

Package ‘survrec’

July 2, 2014

Version 1.2-2

Date 2012-Mar-26

Depends R (>= 2.12.0), boot

Title Survival analysis for recurrent event data

Author Juan R González, Edsel A Peña, and Robert L Strawderman

Maintainer Juan R González <jrgonzalez@creal.cat>

Description Estimation of survival function for recurrent event data using Peña-Strawderman-Hollander, Whang-Chang estimators and MLE estimation under a Gamma Frailty model.

License GPL (>= 2)

URL <http://www.r-project.org>

Encoding latin1

Repository CRAN

Date/Publication 2013-12-13 07:46:53

NeedsCompilation yes

R topics documented:

colon	2
mlefrailty.fit	2
MMC	5
plot.survfitr	5
print.survfitr	6
psh.fit	7
q.search	8
summary.survfitr	9
surv.search	10

survdiff	11
survfitr	12
Surv	14
wc.fit	15

Index	17
--------------	-----------

colon	<i>Rehospitalization colorectal cancer</i>
-------	--

Description

This contains rehospitalization times after surgery in patients with colorectal cancer

Usage

```
data(colon)
```

Format

This data frame contains the following columns:

hc identifier of each subject. Repeated for each recurrence

time rehospitalization o censoring time

event censoring status. All event are 1 for each subject excepting last one that it is 0

chemoter Did patient receive chemotherapy? 1: No 2:Yes

dukes Dukes' tumoral stage: 1:A-B 2:C 3:D

distance distance from living place to hospital 1:<=30 Km. 2:>30 Km.

Source

González, JR., Fernandez, E., Moreno, V. et al. Gender differences in hospital readmission among colorectal cancer patients. Currently submitted to JCO

mlefrailty.fit	<i>Survival function estimator for correlated recurrence time data under a Gamma Frailty Model</i>
----------------	--

Description

Estimation of survival function for correlated recurrence time data under a Gamma Frailty model using the maximum likelihood criterion. The resulting object of class "survfitr" is plotted by 'plot.survfitr', before it is returned.

Usage

```
mlefrailty.fit(x,tvals, lambda=NULL, alpha=NULL, alpha.min, alpha.max,
tol=1e-07, maxiter=500,alpha.console=TRUE)
```

Arguments

x	a survival recurrent event object.
tvals	vector of times where the survival function can be estimated.
lambda	optional vector of baseline hazard probabilities at t (see details). Default is numdeaths/apply(AtRisk,2,sum).
alpha	optional parameter of shape and scale for the frailty distribution. If this parameter is unknown is estimate via EM algorithm. In order to obtain the convergence of this algorithm a seed is calculated (see details).
alpha.min	optional left bound of the alpha parameter in order to obtain a seed to estimate alpha parameter. Default value is 0.5.
alpha.max	optional right bound of the alpha parameter in order to obtain a seed to estimate alpha parameter. Default value is the maximum of distinct times of events.
tol	optional tolerance of the EM algorithm used to estimate the alpha parameter. Default is 10e-7
maxiter	optional maximum number of iterations of the EM algorithm used to estimate the alpha parameter. Default is 500.
alpha.console	if TRUE prints in the console the estimates initial value for alpha and the alpha estimate via the EM algorithm, if FALSE not.

Details

The product limit estimator developed by Peña, Strawderman and Hollander (2001) are valid when the interoccurrence times are assumed to represent an IID sample from some underlying distribution F. This assumption is clearly restrictive in biomedical applications, and one obvious generalization that allows association between interoccurrence times is a frailty model.

A common and convenient choice of frailty distribution is a gamma distribution with shape and scale parameters set equal to an unknown parameter α . The common marginal survival function can be written as following

$$\bar{F}(t) = \left[\frac{\alpha}{\alpha + \Lambda_0(t)} \right]^\alpha$$

The parameter α controls the degree of association between interoccurrence times within a unit. Peña, Strawderman and Hollander (2001) showed that the estimation of α and Λ_0 can be obtained via the maximisation of the marginal likelihood function and the expectation-maximisation (EM) algorithm. For details and the theory behind this estimator, please refer to Peña, Strawderman and Hollander (2001, JASA).

In order to obtain a good convergence, α is estimated previously. This estimation is used as a initial value in the EM procedure and it's carried out by the maximisation of the profile likelihood for α . In this case the arguments of **mlefrailty.fit** function called alpha.min and alpha.max are the boundaries of this maximisation. The maximum is obtained using the golden section search method.

