minimizing the local MSE. "knn" - k nearest neighbors distance bandwidth. The optimal number of neighbors is obtained by minimizing the IMSE. Default value is "local". Only the first letter needs to be given (e.g. "g", instead of "global").
b.cor	Boundary correction type. Possible values are: "none" - no boundary correction "left" - left only correction "both" - left and right corrections Default value is "both". Only the first letter needs to be given (e.g. b.cor="n").
n.min.grid	Number of points in the minimization grid. This value greatly influences the computing time. Default value is 51.
n.est.grid	Number of points in the estimation grid, where hazard estimates are computed. Default value is 101.
kern	Boundary kernel function to be used. Possible values are: "rectangle", "epanechnikov", "biquadratic", "triquadratic". Default value is "epanechnikov". Only the first letter needs to be given (e.g. kern="b").

Details

The muhaz object contains a list of the input data and parameter values as well as a variety of output data. The hazard function estimate is contained in the haz.est element and the corresponding time points are in est.grid. The unsmoothed local bandwidths are in bw.loc and the smoothed local bandwidths are in bw.loc.sm.

For bw.method = 'local' or 'knn', to check the shape of the bandwidth function used in the estimation, use plot(fitpinmin.grid, fit$bw.loc) to plot the unsmoothed bandwidths and use

`lines(fit$est.grid, fit$bw.loc.sm)` to superimpose the smoothed bandwidth function. Use bw.smooth to change the amount of smoothing used on the bandwidth function.

For bw.method='global', to check the minimization process, plot the estimated IMSE values over the bandwidth search grid. Use `plot(fit$bw.grid, fit$globlmse)`. Use k.grid and k.imse for bw.method='k'. You may want to repeat the search using a finer grid over a shorter interval to fine-tune the optimization or if the observed minimum is at the extreme of the grid you should specify a different grid.

Value

Returns an object of class 'muhaz', containing input and output values. Methods working on such an object are: plot, lines, summary. For a detailed description of its components, see object.muhaz.

References

1. H.G. Mueller and J.L. Wang - Hazard Rates Estimation Under Random Censoring with Varying Kernels and Bandwidths, Biometrics 50, 61-76, March 1994

2. O. Gefeller and H. Dette - Nearest Neighbour Kernel Estimation of the Hazard Function From Censored Data, J. Statist. Comput. Simul., Vol.43, 1992, 93-101

3. K.R. Hess, D.M. Serachitopol and B.W. Brown - Hazard Function Estimators: A Simulation Study, Statistics in Medicine (in press).

See Also

summary.muhaz, plot.muhaz, lines.muhaz, muhaz.object

Examples

```
# to compute a locally optimal estimate
data(ovarian, package="survival")
attach(ovarian)
fit1 <- muhaz(futime, fustat)
plot(fit1)
summary(fit1)
# to compute a globally optimal estimate
fit2 <- muhaz(futime, fustat, bw.method="g")
# to compute an estimate with global bandwidth set to 5
fit3 <- muhaz(futime, fustat, bw.method="g", bw.grid=5)
```

---

muhaz.object   *Estimated Hazard Rate Object*

Description

This class of objects is returned by `muhaz`, which estimates the hazard function from censored data.

Arguments

pin.common	list containing the input parameters. Its components are: * times - the survival times vector * delta - the censoring vector * nobs - the number of observations * min.time - the minimum time used in analysis * max.time - the maximum time used in analysis * n.min.grid - number of points in the minimization grid * min.grid - the minimization grid * n.est.grid - number of points in the estimation grid * bw.pilot - the pilot bandwidth * bw.smooth - the smoothing bandwidth for the local optimal bandwidths * bw.method - the method used to estimate the hazard rates * b.cor - boundary correction used * kernel.type - kernel function used in the smoothing
est.grid	the estimation grid where the hazard rates are computed.
haz.est	the hazard estimates calculated at the estimation grid points.
imse.opt	IMSE for the optimal bandwidth.
bw.glob	optimal global bandwidth. For bw.method="global" only.
glob.imse	vector of IMSE, computed at each point in bw.grid. For bw.method="global" only.
bw.grid	Grid of bandwidth values used in the minimization. For bw.method="global" or "local".
bw.loc	vector of optimal local bandwidths computed by minimizing the MSE at each point in the minimization grid. Not used for bw.method="local".
bw.loc.sm	vector of smoothed local bandwidths, computed at each point in the estimation grid by smoothing bw.loc using bw.smooth as the smoothing bandwidth. Not used for bw.method="global".
bias.min	vector of minimized bias, computed at each point in the minimization grid for the optimal local bandwidth. For bw.method="local" only.
var.min	vector of minimized variance, computed at each point in the minimization grid for the optimal local bandwidth. For bw.method="local" only.
k.grid	grid of nearest neighbor numbers used in the minimization. For bw.method="knn" only.
k.imse	vector of IMSE, computed at each of the points of k.grid. For bw.method="knn" only.
k.opt	optimum number of nearest neighbors. For bw.method="knn" only.

