Package: survrec (via r-universe)

October 22, 2026

Type Package

Version 2.0.0

Date 2026-09-12

Title Survival Analysis for Recurrent Event Data

Description Estimation of the survival function of inter-occurrence times for recurrent event data, using the estimators of Peña, Strawderman and Hollander (2001) <doi:10.1198/016214501753381922> and Wang and Chang (1999) <doi:10.1080/01621459.1999.10473831>, and maximum likelihood estimation under a gamma frailty model. Includes bootstrap comparison of survival quantiles between groups.

Depends R (>= 3.5.0)

Imports Rcpp, boot, ggplot2, graphics, grDevices, rlang, stats, utils

LinkingTo Rcpp

Suggests testthat (>= 3.0.0), knitr, rmarkdown, BiocStyle

VignetteBuilder knitr

Config/testthat/edition 3

License GPL (>= 2)

URL https://github.com/isglobal-brge/survrec

BugReports https://github.com/isglobal-brge/survrec/issues

Encoding UTF-8

LazyData true

RoxygenNote 7.3.3

NeedsCompilation yes

Author Dolors Pelegri-Siso [aut, cre] (ORCID: <https://orcid.org/0000-0002-5993-3003>), Juan R. González [aut], Edsel A. Peña [aut], Robert L. Strawderman [aut]

Maintainer Dolors Pelegri-Siso <dolors.pelegri@isglobal.org>

Repository https://cran.r-universe.dev

Date/Publication 2026-09-22 07:00:02 UTC

RemoteUrl https://github.com/cran/survrec

RemoteRef HEAD

RemoteSha 46d5aef2fb994af869bd9a0cbab26bda22804acf

Contents

survrec-package . . . 2
as.data.frame.survfitr . . . 3
autoplot.mcf . . . 4
autoplot.survdiffr . . . 5
autoplot.survfitr . . . 5
colon . . . 7
mcf . . . 7
mlefrailty_fit . . . 8
MMC . . . 11
plot.survfitr . . . 11
plotEstimators . . . 12
print.survfitr . . . 13
psh_fit . . . 14
q_search . . . 15
quantile.survfitr . . . 16
summary.survdiffr . . . 17
summary.survfitr . . . 18
surv_search . . . 19
survdiffr . . . 19
survfitr . . . 21
Survr . . . 23
survrecThreads . . . 24
theme_survrec . . . 25
wc_fit . . . 25

Index **28**

`survrec-package` *survrec: Survival Analysis for Recurrent Event Data*

Description

Estimation of the survival function of inter-occurrence times for recurrent event data. Three estimators are available: the generalized product-limit estimator of Peña, Strawderman and Hollander (2001), the estimator of Wang and Chang (1999) for correlated inter-occurrence times, and maximum likelihood estimation under a gamma frailty model. Survival quantiles can be compared between groups through several bootstrap schemes.

Details

The main entry points are `Survr()` to build the response object, `survfitr()` to estimate survival curves (optionally by group) and `survdiffr()` to bootstrap survival quantiles.

Author(s)

Maintainer: Dolors Pelegri-Siso <dolors.pelegri@isglobal.org> (ORCID)

Authors:

- Juan R. González <juanr.gonzalez@isglobal.org>
- Edsel A. Peña
- Robert L. Strawderman

References

Peña, E.A., Strawderman, R. and Hollander, M. (2001). Nonparametric Estimation with Recurrent Event Data. *Journal of the American Statistical Association* **96**, 1299–1315.

Wang, M.-C. and Chang, S.-H. (1999). Nonparametric Estimation of a Recurrent Survival Function. *Journal of the American Statistical Association* **94**, 146–153.

See Also

Useful links:

- https://github.com/isglobal-brge/survrec
- Report bugs at https://github.com/isglobal-brge/survrec/issues

---

`as.data.frame.survfitr`

Coerce a survfitr object to a data frame

Description

Returns the estimated curves in long ("tidy") format, one row per distinct event time (and group), ready for further processing or plotting.

Usage

```
## S3 method for class 'survfitr'
as.data.frame(x, row.names = NULL, optional = FALSE, ...)
```

Arguments

`x`	a `survfitr` object.
`row.names, optional`	see `as.data.frame()`; ignored.
`...`	ignored.