Value

If the convergence of EM algorithm is not obtained, the initial value of alpha can be used as a alpha.min argument and recalculate.

n	number of unit or subjects observed.
m	vector of number of recurrences in each subject (length n).
failed	vector of number of recurrences in each subject (length n*m). Vector ordered (e.g. times of first unit, times of second unit , ..., times of n-unit).
censored	vector of times of censorship for each subject (length n).
numdistinct	number of distinct failures times.
distinct	vector of distinct failures times.
status	0 if the estimation is can be provided and 1 if not depending if alpha could be estimate or not.
alpha	parameter of Gamma Frailty Model.
lambda	Estimates of the hazard probabilities at distinct failures times.
survfunc	vector of survival estimated in distinct times.
tvals	copy of argument.
MLEAttvals	vector of survival estimated in tvals times.

References

Peña, E.A., Strawderman, R. and Hollander, M. (2001). Nonparametric Estimation with Recurrent Event Data. *J. Amer. Statist. Assoc* **96**, 1299-1315.

See Also

[survfitr](#) [Surv](#)

Examples

```
data(MMC)
fit<-mlefrailty.fit(Surv(MMC$id,MMC$time,MMC$event))
fit
plot(fit)

# compare with pena-straderman-hollander

fit<-psh.fit(Surv(MMC$id,MMC$time,MMC$event))
fit
lines(fit,lty=2)

# and with wang-chang

fit<-wc.fit(Surv(MMC$id,MMC$time,MMC$event))
fit
lines(fit,lty=3)
```

MMC *Migratory Motor Complex*

Description

This contains the Migratory Motor Complex data

Usage

```
data(MMC)
```

Format

This data frame contains the following columns:

id ID of each subject. Repeated for each recurrence

time recurrence or censoring time

event censoring status. All events are 1 for each subject excepting last one that it is 0

group a factor with levels

Note: The groups have been created (at random) to illustrate a group comparison

Source

Husebye E, Skar V, Aalen O and Osnes M (1990), Digestive Diseases and Sciences, p1057

plot.survfitr *Plots estimated survival function from an object of class 'survrec'.*

Description

Additional plots can be added to the same axes using 'lines.survrec'.

Usage

```
## S3 method for class 'survfitr'
plot(x, conf.int=TRUE, prob = FALSE, ...)
```

Arguments

x	Object of class survrec (output from calling survrec function).
conf.int	Print the pointwise confidence intervals of the probability or survival function if its value is TRUE or FALSE.
prob	Print of the probability or survival function if its value is TRUE or FALSE respectively.
...	additional arguments passed to the plot function.

Value

Print a plot of class survrec

See Also

[psh.fit](#) [mlefrailty.fit](#) [wc.fit](#)

`print.survfitr`

Print a Short Summary of a Survival Recurrent Curve

Description

Print number of observations, number of events, the restricted mean survival and its standard error, the median survival and the minimum, maximum and median number of recurrences for each subject.

Usage

```
## S3 method for class 'survfitr'
print(x, scale=1, digits=max(options())$digits - 4, 3), ...)
```

Arguments

<code>x</code>	the result of a call to the <code>survfit</code> , <code>psh.fit</code> , <code>wc.fil</code> or <code>mlefrailty.fit</code> functions
<code>scale</code>	a numeric value to rescale the survival time, e.g., if the input data to <code>survfit</code> were in days, <code>scale=365</code> would scale the printout to years
<code>digits</code>	number of digits to print
<code>...</code>	other unused arguments

Details

The restricted mean and its standard error are based on a truncated estimator. If the last observation(s) is not a death, then the survival curve estimate does not go to zero and the mean survival time cannot be estimated. Instead, the quantity reported is the mean of survival restricted to the time before the last censoring. When the last censoring time is not random this quantity is occasionally of interest.

The median are defined by drawing a horizontal line at 0.5 on the plot of the survival curve.

Value

`x`, with the invisible flag set.

The number of observations, the number of events, the restricted mean survival and its standard error, the median survival and the minimum, maximum and median number of recurrences are printed. If there are multiple curves, there is one line of output for each.

See Also

[summary.survfitr](#), [survfitr](#)

Examples

```
data(MMC)
fit<-survfitr(Survr(id,time,event)~group,data=MMC)
print(fit)
```

psh.fit	<i>Survival function estimator for recurrence time data using the estimator NA</i>
---------	--

Description

Estimation of survival function for recurrence time data by means the generalized product limit estimator (PLE) method developed by Pe a, Strawderman and Hollander. The resulting object of class "survfitr" is plotted by 'plot.survfitr', before it is returned.