METHODS

Objects of this class have methods for the functions summary, plot, and lines.

STRUCTURE

Common components of a muhaz object:

See Also

[muhaz](), [plot.muhaz](), [summary.muhaz]().

---

pehaz	*Estimates piecewise exponential hazard function from right-censored data.*

Description

Divides the time domain into bins of equal width, and then estimates the hazard in each bin as the number of events in that bin divided by the total follow-up time in that bin.

Usage

```
pehaz(times, delta=NA, width=NA, min.time=0, max.time=NA)
```

Arguments

times	Vector of survival times. Does not need to be sorted.
delta	Vector indicating censoring 0 - censored (alive) 1 - uncensored (dead) If status is missing, all observations are assumed uncensored.
width	Bin width. Default value is that recommended by Mueller, $width = (max.time - min.time)/(8 * (nu)^0.2)$ where nu is the number of uncensored observations.
min.time	Left bound of the time domain used in the analysis. If missing, min.time is considered 0.
max.time	Right bound of the time domain used in the analysis. If missing, max.time is considered max(times).

Value

Returns an object of class 'pehaz', containing input and output values. Methods working on such an object are: plot, lines, print. For a detailed description of its components, see object.pehaz.

See Also

pehaz.object

Examples

```
data(ovarian, package="survival")
attach(ovarian)
fit <- pehaz(futime, fustat)
plot(fit)
```

`pehaz.object`	*Estimated Piecewise Exponential Hazard Rate Object*

Description

This class of objects is returned by `pehaz`, which estimates the hazard function from censored data.

METHODS

Objects of this class have methods for the functions `summary`, `plot`, and `lines`.

STRUCTURE

Common components of a `pehaz` object:

A list containing the following components:

call the call to pehaz

Width the width of the bins

Cuts the cutpoints used for the bins

Hazard the estimated hazard for each bin

Events the number of events in each bin

At.Risk the number at risk in each bin

F.U.Time the followup time (for the bin?)

See Also

pehaz

`plot.muhaz`	*Plots estimated hazard function from an object of class muhaz.*

Description

Default time limits are those provided to `muhaz`, which default to zero and the time corresponding to when ten patients remain at risk. Default y-axis limits are 0 and the maximum estimated hazard rate. Additional lines can be added to the same set of axes using the `lines` method.

Usage

```
## S3 method for class 'muhaz'
plot(x, ylim, type, xlab, ylab, ...)
## S3 method for class 'muhaz'
lines(x, ...)
```

Arguments

x	Object of class `muhaz` (output from calling muhaz function)
ylim	Limits for the y axis.
type	type argument for `plot`.
xlab	Label for the x axis.
ylab	Label for the y axis.
...	Additional arguments to be passed along.

See Also

muhaz

---

plot.pehaz   *Plot a pehaz object.*

Description

A plot of the pehaz object is produced on the current device. If `lines.pehaz` was called then the estimated curve is added to the current plot.

Usage

```
## S3 method for class 'pehaz'
plot(x, xlab="Time", ylab="Hazard Rate", ...)
## S3 method for class 'pehaz'
lines(x, lty=2, ...)
```

Arguments

x	A pehaz object.
xlab	The x-axis label.
ylab	The y-axis label.
lty	The line type to use when plotting.
...	Other graphical parameters, passed to `plot`

Value

No value is returned, the object is plotted on the active device.

See Also

pehaz.object

Examples

```
data(ovarian, package="survival")
attach(ovarian)
fit <- pehaz(futime, fustat)
plot(fit)
```

---

print.pehaz   *Print a pehaz object.*

Description

The pehaz object is printed.

Usage

```
## S3 method for class 'pehaz'
print(x, ...)
```

Arguments

x   An object of class pehaz.

...   Ignored.

Value

No value is returned. x is printed.

See Also

pehaz.object

---

summary.muhaz   *Display the most important input parameters used in calling the 'muhaz' function.*

Description

It also displays some of the output data. Common to all three methods: * number of observations * number of censored observations * bandwidth method used (global, local or nearest neighbor) * boundary correction type (none, left only, both left and right) * kernel type (rectangle, Epanechnikov, biquadradic, triquadratic) * minimum time * maximum time * number of points in MSE minimization grid * number of points in estimation grid * pilot bandwidth * estimated IMSE for optimal bandwidth For bw.method="global" also reports optimal global bandwidth. For bw.method="knn" also reports optimal number of nearest neighbors. For bw.method="local" and bw.method="knn" also reports smoothing bandwidth used to smooth the optimal local bandwidths.

Usage

```
## S3 method for class 'muhaz'
summary(object, ...)
```

Arguments

object	Object of class muhaz (output from calling muhaz function)
...	Ignored for now.

See Also

[muhaz](#), [muhaz.object](#)

Examples

```
data(ovarian, package="survival")
attach(ovarian)
fit <- muhaz(futime, fustat)
summary(fit)
```

Index