Value

A data frame with columns `group` (only when the fit has strata), `time`, `n.event`, `n.risk`, `surv` and `std.error` (`NA` for estimators without standard errors).

Examples

```
data(MMC)
fit <- survfitr(Survr(id, time, event) ~ group,
  data = MMC,
  type = "wang-chang"
)
head(as.data.frame(fit))
```

---

autoplot.mcf *Plot a mean cumulative function*

Description

Draws the mean cumulative function estimated by `mcf()` as a step function, one colour per group.

Usage

```
## S3 method for class 'mcf'
autoplot(object, ...)
```

Arguments

`object`	an object of class `mcf`.
`...`	ignored; kept for compatibility with the generic.

Value

A `ggplot` object.

See Also

`mcf()`, `theme_survrec()`

Examples

```
data(colon)
autoplot(mcf(Survr(hc, time, event) ~ as.factor(dukes), data = colon))
```

autoplot.survdiffr *Plot the bootstrap distributions of survival quantiles*

Description

Draws the density of the bootstrap replicates of each group, with a vertical line at the observed quantile. Infinite replicates (survival estimate never below the requested level) are excluded from the densities and reported in the caption.

Usage

```
## S3 method for class 'survdiffr'
autoplot(object, ...)
```

Arguments

`object`	a grouped result of `survdiffr()`.
`...`	ignored; kept for compatibility with the generic.

Value

A `ggplot` object.

See Also

`survdiffr()`, `summary.survdiffr()`, `theme_survrec()`

Examples

```
data(colon)
fit <- survdiffr(Survr(hc, time, event) ~ as.factor(dukes),
  data = colon, q = 0.5, seed = 1
)
autoplot(fit)
```

autoplot.survfitr *Plot survival curves with ggplot2*

Description

Draws the estimated survival function of a `survfitr` object as a step function, one colour per group, with an optional pointwise confidence band. The cumulative distribution (`fun = "event"`) and the cumulative hazard (`fun = "cumhaz"`) transformations are also available.

Usage

```
## S3 method for class 'survfitr'
autoplot(
  object,
  fun = c("surv", "event", "cumhaz"),
  conf.int = TRUE,
  level = 0.95,
  conf.type = c("log-log", "plain"),
  ...
)
```

Arguments

`object`	an object of class `survfitr` (output of `survfitr()`, `psh_fit()`, `wc_fit()` or `mlefrailty_fit()`).
`fun`	transformation of the survival curve: `"surv"` (default, the survival function), `"event"` (the cumulative probability of an event, 1 - S) or `"cumhaz"` (the cumulative hazard, -log S).
`conf.int`	draw the pointwise confidence band? Default `TRUE` (ignored for estimators without standard errors).
`level`	confidence level of the band.
`conf.type`	transformation used for the band: `"log-log"` (default) or `"plain"`.
`...`	ignored; kept for compatibility with the generic.

Details

The confidence band is computed from the standard errors of the fit when the estimator provides them (Peña-Strawderman-Hollander and Wang-Chang). With `conf.type = "log-log"` (default) the limits are obtained on the $\log(-\log S)$ scale, so they always stay inside $[0, 1]$; `"plain"` gives the symmetric limits of the base `plot.survfitr()` method.

Value

A `ggplot` object, which can be further styled with the usual `ggplot2` syntax.

See Also

`plot.survfitr()` for the base-graphics version, `plotEstimators()` to compare the three estimators, `theme_survrec()`

Examples

```
data(MMC)
fit <- survfitr(Survr(id, time, event) ~ group,
  data = MMC,
  type = "wang-chang"
)
autoplot(fit)
```

```
autoplot(fit, fun = "cumhaz", conf.int = FALSE)

# it is a ggplot, so it can be restyled
autoplot(fit) + ggplot2::labs(title = "Migratory Motor Complex")
```

`colon` *Rehospitalization in colorectal cancer*

Description

Rehospitalization times after surgery in patients diagnosed with colorectal cancer.

Usage

```
colon
```

Format

A data frame with 861 rows and 6 columns:

hc subject identifier, repeated for each recurrence

time rehospitalization or censoring gap time

event censoring status: 1 for every rehospitalization, 0 for the last (censored) time of each subject

chemoter did the patient receive chemotherapy? 1: no, 2: yes

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

distance distance from residence to hospital: 1: <= 30 km, 2: > 30 km

Source

González, J.R., Fernandez, E., Moreno, V. et al. (2005). Sex differences in hospital readmission among colorectal cancer patients. *Journal of Epidemiology and Community Health* **59**, 506–511.

