Package ‘longevity’

February 3, 2026

Type Package

Title Statistical Methods for the Analysis of Excess Lifetimes

Version 1.3

BugReports `https://github.com/lbelzile/longevity/issues`

URL `https://lbelzile.github.io/longevity/`

Depends R (>= 4.0.0)

Imports numDeriv, Rcpp (>= 1.0.6), rlang, Rsolnp

Suggests knitr, ggplot2 (>= 3.0.0), tinytest, rmarkdown

LinkingTo Rcpp, RcppArmadillo

Description A collection of parametric and nonparametric methods for the analysis of survival data. Parametric families implemented include Gompertz-Makeham, exponential and generalized Pareto models and extended models. The package includes an implementation of the nonparametric maximum likelihood estimator for arbitrary truncation and censoring pattern based on Turnbull (1976) <doi:10.1111/j.2517-6161.1976.tb01597.x>, along with graphical goodness-of-fit diagnostics. Parametric models for positive random variables and peaks over threshold models based on extreme value theory are described in Rootzén and Zholud (2017) <doi:10.1007/s10687-017-0305-5>; Belzile et al. (2021) <doi:10.1098/rsos.202097> and Belzile et al. (2022) <doi:10.1146/annurev-statistics-040120-025426>.

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 7.3.3

VignetteBuilder knitr

NeedsCompilation yes

Author Leo Belzile [aut, cre] (ORCID: <`https://orcid.org/0000-0002-9135-014X`>), Mitchell O'Hara-Wild [ctb] (ORCID: <`https://orcid.org/0000-0001-6729-7695`>)

Maintainer Leo Belzile <`belzilel@gmail.com`>

Repository CRAN

Date/Publication 2026-02-03 12:00:09 UTC

Contents

autoplot.elife_northropcoleman 2
autoplot.elife_par 3
autoplot.elife_tstab 5
dutch 6
endpoint.profile 7
ewsim 8
fit_elife 9
idlmetadata 11
japanese 11
japanese2 12
lpost_elife 13
nc_test 14
nll_elife 16
npsurv 17
np_elife 19
plot.elife_ecdf 21
plot.elife_profile 21
plot.elife_tstab_endpoint 22
samp_elife 23
test_elife 24
tstab 26

Index **28**

`autoplot.elife_northropcoleman` *P-value plot for Northrop and Coleman diagnostic*

Description

The Northrop-Coleman tests for penultimate models are comparing the piece-wise generalized Pareto distribution to the generalized Pareto above the lower threshold.

Usage

```
autoplot.elife_northropcoleman(object, ...)

## S3 method for class 'elife_northropcoleman'
plot(x, plot.type = c("base", "ggplot"), plot = TRUE, ...)
```

Arguments

`object`	object of class `elife_northropcoleman`, with the fitted piecewise-constant generalized Pareto model
`...`	additional arguments for base `plot`
`x`	an object of class `elife_northropcoleman`

`plot.type`	string indicating the type of plot
`plot`	logical; should the routine print the graph if `plot.type` equals `"ggplot"`? Default to `TRUE`.

Value

a base R or ggplot object with p-values for the Northrop-Coleman test against thresholds.

`autoplot.elife_par` *Goodness-of-fit plots for parametric models*

Description

Because of censoring and truncation, the plotting positions must be adjusted accordingly. For right-censored data, the methodology is described in Waller & Turnbull (1992). Only non-censored observations are displayed, which can create distortion.

Usage

```
autoplot.elife_par(object, ...)

## S3 method for class 'elife_par'
plot(
  x,
  plot.type = c("base", "ggplot"),
  which.plot = c("pp", "qq"),
  confint = c("none", "pointwise", "simultaneous"),
  plot = TRUE,
  ...
)
```

Arguments

`object`	an object of class `elife_par` containing the fitted parametric model
`...`	additional arguments, currently ignored by the function.
`x`	a parametric model of class `elife_par`
`plot.type`	string, one of `base` for base R or `ggplot`
`which.plot`	vector of string indicating the plots, among `pp` for probability-probability plot, `qq` for quantile-quantile plot, `erp` for empirically rescaled plot (only for censored data), `exp` for standard exponential quantile-quantile plot or `tmd` for Tukey's mean difference plot, which is a variant of the Q-Q plot in which we map the pair (x, y) is mapped to `((x+y)/2,y-x)` are detrended, `dens` and `cdf` return the empirical distribution function with the fitted parametric density or distribution function curve superimposed.
`confint`	logical; if `TRUE`, creates uncertainty diagnostic via a parametric bootstrap
`plot`	logical; if `TRUE`, creates a plot when `plot.type="ggplot"`. Useful for returning `ggplot` objects without printing the graphs

Details

For truncated data, we first estimate the distribution function nonparametrically, F_n. The uniform plotting positions of the data

$$v_i = [F_n(y_i) - F_n(a_i)]/[F_n(b_i) - F_n(a_i)].$$

For probability-probability plots, the empirical quantiles are transformed using the same transformation, with F_n replaced by the postulated or estimated distribution function F_0. For quantile-quantile plots, the plotting positions v_i are mapped back to the data scale viz.

$$F_0^{-1}\{F_0(a_i) + v_i[F_0(b_i) - F_0(a_i)]\}$$

When data are truncated and observations are mapped back to the untruncated scale (with, e.g., `exp`), the plotting positions need not be in the same order as the order statistics of the data.