Usage

```
psh.fit(x,tvals)
```

Arguments

x	a survival recurrent event object
tvals	vector of times where the survival function can be estimated.

Details

The estimator computed by this object is the nonparametric estimator of the inter-event time survivor function under the assumption of a renewal or IID model. This generalizes the product-limit estimator to the situation where the event is recurrent. For details and the theory behind this estimator, please refer to Pe\~na, Strawderman and Hollander (2001, JASA).

Value

n	number of unit or subjects observed.
m	vector of number of recurrences in each subject (length n).
failed	vector of number of recurrences in each subject (length n*m). Vector ordered (e.g. times of first unit, times of second unit, ..., times of n-unit)
censored	vector of times of censorship for each subject (length n).
numdistinct	number of distinct failures times
distinct	vector of distinct failures times
AtRisk	matrix of number of persons-at-risk at each distinct time and for each subject

survfunc vector of survival estimated in distinct times
 tvals copy of argument
 PSHpleAttvals vector of survival estimated in tvals times

References

Peña, E.A., Strawderman, R. and Hollander, M. (2001). Nonparametric Estimation with Recurrent Event Data. *J. Amer. Statist. Assoc* **96**, 1299-1315.

See Also

[survfitr](#) [Surv](#)

Examples

```
data(MMC)
fit<-psh.fit(Surv(MMC$id,MMC$time,MMC$event))
fit
plot(fit,conf.int=FALSE)

# compare with MLE Frailty

fit<-mlefrailty.fit(Surv(MMC$id,MMC$time,MMC$event))
fit
lines(fit,lty=2)

# and with wang-chang

fit<-wc.fit(Surv(MMC$id,MMC$time,MMC$event))
fit
lines(fit,lty=3)
```

q.search

Calculate the survival time of a selected quantile

Description

Auxiliary function called from survdiff function. Given a survfit object we obtain the quantile from a survival function

Usage

```
q.search(f, q = 0.5)
```


Arguments

f survdifr object
 q quantile. Default is 0.5

Value

Returns the time in a selected quantile

Examples

```
data(MMC)
fit<-survfitr(Survr(id,time,event)~1,data=MMC)

# 75th percentile from the survival function
q.search(fit,q=0.75)
```

summary.survfitr *Summary of a Survival of Recurrences Curve*

Description

Returns a matrix containing the survival curve and other information. If there are multiple curves, returns a list that contains the previous matrix for each curve.

Usage

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

Arguments

object output from a call to survfitr, psh.fit, wc.fit or mlefrailty.fit.
 ... other unused arguments.

Value

For one survival curve returns a matrix, and for multiple curves a list with the same matrix for each curve. This matrix contains the distinct failure times, and the number of events, at risk subjects, survival and standard error for each distinct time

See Also

[survfitr](#)

Examples

```
data(MMC)
summary(survfitr(Survr(id,time,event)~group,data=MMC))
```

surv.search

Calculate the survival in selected times

Description

Auxiliary function called from pshPLE, wcPLE and MLEFrailty.

The estimation using PLE (e.g. Kaplan-Meier) is a decreasing constant piecewise function with jumps in the times with events. Thus, to estimate the survival at any time we take the time of the previous event.

Usage

```
surv.search(tvals,time,surv)
```

Arguments

tvals	vector of times where the survival function has to be estimated
time	vector of failures times (distinct)
surv	vector of survival of each time

Value

Returns the survival in each selected time (tvals) from a vector of survival values

Examples

```
# we have the times 4,7,9,15,21,67
time<-c(4,7,9,15,21,67)

# and its survival (note: in this example there may be more
# than one event in some times)
surv<-c(0.8,0.7,0.65,0.55,0.43,0.22)

# We want to calculate the survival at times 1, 10, 32,64
surv.search(c(1,10,32,74),time,surv)
```

survdiff *Test median survival differences (or other quantile)*

Description

Obtain bootstrap replicates of the median survival time for different groups of subjects. We can compute confidence intervals using boot package.

Usage

```
survdiff(formula, data, q, B = 500, boot.F = "WC", boot.G = "none", ...)
```

Arguments

formula	A formula object. If a formula object is supplied it must have a Survr object as the response on the left of the ~ operator and a term on the right. For a single bootstrap median survival the "~1" part of the formula is required.
data	A data frame in wich to interpret the variables named in the formula.
q	Quantile that we are interested in to obtain a bootstrap sample from survival function
B	Number of bootstrap samples
boot.F	a character string specifying the bootstrap procedure. Possible value are either "PSH" or "WC" for nonparametric bootstrap or "semiparametric" for semiparametric bootstrap. The default is "WC". Only the first words are required, e.g "P", "W", "se"
boot.G	a character string specifying if we also resample form censored empirical distribution. Possible value are either "none" or "empirical". The default is "none". Only the first words are required, e.g "n", "e"
...	additional arguments passed to the type of estimator.