`mcf` *Mean cumulative function of recurrent event data*

Description

Nonparametric (Nelson-Aalen type) estimate of the mean cumulative function, i.e. the expected number of events experienced by a subject up to each calendar time, optionally by group.

Usage

```
mcf(formula, data)

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

Arguments

formula	a formula object with a Survr() object as the response on the left of the ~ operator and a grouping term (or 1) on the right.
data	a data frame in which to interpret the variables named in the formula.
x	an mcf object.
...	further arguments passed to print.data.frame().

Details

Calendar times are reconstructed as the within-subject cumulative sums of the gap times of the Survr() response. At each distinct event calendar time t the estimate increases by $dN(t)/Y(t)$, where $dN(t)$ is the number of events observed at t and $Y(t)$ the number of subjects still under observation (total observation time $\geq t$).

Value

An object of class mcf: a data frame with columns group, time, n.event, n.risk and mcf, with print and autoplot methods.

References

Lawless, J.F. and Nadeau, C. (1995). Some Simple Robust Methods for the Analysis of Recurrent Events. *Technometrics* **37**, 158–168.

See Also

autoplot.mcf(), survfitr()

Examples

```
data(MMC)
m <- mcf(Survr(id, time, event) ~ group, data = MMC)
m
autoplot(m)
```

---

mlefrailty_fit *Survival estimator under a gamma frailty model*

Description

Maximum likelihood estimation of the survival function of correlated inter-occurrence times of recurrent event data under a gamma frailty model.

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, see Survr().
tvals	optional vector of times at which the survival function is also evaluated.
lambda	optional vector of baseline hazard probabilities at the distinct event times. Defaults to the occurrence/exposure rates n.event/colSums(AtRisk).
alpha	optional shape and scale parameter of the frailty distribution. If unknown it is estimated via the EM algorithm, starting from a seed obtained by maximising the profile likelihood (see details).
alpha.min	optional left bound of the interval used to find the seed of alpha. Default is 0.5.
alpha.max	optional right bound of the interval used to find the seed of alpha. Default is the maximum distinct event time.
tol	convergence tolerance of the EM algorithm. Default is 1e-7.
maxiter	maximum number of EM iterations. Default is 500.
alpha.console	if TRUE, prints the seed value and the final estimate of alpha.

Details

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

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

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

The parameter α controls the degree of association between inter-occurrence times within a unit. Peña, Strawderman and Hollander (2001) showed that α and Λ_0 can be estimated by maximising the marginal likelihood function with an expectation-maximisation (EM) algorithm.

To achieve good convergence, a seed value for α is estimated first by maximising the profile likelihood for α over the interval (alpha.min, alpha.max) with Brent's method. If the EM algorithm

does not converge from that seed, it is restarted from up to six additional seeds around it. If convergence still fails, the initial value of alpha can be used as the `alpha.min` argument and the fit recomputed.

Value

A list with class `survfitr` containing:

n number of units or subjects observed.

m number of events of each subject.

failed uncensored gap times, ordered by subject.

censored censored gap time of each subject.

time ordered distinct event times.

n.event number of events at each distinct time.

AtRisk matrix of units at risk at each distinct time (rows are subjects).

status 1 if the EM algorithm converged, -1 if it reached `maxiter` without converging (in that case no estimates are provided).

alpha estimate of the gamma frailty parameter.

lambda estimated hazard probabilities at the distinct event times.

frailties posterior frailty estimate of each subject; values spread away from 1 indicate heterogeneity between subjects.

survfunc survival estimate at each distinct time.

tvals copy of the `tvals` argument.

MLEAttvals survival estimate at the `tvals` times.

References

Peña, E.A., Strawderman, R. and Hollander, M. (2001). Nonparametric Estimation with Recurrent Event Data. *Journal of the American Statistical Association* **96**, 1299–1315.

See Also

`survfitr()`, `Survr()`

Examples

```
data(MMC)
fit <- mlefrailty_fit(Survr(MMC$id, MMC$time, MMC$event))
fit
plot(fit)