Value

The function produces graphical goodness-of-fit plots using base R or ggplot objects (returned as an invisible list).

Examples

```
set.seed(1234)
samp <- samp_elife(
 n = 200,
 scale = 2,
 shape = 0.3,
 family = "gomp",
 lower = 0, upper = runif(200, 0, 10),
 type2 = "ltrc")
fitted <- fit_elife(
 time = samp$dat,
 thresh = 0,
 event = ifelse(samp$rcens, 0L, 1L),
 type = "right",
 family = "exp",
 export = TRUE)
plot(fitted, plot.type = "ggplot")
# Left- and right-truncated data
n <- 40L
samp <- samp_elife(
 n = n,
 scale = 2,
 shape = 0.3,
 family = "gp",
 lower = ltrunc <- runif(n),
 upper = rtrunc <- ltrunc + runif(n, 0, 15),
 type2 = "ltrt")
fitted <- fit_elife(
 time = samp,
 thresh = 0,
 ltrunc = ltrunc,
```

```
 rtrunc = rtrunc,
 family = "gp",
 export = TRUE)
plot(fitted,  which.plot = c("tmd", "dens"))
```

`autoplot.elife_tstab` *Threshold stability plots*

Description

Threshold stability plots

Usage

```
autoplot.elife_tstab(object, ...)

## S3 method for class 'elife_tstab'
plot(
  x,
  plot.type = c("base", "ggplot"),
  which.plot = c("scale", "shape"),
  plot = TRUE,
  ...
)
```

Arguments

`object`	object of class `elife_tstab`, representing parameter estimates to draw threshold stability plots
`...`	additional arguments, currently ignored by the function.
`x`	an object of class `elife_tstab` containing the fitted parameters as a function of threshold
`plot.type`	string, one of `base` for base R or `ggplot`
`which.plot`	vector of string indicating the plots, among `pp` for probability-probability plot, `qq` for quantile-quantile plot, `erp` for empirically rescaled plot (only for censored data), `exp` for standard exponential quantile-quantile plot or `tmd` for Tukey's mean difference plot, which is a variant of the Q-Q plot in which we map the pair (x, y) is mapped to `((x+y)/2,y-x)` are detrended, `dens` and `cdf` return the empirical distribution function with the fitted parametric density or distribution function curve superimposed.
`plot`	logical; if `TRUE`, creates a plot when `plot.type="ggplot"`. Useful for returning `ggplot` objects without printing the graphs

Value

This S3 method is used to create plots; it only returns a list of graphs if `plot.type="ggplot"`

`dutch` *Dutch survival data*

Description

This data frame contains information about all Dutch who died above age 92 years between 1986 and 2015. Observations are doubly truncated and such bounds are calculated based on the range of plausible values for these variables. There are 226 records that are interval-censored and interval-truncated for which `bdate`, `ddate` and `ndays` is missing (`NA`).

Usage

```
dutch
```

Format

A data frame with 305143 rows and 11 variables:

ndays survival time (in days)

bdate the smallest plausible birth date given information about month of birth and death and survival (`Date`)

bmonth month of birth

byear year of birth

ddate the largest plausible death date given information about month of birth and death and survival (`Date`)

dmonth month of death

dyear year of death

ltrunc minimum age (in days); the maximum of either 92 years or the number of days reached in 1986

rtrunc maximum age (in days) an individual could have reached by the end of 2015

gender factor indicating gender of individual, either `female` or `male`

valid quality flag; `A` for individuals born in the Netherlands, `B` for individuals born abroad who died in the Netherlands

Source

Statistics Netherlands (CBS). Accessed via the Supplemental material of Einmahl, Einmahl and de Haan (2019)

References

Einmahl, J.J., J.H.J. Einmahl and L. de Haan (2019). *Limits to Human Life Span Through Extreme Value Theory*, Journal of the American Statistical Association, **114**(527), 1075-1080. doi:10.1080/01621459.2018.1537912

`endpoint.profile` *Profile likelihood for the endpoint of the generalized Pareto distribution*

Description

This function returns the profile log likelihood over a grid of values of `psi`, the endpoints.