Details

See reference. Some procedures can be slow

Value

A boot object. Bootstrap confidence intervals can be computed using boot.ci function from boot package

References

Gonzalez JR, Peña EA. Bootstraping median survival with recurrent event data. IX Conferencia Española de Biometría; 2003 May 28-30; A Coruña, España.

Paper available upon request to the mantainer

See Also

[survfitr](#), [boot.ci](#)

Examples

```
data(colon)

#We will compare the median survival time for three dukes stages
fit<-survdiff(Surv(hc,time,event)~as.factor(dukes),data=colon,q=0.5)
boot.ci(fit$"1", type=c("norm","basic", "perc"))
boot.ci(fit$"2", type=c("norm","basic", "perc"))
boot.ci(fit$"3", type=c("norm","basic", "perc"))

# 75th quantile of survival function
fit<-survdiff(Surv(hc,time,event)~as.factor(dukes),data=colon,q=0.75)
# bootstrap percentile confidence interval
quantile(fit$"1"$t,c(0.025,0.975))
quantile(fit$"2"$t,c(0.025,0.975))
quantile(fit$"3"$t,c(0.025,0.975))

# We could execute this if there is none Inf value
# boot.ci(fit$"1")
# boot.ci(fit$"2")
# boot.ci(fit$"3")

# We can modify the bootstrap procedure modifying boot.F parameter
fit<-survdiff(Surv(hc,time,event)~as.factor(dukes),data=colon,q=0.5,boot.F="PSH")
# bootstrap percentile confidence interval
quantile(fit$"1"$t,c(0.025,0.975))
quantile(fit$"2"$t,c(0.025,0.975))
quantile(fit$"3"$t,c(0.025,0.975))
```

survfitr

Compute a Survival Curve for Recurrent Event Data given a covariate

Description

Computes an estimate of a survival curve for recurrent event data using either the Peña-Strawderman-Hollander, Wang-Chang or MLE Frailty estimators. It also computes the asymptotic standard errors. The resulting object of class "survfitr" is plotted by 'plot.survfitr', before it is returned.

Usage

```
survfitr(formula, data, type="MLEfrailty",...)
```

Arguments

formula	A formula object. If a formula object is supplied it must have a Survr object as the response on the left of the ~ operator and a term on the right. For a single survival curve the "~1" part of the formula is required.
data	a data frame in wich to interpret the variables named in the formula.
type	a character string specifying the type of survival curve. Possible value are "pena-strawderman-hollander", "wang-chang" or "MLEfrailty". The default is "MLEfrailty". Only the first words are required, e.g "pe","wa","ML"
...	additional arguments passed to the type of estimator.

Details

See the help details of psh.fit, wc.fit or mlefrailty depending on the type chosen

Value

a survfitr object. Methods defined for survfitr objects are provided for print,plot,lines and summary.

Note

The mantainer wishes to thank Professors Chiung-Yu Huang and Shu-Hui Chang for their help for providing us with the Fortran code which computes standard errors of Wang and Chang's estimator.

References

1. Peña, E.A., Strawderman, R. and Hollander, M. (2001). Nonparametric Estimation with Recurrent Event Data. *J. Amer. Statist. Assoc* **96**, 1299-1315.
2. Wang, M.-C. and Chang, S.-H. (1999). Nonparametric Estimation of a Recurrent Survival Function. *J. Amer. Statist. Assoc* **94**, 146-153.

See Also

[print.survfitr](#),[plot.survfitr](#),[lines.survfitr](#),[summary.survfitr](#),[Survr.psh.fit](#),[wc.fit](#),[mlefrailty.fit](#)

Examples

```
data(colon)
# fit a pena-strawderman-hollander and plot it
fit<-survfitr(Survr(hc,time,event)~as.factor(dukes),data=colon,type="pena")
plot(fit,ylim=c(0,1),xlim=c(0,2000))
# print the survival estimators
fit
summary(fit)

# fit a MLE Frailty and plot it (in this case do not show s.e.)
fit<-survfitr(Survr(hc,time,event)~as.factor(dukes),data=colon,type="MLE")
plot(fit)
# print the survival estimators
```

```
fit
summary(fit)
```

Survr

Create a Survival recurrent object

Description

Create a survival recurrent object, usually used as a response variable in a model formula

Usage

```
Survr(id, time, event)
is.Survr(x)
```

Arguments

id	Identificator of each subject. This value is the same for all recurrent times of each subject.
time	time of recurrence. For each subject the last time are censored.
event	The status indicator, 0=no recurrence 1=recurrence. Only this values are accepted.
x	any R object.