# compare with Peña-Strawderman-Hollander
fit <- psh_fit(Survr(MMC$id, MMC$time, MMC$event))
lines(fit, lty = 2)

# and with Wang-Chang
fit <- wc_fit(Survr(MMC$id, MMC$time, MMC$event))
lines(fit, lty = 3)
```

MMC *Migratory Motor Complex*

Description

Times of the Migratory Motor Complex (MMC) of 19 healthy individuals, measured through small bowel manometry.

Usage

```
MMC
```

Format

A data frame with 99 rows and 4 columns:

id subject identifier, repeated for each recurrence

time recurrence or censoring gap time

event censoring status: 1 for every recurrence, 0 for the last (censored) time of each subject

group a factor with levels `Males` and `Females`. Note: the groups were created at random to illustrate a group comparison

Source

Husebye, E., Skar, V., Aalen, O.O. and Osnes, M. (1990). Digital ambulatory manometry of the small intestine in healthy adults. *Digestive Diseases and Sciences* **35**, 1057–1065.

plot.survfitr *Plot survival curves of recurrent event data*

Description

Plots the estimated survival (or probability) function from a `survfitr` object, one curve per group when the fit has strata. Additional fits can be added to the same axes with the `lines` method.

Usage

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

## S3 method for class 'survfitr'
lines(x, prob = FALSE, ...)
```

Arguments

x	an object of class survfitr (output of survfitr(), psh_fit(), wc_fit() or mlefrailty_fit()).
conf.int	if TRUE (default), pointwise 95\ are drawn for the estimators that provide standard errors.
prob	if TRUE the probability function (1 - survival) is drawn instead of the survival function.
...	additional arguments passed to plot().

Value

No return value; called for its side effect.

See Also

psh_fit(), wc_fit(), mlefrailty_fit()

Examples

```
data(MMC)
fit <- survfitr(Survr(id, time, event) ~ group,
  data = MMC,
  type = "wang-chang"
)
plot(fit)
```

---

plotEstimators *Compare the three survival estimators on the same data*

Description

Fits the Peña-Strawderman-Hollander, the Wang-Chang and the gamma frailty MLE estimators to the same sample and draws the three curves on the same axes. Agreement between the PSH and the frailty estimates suggests independent inter-occurrence times, while a Wang-Chang curve separated from the PSH one points to within-subject correlation.

Usage

```
plotEstimators(x, conf.int = FALSE, level = 0.95, alpha.console = FALSE, ...)
```

Arguments

x	a survival recurrent event object, see Survr().
conf.int	draw the pointwise confidence bands of the estimators that provide them? Default FALSE.
level	confidence level of the bands.
alpha.console	passed to mlefrailty_fit(); default FALSE.
...	additional arguments passed to the three fitting functions.

Value

A ggplot object.

See Also

autoplot.survfitr(), psh_fit(), wc_fit(), mlefrailty_fit()

Examples

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

---

print.survfitr *Print a short summary of survival curves for recurrent event data*

Description

Prints, for each curve, the number of subjects, the number of events, the restricted mean survival and its standard error, the median survival and the minimum, maximum and median number of recurrences per subject.

Usage

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

Arguments

x	the result of a call to survfitr(), psh_fit(), wc_fit() or mlefrailty_fit().
scale	a numeric value to rescale the survival time, e.g. if the input data are in days, scale = 365 scales the printout to years.
digits	number of digits to print.
...	other unused arguments.

Details

The restricted mean and its standard error are based on a truncated estimator: if the survival curve does not reach zero, the reported quantity is the mean survival restricted to the time before the last censoring. The median is the time at which the survival curve crosses 0.5.

Value

x, invisibly.

See Also

summary.survfitr(), survfitr()

Examples

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

---

psh_fit *Peña-Strawderman-Hollander survival estimator*

Description

Estimates the survival function of the inter-occurrence times of recurrent event data with the generalized product-limit estimator (PLE) of Peña, Strawderman and Hollander (2001).

Usage

```
psh_fit(x, tvals)
```

Arguments

x	a survival recurrent event object, see Survr().
tvals	optional vector of times at which the survival function is also evaluated.