Usage

```
endpoint.profile(
  time,
  time2 = NULL,
  event = NULL,
  thresh = 0,
  type = c("right", "left", "interval", "interval2"),
  ltrunc = NULL,
  rtrunc = NULL,
  weights = rep(1, length(time)),
  psi = NULL,
  confint = FALSE,
  level = 0.95,
  arguments = NULL,
  ...
)
```

Arguments

`time`	excess time of the event of follow-up time, depending on the value of event
`time2`	ending excess time of the interval for interval censored data only.
`event`	status indicator, normally 0=alive, 1=dead. Other choices are `TRUE`/`FALSE` (`TRUE` for death). For interval censored data, the status indicator is 0=right censored, 1=event at time, 2=left censored, 3=interval censored. Although unusual, the event indicator can be omitted, in which case all subjects are assumed to have experienced an event.
`thresh`	vector of thresholds
`type`	character string specifying the type of censoring. Possible values are `"right"`, `"left"`, `"interval"`, `"interval2"`.
`ltrunc`	lower truncation limit, default to `NULL`
`rtrunc`	upper truncation limit, default to `NULL`
`weights`	weights for observations
`psi`	mandatory vector of endpoints at which to compute the profile
`confint`	logical; if `TRUE`, return a `level` confidence interval instead of a list with the profile log-likelihood components

`level`	numeric; the level for the confidence intervals
`arguments`	a named list specifying default arguments of the function that are common to all `elife` calls
`...`	additional parameters, currently ignored

Value

a list with the maximum likelihood estimate of the endpoint and the profile log-likelihood

Examples

```
set.seed(2023)
time <- relife(n = 100, scale = 3, shape = -0.3, family = "gp")
endpt <- endpoint.profile(
  time = time,
  psi = seq(max(time) + 1e-4, max(time) + 40, length.out = 51L))
print(endpt)
plot(endpt)
confint(endpt)
```

`ewsim` *England and Wales simulated supercentenarian data*

Description

This data frame contains information about 179 fake records mimicking Welsh and English who died age 110 and above

Usage

```
ewsim
```

Format

A data frame with 179 rows and 3 variables:

time survival time above 110 (in years)

ltrunc minimum age above 110 (in years), or zero;

rtrunc maximum age (in years) an individual could have reached by the end of the time frame

`fit_elife` *Fit excess lifetime models by maximum likelihood*

Description

This function is a wrapper around constrained optimization routines for different models with non-informative censoring and truncation patterns.

Usage

```
fit_elife(
  time,
  time2 = NULL,
  event = NULL,
  type = c("right", "left", "interval", "interval2"),
  ltrunc = NULL,
  rtrunc = NULL,
  thresh = 0,
  status = NULL,
  family = c("exp", "gp", "weibull", "gomp", "gompmake", "extgp", "gppiece",
    "extweibull", "perks", "perksmake", "beard", "beardmake"),
  weights = NULL,
  export = FALSE,
  start = NULL,
  restart = FALSE,
  arguments = NULL,
  check = FALSE,
  ...
)
```

Arguments

`time`	excess time of the event of follow-up time, depending on the value of event
`time2`	ending excess time of the interval for interval censored data only.
`event`	status indicator, normally 0=alive, 1=dead. Other choices are `TRUE`/`FALSE` (`TRUE` for death). For interval censored data, the status indicator is 0=right censored, 1=event at time, 2=left censored, 3=interval censored. Although unusual, the event indicator can be omitted, in which case all subjects are assumed to have experienced an event.
`type`	character string specifying the type of censoring. Possible values are "`right`", "`left`", "`interval`", "`interval2`".
`ltrunc`	lower truncation limit, default to `NULL`
`rtrunc`	upper truncation limit, default to `NULL`
`thresh`	vector of thresholds

`status`	integer vector giving status of an observation. If `NULL` (default), this argument is computed internally based on `type`.
`family`	string; choice of parametric family
`weights`	weights for observations
`export`	logical; should data be included in the returned object to produce diagnostic plots? Default to `FALSE`.
`start`	vector of starting values for the optimization routine. If `NULL`, the algorithm attempts to find default values and returns a warning with false convergence diagnostic if it cannot.
`restart`	logical; should multiple starting values be attempted? Default to `FALSE`.
`arguments`	a named list specifying default arguments of the function that are common to all `elife` calls
`check`	logical; if `TRUE`, fit all submodels to ensure that simpler models fit worst or as well
`...`	additional parameters, currently ignored

Value

an object of class `elife_par`

Note

The extended generalized Pareto model is constrained to avoid regions where the likelihood is flat so $\xi \in [-1, 10]$ in the optimization algorithm.

The standard errors are obtained via the observed information matrix, calculated using the hessian. In many instances, such as when the shape parameter is zero or negative, the hessian is singular and no estimates are returned.