Value

An object of class Survr. Survr objects are implemented as a matrix of 3 columns. No method for print.

In the case of is.Survr, a logical value T if x inherits from class "Survr", otherwise an F.

See Also

[survfitr](#), [psh.fit](#), [wc.fit](#), [mlefrailty.fit](#)

Examples

```
data(MMC)
Survr(MMC$id, MMC$time, MMC$event)
```

wc.fit	<i>Survival function estimator for recurrence time data using the estimator developed by Wang and Chang.</i>
--------	--

Description

Estimation of survival function for correlated or i.i.d. recurrence time data by means of the product limit estimator (PLE) method developed by Wang and Chang. The resulting object of class "survfitr" is plotted by 'plot.survfitr', before it is returned.

Usage

```
wc.fit(x, tvals)
```

Arguments

x	a survival recurrent event object.
tvals	vector of times where the survival function can be estimated.

Details

Wang and Chang (1999) proposed an estimator of the common marginal survivor function in the case where within-unit interoccurrence times are correlated. The correlation structure considered by Wang and Chang (1999) is quite general and contains, in particular, both the i.i.d. and multiplicative (hence gamma) frailty model as special cases.

This estimator removes the bias noted for the product-limit estimator developed by Pea, Strawderman and Hollander (PSH, 2001) when interoccurrence times are correlated within units. However, when applied to i.i.d. interoccurrence times, this estimator is not expected to perform as well as the PSH estimator, especially with regard to efficiency.

Value

n	number of unit or subjects observed.
m	vector of number of recurrences in each subject (length n).
failed	vector of number of recurrences in each subject (length n*m). Vector ordered (e.g. times of first unit, times of second unit, ..., times of n-unit)
censored	vector of times of censorship for each subject (length n).
numdistinct	number of distinct failures times.
distinct	vector of distinct failures times.
AtRisk	matrix of number of persons-at-risk at each distinct time and for each subject.
survfunc	vector of survival estimated in distinct times.
tvals	copy of argument.
PSHpleAttvals	vector of survival estimated in tvals times.

Note

The maintainer wishes to thank Professors Chiung-Yu Huang and Shu-Hui Chang for their help for providing us with the Fortran code which computes standard errors of Wang and Chang's estimator.

References

Wang, M.-C. and Chang, S.-H. (1999). Nonparametric Estimation of a Recurrent Survival Function. *J. Amer. Statist. Assoc* **94**, 146-153.

See Also

[survfitr](#) [Surv](#)

Examples

```
data(MMC)

fit<-wc.fit(Surv(MMC$id,MMC$time,MMC$event))
fit
plot(fit,conf.int=FALSE)

# compare with pena-straderman-hollander

fit<-psh.fit(Surv(MMC$id,MMC$time,MMC$event))
fit
lines(fit,lty=2)

# and with MLE frailty

fit<-mlefrailty.fit(Surv(MMC$id,MMC$time,MMC$event))
fit
lines(fit,lty=3)
```


Index

*Topic **datasets**

colon, [2](#)

MMC, [5](#)

*Topic **survival**

mlefrailty.fit, [2](#)

plot.survfitr, [5](#)

print.survfitr, [6](#)

psh.fit, [7](#)

q.search, [8](#)

summary.survfitr, [9](#)

surv.search, [10](#)

survdiffrr, [11](#)

survfitr, [12](#)

Survrr, [14](#)

wc.fit, [15](#)

boot.ci, [12](#)

colon, [2](#)

is.Survrr (Survrr), [14](#)

lines.survfitr, [13](#)

lines.survfitr (plot.survfitr), [5](#)

mlefrailty.fit, [2](#), [6](#), [13](#), [14](#)

MMC, [5](#)

plot.survfitr, [5](#), [13](#)

print.summary.survfitr

(summary.survfitr), [9](#)

print.survfitr, [6](#), [13](#)

psh.fit, [6](#), [7](#), [13](#), [14](#)

q.search, [8](#)

summary.survfitr, [7](#), [9](#), [13](#)

surv.search, [10](#)

survdiffrr, [11](#)

survfitr, [4](#), [7-9](#), [12](#), [12](#), [14](#), [16](#)

Survrr, [4](#), [8](#), [13](#), [14](#), [16](#)

wc.fit, [6](#), [13](#), [14](#), [15](#)