Details

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

Value

A list with class survfitr containing:

n number of units or subjects observed.

m number of events of each subject.

failed uncensored gap times, ordered by subject.

censored censored gap time of each subject.

time ordered distinct event times.

n.event number of events at each distinct time.

AtRisk matrix of units at risk at each distinct time (rows are subjects).

survfunc survival estimate at each distinct time.

std.error standard error of the survival estimate.

tvals copy of the tvals argument.

PSHpleAttvals survival estimate at the tvals times.

References

Peña, E.A., Strawderman, R. and Hollander, M. (2001). Nonparametric Estimation with Recurrent Event Data. *Journal of the American Statistical Association* **96**, 1299–1315.

See Also

`survfitr()`, `Survr()`

Examples

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

# compare with the gamma frailty MLE
fit <- mlefrailty_fit(Survr(MMC$id, MMC$time, MMC$event))
lines(fit, lty = 2)

# and with Wang-Chang
fit <- wc_fit(Survr(MMC$id, MMC$time, MMC$event))
lines(fit, lty = 3)
```

---

q_search *Survival time of a selected quantile*

Description

Given a `survfitr` object, returns the first time at which the estimated survival function falls to the quantile `q` or below.

Usage

```
q_search(f, q = 0.5)
```

Arguments

`f`	a `survfitr` object.
`q`	quantile. Default is 0.5 (the median survival time).

Value

The survival time of the selected quantile, or `NA` (with a warning) when the survival estimates are not available.

Examples

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

# 75th percentile of the survival function
q_search(fit, q = 0.75)
```

`quantile.survfitr` *Quantiles of the survival time*

Description

Returns, for each requested probability `p`, the first time at which the estimated survival function falls to `1 - p` or below (so `probs = 0.5` is the median survival time). `NA` is returned when the curve does not reach the requested level.

Usage

```
## S3 method for class 'survfitr'
quantile(x, probs = c(0.25, 0.5, 0.75), ...)
```

Arguments

`x`	a `survfitr` object.
`probs`	vector of probabilities.
`...`	ignored; kept for compatibility with the generic.

Value

A named vector of survival times, or a matrix with one row per group when the fit has strata.

See Also

`q_search()`, `survfitr()`

Examples

```
data(MMC)
fit <- survfitr(Survr(id, time, event) ~ group,
  data = MMC,
  type = "wang-chang"
)
quantile(fit, probs = c(0.25, 0.5, 0.75))
```

`summary.survdiffr` *Summarize a grouped bootstrap of survival quantiles*

Description

Computes, for each group, the observed quantile and a percentile bootstrap confidence interval, and for each pair of groups the difference of quantiles with its percentile interval and a two-sided bootstrap p-value for the null hypothesis of no difference.

Usage

```
## S3 method for class 'survdiffr'
summary(object, level = 0.95, ...)

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

Arguments

`object`	a grouped result of `survdiffr()`.
`level`	confidence level of the percentile intervals.
`...`	ignored; kept for compatibility with the generic.
`x`	a `summary.survdiffr` object.
`digits`	number of digits to print.

Details

Replicates in which the survival estimate never fell below the requested level are treated as `+Inf` (the quantile exceeds the observed follow-up); pairs where both replicates are infinite are dropped from the difference. The p-value is $2\min(P(D \le 0), P(D \ge 0))$ over the replicated differences D.

Value

A list of class `summary.survdiffr` with components `groups` (data frame with the observed quantile and its interval per group) and `contrasts` (data frame with the pairwise differences, intervals and p-values).

See Also

`survdiffr()`, `boot::boot.ci()`

Examples

```
data(colon)
fit <- survdiffr(Survr(hc, time, event) ~ as.factor(dukes),
  data = colon, q = 0.5, seed = 1
)
summary(fit)
```

`summary.survfitr` *Summary of survival curves for recurrent event data*

Description

Returns a matrix with the distinct event times and, at each of them, the number of events, the number of subjects at risk, the survival estimate and (when available) its standard error. If the fit has multiple curves, a list with one matrix per curve is returned.

Usage

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

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

Arguments

`object`	output of a call to `survfitr()`, `psh_fit()`, `wc_fit()` or `mlefrailty_fit()`.
`...`	other unused arguments.
`x`	a `summary.survfitr` object.
`scale`	a numeric value to rescale the survival time.
`digits`	number of digits to print.

Value

For a single curve, a matrix; for multiple curves, a list of class `summary.survfitr` with one matrix per curve.

See Also

`survfitr()`

Examples

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

surv_search *Survival estimate at selected times*

Description

Evaluates a step survival function at arbitrary times. The estimate is a decreasing piecewise-constant function with jumps at the event times, so the value at any time is the one of the previous event time.

Usage