Examples

```
data(ewsim, package = "longevity")
fit1 <- fit_elife(
   arguments = ewsim,
   export = TRUE,
   family = "exp")
fit2 <- fit_elife(
   arguments = ewsim,
   export = TRUE,
   family = "gp")
plot(fit1)
summary(fit1)
anova(fit2, fit1)
```

`idlmetadata` *IDL metadata*

Description

This data frame contains country codes and the associated data collection period corresponding to the range for age at death.

Usage

```
idlmetadata
```

Format

A data frame with 21 rows and 4 variables:

country factor, one of `AUT` (Austria), `BEL` (Belgium), `CAN` (Quebec), `DEU` (Germany), `DNK` (Denmark), `ESP` (Spain), `FIN` (Finland), `FRA` (France), `JPN` (Japan), `NOR` (Norway), `SWE` (Sweden), `EAW` (England and Wales) and `USA` (United States of America)

group factor, either `105-109` for semi-supercentenarians or `110+` for supercentenarians"

ldate Date, smallest death date

rdate Date, latest death date

Details

Due to confidentiality restrictions, some data that were available in previous versions of the IDL for Switzerland, Italy and some entries for Japan and Belgium have been removed. As the IDL metadata are updated somewhat regularly and former versions of the database are not preserved, results from published analyses are replicable but not reproducible.

References

International Database on Longevity, extracted on February 13th, 2023

`japanese` *Japanese survival data*

Description

This data frame contains information about the counts of dead Japanese by gender and year of birth (cohort), categorized by the whole part of age attained at death.

Usage

```
japanese
```

Format

A data frame with 1038 rows and 4 variables:

age integer, age (to the smallest year) at death (in years)

byear integer, birth year

count integer, number of death for cohort at given age

gender factor, the gender of the individuals; either `male` or `female`

Details

These data were obtained from the Annual Vital Statistics Report of Japan, released by the Japanese government every year since 1947. The authors note that "All the members of that cohort have died by the end of the observation period, a procedure referred to as the extinct cohort method". The data were obtained from the Human Mortality Database by the authors. Only positive counts are reported and two records (Misao Okawa and Jiroemon Kimura) are excluded because they do not correspond to the same selection mechanism.

Source

Table extracted from Hanayama & Sibuya (2016).

References

Hanayama, N. and M. Sibuya (2016). Estimating the Upper Limit of Lifetime Probability Distribution, Based on Data of Japanese Centenarians, *The Journals of Gerontology: Series A*, 71(**8**), 1014–1021. doi:10.1093/gerona/glv113

`japanese2` *Japanese survival data (2)*

Description

This data frame is extracted from Table 10.3 from Chapter 10, "Centenarians and Supercentenarians in Japan", in the Monograph Exceptional lifespans. The data were constructed by the extinct cohort method and are stratified by age cohort (five year group, except 1899-1900) and by sex. Note that the family registry system (KOSEKI), introduced in 1872, was standardized in 1886.

Usage

```
japanese2
```

Format

A data frame with 216 rows and 4 variables:

age integer, age (to the smallest year) at death (in years)

bcohort factor, birth cohort

count integer, number of death for cohort at given age

gender factor, the gender of the individuals; either `male` or `female`

Source

Table 10.3

References

Saito, Yasuhiko and Futoshi Ishii, and Jean-Marie Robine (2021). *Centenarians and Supercentenarians in Japan*. In *Exceptional lifespans*, Maier, H., Jeune, B., Vaupel, J. W. (Eds.), Demographic research monographs 17 VII, pp. 125-145. Cham, Springer.

`lpost_elife` *Log posterior distribution with MDI priors*

Description

Log of the posterior distribution for excess lifetime distribution with maximal data information priors.

Usage

```
lpost_elife(
  par,
  time,
  time2 = NULL,
  event = NULL,
  type = c("right", "left", "interval", "interval2"),
  ltrunc = NULL,
  rtrunc = NULL,
  family = c("exp", "gp", "gomp"),
  thresh = 0,
  weights = rep(1, length(time)),
  status = NULL,
  arguments = NULL,
  ...
)
```

Arguments

`par`	vector of parameters, in the following order: scale, rate and shape
`time`	excess time of the event of follow-up time, depending on the value of event
`time2`	ending excess time of the interval for interval censored data only.
`event`	status indicator, normally 0=alive, 1=dead. Other choices are `TRUE`/`FALSE` (`TRUE` for death). For interval censored data, the status indicator is 0=right censored, 1=event at time, 2=left censored, 3=interval censored. Although unusual, the event indicator can be omitted, in which case all subjects are assumed to have experienced an event.

`type`	character string specifying the type of censoring. Possible values are `"right"`, `"left"`, `"interval"`, `"interval2"`.
`ltrunc`	lower truncation limit, default to `NULL`
`rtrunc`	upper truncation limit, default to `NULL`
`family`	string; choice of parametric family
`thresh`	vector of thresholds
`weights`	weights for observations
`status`	integer vector giving status of an observation. If `NULL` (default), this argument is computed internally based on `type`.
`arguments`	a named list specifying default arguments of the function that are common to all `elife` calls
`...`	additional arguments for optimization, currently ignored.

Value

a vector proportional to the log posterior (the sum of the log likelihood and log prior)

`nc_test` *Score test of Northrop and Coleman*

Description

This function computes the score test with the piecewise generalized Pareto distribution under the null hypothesis that the generalized Pareto with a single shape parameter is an adequate simplification. The score test statistic is calculated using the observed information matrix; both hessian and score vector are obtained through numerical differentiation.

Usage

```
nc_test(
  time,
  time2 = NULL,
  event = NULL,
  thresh = 0,
  ltrunc = NULL,
  rtrunc = NULL,
  type = c("right", "left", "interval", "interval2"),
  weights = rep(1, length(time)),
  test = c("score", "lrt"),
  arguments = NULL,
  ...
)
```

Arguments

`time`	excess time of the event of follow-up time, depending on the value of event
`time2`	ending excess time of the interval for interval censored data only.
`event`	status indicator, normally 0=alive, 1=dead. Other choices are `TRUE`/`FALSE` (`TRUE` for death). For interval censored data, the status indicator is 0=right censored, 1=event at time, 2=left censored, 3=interval censored. Although unusual, the event indicator can be omitted, in which case all subjects are assumed to have experienced an event.
`thresh`	a vector of thresholds
`ltrunc`	lower truncation limit, default to `NULL`
`rtrunc`	upper truncation limit, default to `NULL`
`type`	character string specifying the type of censoring. Possible values are `"right"`, `"left"`, `"interval"`, `"interval2"`.
`weights`	weights for observations
`test`	string, either `"score"` for the score test or `"lrt"` for the likelihood ratio test.
`arguments`	a named list specifying default arguments of the function that are common to all `elife` calls
`...`	additional parameters, currently ignored

Details

The score test is much faster and perhaps less fragile than the likelihood ratio test: fitting the piecewise generalized Pareto model is difficult due to the large number of parameters and multimodal likelihood surface.

The reference distribution is chi-square

Value

a data frame with the following variables:

- `thresh`: threshold for the generalized Pareto distribution
- `nexc`: number of exceedances
- `score`: score statistic
- `df`: degrees of freedom
- `pval`: the p-value obtained from the asymptotic chi-square approximation.

Examples

```
set.seed(1234)
n <- 100L
x <- samp_elife(n = n,
                scale = 2,
                shape = -0.2,
                lower = low <- runif(n),
                upper = upp <- runif(n, min = 3, max = 20),
```

```
                type2 = "ltrt",
                family = "gp")
test <- nc_test(
  time = x,
  ltrunc = low,
  rtrunc = upp,
  thresh = quantile(x, seq(0, 0.5, by = 0.1)))
print(test)
plot(test)
```

`nll_elife` *Likelihood for arbitrary censored and truncated data*

Description

Computes the log-likelihood for various parametric models suitable for threshold exceedances. If threshold is non-zero, then only right-censored, observed event time and interval censored data whose timing exceeds the thresholds are kept.

Usage

```
nll_elife(
  par,
  time,
  time2 = NULL,
  event = NULL,
  type = c("right", "left", "interval", "interval2"),
  ltrunc = NULL,
  rtrunc = NULL,
  family = c("exp", "gp", "gomp", "gompmake", "weibull", "extgp", "gppiece",
    "extweibull", "perks", "beard", "perksmake", "beardmake"),
  thresh = 0,
  weights = NULL,
  status = NULL,
  arguments = NULL,
  ...
)
```

Arguments

`par`	vector of parameters, in the following order: scale, rate and shape
`time`	excess time of the event of follow-up time, depending on the value of event
`time2`	ending excess time of the interval for interval censored data only.
`event`	status indicator, normally 0=alive, 1=dead. Other choices are `TRUE`/`FALSE` (`TRUE` for death). For interval censored data, the status indicator is 0=right censored, 1=event at time, 2=left censored, 3=interval censored. Although unusual, the

	event indicator can be omitted, in which case all subjects are assumed to have experienced an event.
`type`	character string specifying the type of censoring. Possible values are `"right"`, `"left"`, `"interval"`, `"interval2"`.
`ltrunc`	lower truncation limit, default to `NULL`
`rtrunc`	upper truncation limit, default to `NULL`
`family`	string; choice of parametric family
`thresh`	vector of thresholds
`weights`	weights for observations
`status`	integer vector giving status of an observation. If `NULL` (default), this argument is computed internally based on `type`.
`arguments`	a named list specifying default arguments of the function that are common to all `elife` calls
`...`	additional arguments for optimization, currently ignored.