```
surv_search(tvals, time, surv)
```

Arguments

tvals	vector of times at which the survival function is evaluated.
time	vector of ordered distinct event times.
surv	vector of survival estimates at each event time.

Value

The survival estimate at each of the tvals times.

Examples

```
time <- c(4, 7, 9, 15, 21, 67)
surv <- c(0.8, 0.7, 0.65, 0.55, 0.43, 0.22)

# survival at times 1, 10, 32 and 74
surv_search(c(1, 10, 32, 74), time, surv)
```

survdiffr *Bootstrap survival quantiles for recurrent event data*

Description

Obtains bootstrap replicates of a quantile (e.g. the median) of the survival time for each group of subjects. Confidence intervals of the replicates can be computed with boot::boot.ci().

Usage

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

Arguments

formula	a formula object with a Survr() object as the response on the left of the ~ operator and a grouping term on the right.
data	a data frame in which to interpret the variables named in the formula.
q	quantile of the survival function whose time is bootstrapped.
B	number of bootstrap samples.
boot.F	a character string specifying the bootstrap procedure: "PSH" or "WC" (nonparametric) or "semiparametric". The default is "WC". Only the first letters are required, e.g. "P", "W", "se".
boot.G	a character string specifying whether the censoring times are also resampled from their empirical distribution: "none" (default) or "empirical". Only the first letters are required.
seed	optional random seed for the bootstrap resampling (passed to set.seed()). If NULL (default) the current state of the random number generator is used.
...	additional arguments passed to the estimator used for the observed (non-bootstrap) quantile.

Details

Three resampling schemes are available through boot.F: nonparametric resampling of gap times from the Peña-Strawderman-Hollander ("PSH") or the Wang-Chang ("WC") estimates of the inter-occurrence time distribution, and semiparametric resampling under the fitted gamma frailty model ("semiparametric"). With boot.G = "empirical" the per-subject censoring times are additionally resampled from their empirical distribution.

All resampling uses R's random number generator, so results are reproducible with set.seed() or the seed argument. The re-estimation of the survival curve on each replicate can run in parallel, see survrecThreads(). Some procedures (notably the semiparametric one, which refits the frailty model on every replicate) can be slow.

Value

An object of class boot (or a named list with one per group) whose t0 is the observed quantile and t the bootstrap replicates. A replicate is -1 when the survival estimate does not fall below q, and NA

when the resampled data had no events. For the semiparametric procedure the component `alpha` contains the frailty parameter estimated on each replicate.

References

González, J.R. and Peña, E.A. (2003). Bootstrapping median survival with recurrent event data. *IX Conferencia Española de Biometría*.

See Also

`survfitr()`, `boot::boot.ci()`, `survrecThreads()`

Examples

```
data(colon)

# compare the median survival time between the three Dukes stages
fit <- survdiffr(Survr(hc, time, event) ~ as.factor(dukes),
  data = colon, q = 0.5, seed = 1
)
boot::boot.ci(fit$"1", type = c("norm", "basic", "perc"))
boot::boot.ci(fit$"2", type = c("norm", "basic", "perc"))
boot::boot.ci(fit$"3", type = c("norm", "basic", "perc"))

# 75th quantile with percentile confidence intervals
fit <- survdiffr(Survr(hc, time, event) ~ as.factor(dukes),
  data = colon, q = 0.75, seed = 1
)
quantile(fit$"1"$t, c(0.025, 0.975))