Value

log-likelihood values

Examples

```
data(ewsim, package = "longevity")
nll_elife(par = c(5, 0.3),
          family = "gp",
          arguments = ewsim)
```

`npsurv` *Nonparametric maximum likelihood estimation for arbitrary truncation*

Description

The syntax is reminiscent of the Surv function, with additional vectors for left-truncation and right-truncation.

Usage

```
npsurv(
  time,
  time2 = NULL,
  event = NULL,
  type = c("right", "left", "interval", "interval2"),
  ltrunc = NULL,
  rtrunc = NULL,
  weights = NULL,
  arguments = NULL,
  ...
)
```

Arguments

`time`	excess time of the event of follow-up time, depending on the value of event
`time2`	ending excess time of the interval for interval censored data only.
`event`	status indicator, normally 0=alive, 1=dead. Other choices are `TRUE`/`FALSE` (`TRUE` for death). For interval censored data, the status indicator is 0=right censored, 1=event at time, 2=left censored, 3=interval censored. Although unusual, the event indicator can be omitted, in which case all subjects are assumed to have experienced an event.
`type`	character string specifying the type of censoring. Possible values are "`right`", "`left`", "`interval`", "`interval2`".
`ltrunc`	lower truncation limit, default to `NULL`
`rtrunc`	upper truncation limit, default to `NULL`
`weights`	vector of weights, default to `NULL` for equiweighted
`arguments`	a named list specifying default arguments of the function that are common to all `elife` calls
`...`	additional arguments passed to the functions

Value

a list with components

- `xval`: unique ordered values of sets on which the distribution function is defined
- `prob`: estimated probability of failure on intervals
- `convergence`: logical; `TRUE` if the EM algorithm iterated until convergence
- `niter`: logical; number of iterations for the EM algorithm
- `cdf`: nonparametric maximum likelihood estimator of the distribution function

Note

Contrary to the Kaplan-Meier estimator, the mass is placed in the interval [`max(time)`, `Inf`) so the resulting distribution function is not deficient.

See Also

`Surv`

Examples

```
# Toy example with interval censoring and right censoring
# Two observations: A1: [1,3], A2: 4
# Probability of 0.5

test_simple2 <- npsurv(
  time = c(1,4),
  time2 = c(3,4),
  event = c(3,1),
  type = "interval"
)
```

`np_elife` *Nonparametric estimation of the survival function*

Description

The survival function is obtained through the EM algorithm described in Turnbull (1976); censoring and truncation are assumed to be non-informative. The survival function changes only at the `J` distinct exceedances $y_i - u$ and truncation points.

Usage

```
np_elife(
  time,
  time2 = NULL,
  event = NULL,
  type = c("right", "left", "interval", "interval2"),
  thresh = 0,
  ltrunc = NULL,
  rtrunc = NULL,
  tol = 1e-12,
  weights = NULL,
  method = c("em", "sqp"),
  arguments = NULL,
  maxiter = 100000L,
  ...
)
```

Arguments

`time`	excess time of the event of follow-up time, depending on the value of event
`time2`	ending excess time of the interval for interval censored data only.
`event`	status indicator, normally 0=alive, 1=dead. Other choices are `TRUE`/`FALSE` (`TRUE` for death). For interval censored data, the status indicator is 0=right censored, 1=event at time, 2=left censored, 3=interval censored. Although unusual, the event indicator can be omitted, in which case all subjects are assumed to have experienced an event.
`type`	character string specifying the type of censoring. Possible values are `"right"`, `"left"`, `"interval"`, `"interval2"`.
`thresh`	double thresh
`ltrunc`	lower truncation limit, default to `NULL`
`rtrunc`	upper truncation limit, default to `NULL`
`tol`	double, relative tolerance for convergence of the EM algorithm
`weights`	double, vector of weights for the observations
`method`	string, one of `"em"` for expectation-maximization (EM) algorithm or `"sqp"` for sequential quadratic programming with augmented Lagrange multiplie method.

`arguments`	a named list specifying default arguments of the function that are common to all `elife` calls
`maxiter`	integer, maximum number of iterations for the EM algorithm
`...`	additional arguments, currently ignored

Details

The unknown parameters of the model are $p_j (j = 1, \ldots, J)$ subject to the constraint that $\sum_{j=1}^{J} p_j = 1$.

Value

a list with elements

- `cdf`: right-continuous `stepfun` object defined by probabilities
- `time`: matrix of unique values for the Turnbull intervals defining equivalence classes; only those with non-zero probabilities are returned
- `prob`: `J` vector of non-zero probabilities
- `niter`: number of iterations

References

Turnbull, B. W. (1976). *The Empirical Distribution Function with Arbitrarily Grouped, Censored and Truncated Data.* Journal of the Royal Statistical Society. Series B (Methodological) 38(**3**), 290–295.