# resampling from the PSH estimate instead
fit <- survdiffr(Survr(hc, time, event) ~ as.factor(dukes),
  data = colon, q = 0.5, boot.F = "PSH", seed = 1
)
```

---

survfitr *Survival curves for recurrent event data*

Description

Computes an estimate of the survival curve of inter-occurrence times for recurrent event data, optionally by group, using the Peña-Strawderman-Hollander, the Wang-Chang or the gamma frailty MLE estimators, together with their standard errors.

Usage

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

Arguments

formula	a formula object with a Survr() object as the response on the left of the ~ operator and a grouping term on the right. For a single survival curve use ~ 1.
data	a data frame in which to interpret the variables named in the formula.
type	a character string specifying the estimator: "pena-strawderman-hollander", "wang-chang" or "MLEfrailty" (default). Only the first letters are required, e.g. "pe", "wa", "ML".
...	additional arguments passed to the fitting function of the chosen estimator.

Details

See psh_fit(), wc_fit() and mlefrailty_fit() for details on each estimator; additional arguments in ... are passed to the chosen fitting function.

Value

A survfitr object: the fit of the chosen estimator, or a named list with one fit per group when the formula defines groups. Methods are provided for print, plot, lines and summary.

References

Peña, E.A., Strawderman, R. and Hollander, M. (2001). Nonparametric Estimation with Recurrent Event Data. *Journal of the American Statistical Association* **96**, 1299–1315.

Wang, M.-C. and Chang, S.-H. (1999). Nonparametric Estimation of a Recurrent Survival Function. *Journal of the American Statistical Association* **94**, 146–153.

See Also

print.survfitr(), plot.survfitr(), summary.survfitr(), Survr(), psh_fit(), wc_fit(), mlefrailty_fit()

Examples

```
data(colon)
# fit a Peña-Strawderman-Hollander estimator by Dukes stage
fit <- survfitr(Survr(hc, time, event) ~ as.factor(dukes),
  data = colon, type = "pena"
)
plot(fit, ylim = c(0, 1), xlim = c(0, 2000))
fit
summary(fit)
```

`Survr` *Create a survival recurrent object*

Description

Creates a survival recurrent object, usually used as the response variable in a model formula.

Usage

```
Survr(id, time, event)

is.Survr(x)
```

Arguments

`id`	identifier of each subject; the same value for all the recurrent times of a subject.
`time`	gap times of recurrence. The last time of each subject is censored.
`event`	status indicator: 1 = recurrence, 0 = censored. Only these values are accepted.
`x`	any R object.

Details

Each subject must contribute one censored time (its last, incomplete gap time), so `id` must have exactly as many distinct values as there are zeros in `event`. Rows are assumed to be grouped by subject and in chronological order within subject.

Value

An object of class `Survr`, implemented as a matrix with columns `id`, `time` and `event`.

`is.Survr()` returns `TRUE` if `x` inherits from class `Survr` and `FALSE` otherwise.

See Also

`survfitr()`, `psh_fit()`, `wc_fit()`, `mlefrailty_fit()`

Examples

```
data(MMC)
x <- Survr(MMC$id, MMC$time, MMC$event)
is.Survr(x)
```

`survrecThreads` *Number of threads used by survrec*

Description

Gets or sets the maximum number of threads used by the parallel parts of the package, currently the re-estimation phase of the bootstrap in `survdiffr()`, which refits the survival curve once per bootstrap replicate.

Usage

```
survrecThreads(n)
```

Arguments

`n`	Maximum number of threads, or `NULL` to restore the default (let OpenMP decide). If missing, the current setting is returned without changing it.

Details

By default the package lets OpenMP decide, which means it honours the `OMP_NUM_THREADS` environment variable. Set this option to cap the number of threads regardless of that variable, for instance when running inside a shared machine or a job scheduler.

The bootstrap resampling itself always runs serially because it draws from R's random number generator (which is not thread-safe); results are therefore reproducible with `set.seed()` independently of the number of threads.

If the package was built without OpenMP support (which is the case with the default compiler on macOS), the computation runs sequentially and this setting has no effect.

Value

The current setting, invisibly when it is being changed. `NULL` means "let OpenMP decide".

Examples

```
survrecThreads() # current setting

old <- survrecThreads(2)
survrecThreads(old) # restore
```

theme_survrec *Plot theme used by the survrec graphics*

Description

A clean ggplot2 theme shared by all the plotting functions of the package: light horizontal grid, no panel border, muted axis text and room for the legend at the bottom. Exported so that the figures of a report can be restyled consistently. It matches the theme of the companion package gcmrec.

Usage

```
theme_survrec(base_size = 12)
```

Arguments

base_size	Base font size in points.

Value

A ggplot2 theme object, to be added to any plot.

Examples