Gentleman, R. and C. J. Geyer (1994). *Maximum likelihood for interval censored data: Consistency and computation*, Biometrika, 81(**3**), 618–623.

Frydman, H. (1994). *A Note on Nonparametric Estimation of the Distribution Function from Interval-Censored and Truncated Observations*, Journal of the Royal Statistical Society. Series B (Methodological) **56**(1), 71-74.

Examples

```
set.seed(2021)
n <- 20L
# Create fake data
ltrunc <- pmax(0, runif(n, -0.5, 1))
rtrunc <- runif(n, 6, 10)
dat <- samp_elife(n = n,
                  scale = 1,
                  shape = -0.1,
                  lower = ltrunc,
                  upper = rtrunc,
                  family = "gp",
                  type2 = "ltrt")
npi <- np_elife(time = dat,
                rtrunc = rtrunc,
                ltrunc = ltrunc)
print(npi)
```

```
summary(npi)
plot(npi)
```

`plot.elife_ecdf` *Plot empirical distribution function*

Description

Plot empirical distribution function

Usage

```
## S3 method for class 'elife_ecdf'
plot(x, ...)
```

Arguments

`x`	argument of class `elife_ecdf`
`...`	additional arguments for the plot

Value

base R plot of the empirical distribution function

`plot.elife_profile` *Plot profile of endpoint*

Description

Plot profile of endpoint

Usage

```
## S3 method for class 'elife_profile'
plot(x, plot.type = c("base", "ggplot"), plot = TRUE, ...)

autoplot.elife_profile(object, ...)

autoplot.elife_tstab_endpoint(object, ...)
```

Arguments

`x`	an object of class `elife_profile` containing information about the profile likelihood, maximum likelihood and grid of values for the endpoint
`plot.type`	string indicating whether to use base R for plots or `ggplot2`
`plot`	logical; if `TRUE`, creates a plot when `plot.type="ggplot"`. Useful for returning `ggplot` objects without printing the graphs
`...`	additional arguments to pass to `plot`, currently ignored
`object`	object of class `elife_tstab_endpoint`

Value

base R or ggplot object for a plot of the profile log likelihood of the endpoint of the generalized Pareto distribution

---

`plot.elife_tstab_endpoint`

Plot threshold stability plot for endpoint

Description

Plot threshold stability plot for endpoint

Usage

```
## S3 method for class 'elife_tstab_endpoint'
plot(x, plot.type = c("base", "ggplot"), plot = TRUE, ...)
```

Arguments

`x`	an object of class `elife_tstab_endpoint` containing information about the profile likelihood, maximum likelihood and grid of values for the endpoint
`plot.type`	string indicating whether to use base R for plots or `ggplot2`
`plot`	logical; if `TRUE`, creates a plot when `plot.type="ggplot"`. Useful for returning `ggplot` objects without printing the graphs
`...`	additional arguments to pass to `plot`, currently ignored

Value

base R or ggplot object for a plot of the profile log likelihood of the endpoint of the generalized Pareto distribution

---

`samp_elife` *Simulate excess lifetime with truncation or right-censoring*

Description

This function dispatches simulations accounting for potential left-truncation (remove by setting lower to zero). If `type2=ltrt`, simulated observations will be lower than the upper bounds `upper`. If `type2=ltrc`, simulated observations are capped at `upper` and the observation is right-censored (`rcens=TRUE`).

Usage

```
samp_elife(
  n,
  scale,
  rate,
  shape = NULL,
  lower = 0,
  upper = Inf,
  family = c("exp", "gp", "gomp", "gompmake", "weibull", "extgp", "gppiece",
    "extweibull", "perks", "beard", "perksmake", "beardmake"),
  type2 = c("none", "ltrt", "ltrc", "ditrunc")
)
```

Arguments

`n`	sample size
`scale`	scale parameter(s)
`rate`	rate parameter(s)
`shape`	shape parameter(s)
`lower`	vector of lower bounds
`upper`	vector of upper bounds
`family`	string; choice of parametric family
`type2`	string, either `none`, `ltrt` for left- and right-truncated data or `ltrc` for left-truncated right-censored data

Value

either a vector of observations or, if `type2=ltrc`, a list with `n` observations `dat` and a logical vector of the same length with `TRUE` for right-censored observations and `FALSE` otherwise.