```
data(MMC)
fit <- survfitr(Survr(id, time, event) ~ group,
  data = MMC,
  type = "wang-chang"
)
autoplot(fit) + theme_survrec(base_size = 14)
```

wc_fit *Wang-Chang survival estimator*

Description

Estimates the survival function of correlated or i.i.d. inter-occurrence times of recurrent event data with the product-limit estimator of Wang and Chang (1999).

Usage

```
wc_fit(x, tvals)
```

Arguments

x	a survival recurrent event object, see Survr().
tvals	optional vector of times at which the survival function is also evaluated.

Details

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

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

Value

A list with class `survfitr` containing:

n number of units or subjects observed.

m number of events of each subject.

failed uncensored gap times, ordered by subject.

censored censored gap time of each subject.

time ordered distinct event times.

n.event weighted number of events at each distinct time.

AtRisk weighted number of units at risk at each distinct time.

survfunc survival estimate at each distinct time.

std.error standard error of the survival estimate.

tvals copy of the `tvals` argument.

WCpleAttvals survival estimate at the `tvals` times.

Note

The maintainers wish to thank Professors Chiung-Yu Huang and Shu-Hui Chang for providing the original Fortran code computing the standard errors of Wang and Chang's estimator.

References

Wang, M.-C. and Chang, S.-H. (1999). Nonparametric Estimation of a Recurrent Survival Function. *Journal of the American Statistical Association* **94**, 146–153.

See Also

`survfitr()`, `Survr()`

Examples

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

# compare with Peña-Strawderman-Hollander
fit <- psh_fit(Survr(MMC$id, MMC$time, MMC$event))
lines(fit, lty = 2)

# and with the gamma frailty MLE
fit <- mlefrailty_fit(Survr(MMC$id, MMC$time, MMC$event))
lines(fit, lty = 3)
```

Index

∗ **datasets**
colon, 7
MMC, 11
∗ **survival**
mcf, 7
mlefrailty_fit, 8
plot.survfitr, 11
print.survfitr, 13
psh_fit, 14
q_search, 15
summary.survfitr, 18
surv_search, 19
survdiffr, 19
survfitr, 21
Survr, 23
wc_fit, 25

as.data.frame(), *3*
as.data.frame.survfitr, 3
autoplot.mcf, 4
autoplot.mcf(), *8*
autoplot.survdiffr, 5
autoplot.survfitr, 5
autoplot.survfitr(), *13*

boot::boot.ci(), *17*, *19*, *21*

colon, 7

is.Survr *(*Survr*)*, 23

lines.survfitr *(*plot.survfitr*)*, 11

mcf, 7
mcf(), *4*
mlefrailty_fit, 8
mlefrailty_fit(), *6*, *12, 13*, *18*, *22, 23*
MMC, 11

plot(), *12*
plot.survfitr, 11
plot.survfitr(), *6*, *22*
plotEstimators, 12
plotEstimators(), *6*
print.data.frame(), *8*
print.mcf *(*mcf*)*, 7
print.summary.survdiffr
*(*summary.survdiffr*)*, 17
print.summary.survfitr
*(*summary.survfitr*)*, 18
print.survfitr, 13
print.survfitr(), *22*
psh_fit, 14
psh_fit(), *6*, *12, 13*, *18*, *22, 23*

q_search, 15
q_search(), *16*
quantile.survfitr, 16

set.seed(), *20*, *24*
summary.survdiffr, 17
summary.survdiffr(), *5*
summary.survfitr, 18
summary.survfitr(), *13*, *22*
surv_search, 19
survdiffr, 19
survdiffr(), *3*, *5*, *17*, *24*
survfitr, 21
survfitr(), *3*, *6*, *8*, *10*, *12, 13*, *15, 16*, *18*, *21*, *23*, *26*
Survr, 23
Survr(), *3*, *8–10*, *12*, *14, 15*, *20*, *22*, *25, 26*
survrec *(*survrec-package*)*, 2
survrec-package, 2
survrecThreads, 24
survrecThreads(), *20, 21*

theme_survrec, 25
theme_survrec(), *4–6*

wc_fit, 25
wc_fit(), *6*, *12, 13*, *18*, *22, 23*