Note

As the tails of the Gompertz and Gompertz–Makeham models decrease exponentially fast, the method fails in the rare event case if the lower bound is too large (say larger than the 99.99

Examples

```
set.seed(1234)
n <- 500L
# Simulate interval truncated data
x <- samp_elife(n = n,
                scale = 2,
                shape = 1.5,
                lower = low <- runif(n),
                upper = upp <- runif(n, min = 3, max = 15),
                type2 = "ltrt",
                family = "weibull")
coef(fit_elife(
   time = x,
   ltrunc = low,
   rtrunc = upp,
   family = "weibull"))
# Simulate left-truncated right-censored data
x <- samp_elife(n = n,
                scale = 2,
                shape = 1.5,
                lower = low <- runif(n),
                upper = upp <- runif(n, min = 3, max = 15),
                type2 = "ltrc",
                family = "gomp")
#note that the return value is a list...
coef(fit_elife(
   time = x$dat,
   ltrunc = low,
   event = !x$rcens,
   family = "gomp"))
```

`test_elife` *Likelihood ratio test for covariates*

Description

This function fits separate models for each distinct value of the factor `covariate` and computes a likelihood ratio test to test whether there are significant differences between groups.

Usage

```
test_elife(
  time,
  time2 = NULL,
  event = NULL,
  covariate,
  thresh = 0,
  ltrunc = NULL,
  rtrunc = NULL,
```

```
  type = c("right", "left", "interval", "interval2"),
  family = c("exp", "gp", "weibull", "gomp", "gompmake", "extgp", "extweibull", "perks",
    "perksmake", "beard", "beardmake"),
  weights = rep(1, length(time)),
  arguments = NULL,
  ...
)
```

Arguments

`time`	excess time of the event of follow-up time, depending on the value of event
`time2`	ending excess time of the interval for interval censored data only.
`event`	status indicator, normally 0=alive, 1=dead. Other choices are `TRUE`/`FALSE` (`TRUE` for death). For interval censored data, the status indicator is 0=right censored, 1=event at time, 2=left censored, 3=interval censored. Although unusual, the event indicator can be omitted, in which case all subjects are assumed to have experienced an event.
`covariate`	vector of factors, logical or integer whose distinct values define groups
`thresh`	vector of thresholds
`ltrunc`	lower truncation limit, default to `NULL`
`rtrunc`	upper truncation limit, default to `NULL`
`type`	character string specifying the type of censoring. Possible values are `"right"`, `"left"`, `"interval"`, `"interval2"`.
`family`	string; choice of parametric family
`weights`	weights for observations
`arguments`	a named list specifying default arguments of the function that are common to all `elife` calls
`...`	additional arguments for optimization, currently ignored.

Value

a list with elements

- `stat` likelihood ratio statistic
- `df` degrees of freedom
- `pval` the p-value obtained from the asymptotic chi-square approximation.

Examples

```
test <- with(subset(dutch, ndays > 39082),
 test_elife(
 time = ndays,
 thresh = 39082L,
 covariate = gender,
 ltrunc = ltrunc,
 rtrunc = rtrunc,
 family = "exp"))
 test
```

`tstab` *Threshold stability plots*

Description

The generalized Pareto and exponential distribution are threshold stable. This property, which is used for extrapolation purposes, can also be used to diagnose goodness-of-fit: we expect the parameters ξ and $\tilde{\sigma} = \sigma + \xi u$ to be constant over a range of thresholds. The threshold stability plot consists in plotting maximum likelihood estimates with pointwise confidence interval. This function handles interval truncation and right-censoring.

Usage

```
tstab(
  time,
  time2 = NULL,
  event = NULL,
  thresh = 0,
  ltrunc = NULL,
  rtrunc = NULL,
  type = c("right", "left", "interval", "interval2"),
  family = c("gp", "exp"),
  method = c("wald", "profile"),
  level = 0.95,
  plot = TRUE,
  plot.type = c("base", "ggplot"),
  which.plot = c("scale", "shape"),
  weights = NULL,
  arguments = NULL,
  ...
)
```

Arguments

`time`	excess time of the event of follow-up time, depending on the value of event
`time2`	ending excess time of the interval for interval censored data only.
`event`	status indicator, normally 0=alive, 1=dead. Other choices are `TRUE`/`FALSE` (`TRUE` for death). For interval censored data, the status indicator is 0=right censored, 1=event at time, 2=left censored, 3=interval censored. Although unusual, the event indicator can be omitted, in which case all subjects are assumed to have experienced an event.
`thresh`	vector of thresholds
`ltrunc`	lower truncation limit, default to `NULL`
`rtrunc`	upper truncation limit, default to `NULL`
`type`	character string specifying the type of censoring. Possible values are "`right`", "`left`", "`interval`", "`interval2`".

`family`	string; distribution, either generalized Pareto (`gp`) or exponential (`exp`)
`method`	string; the type of pointwise confidence interval, either Wald (`wald`) or profile likelihood (`profile`)
`level`	probability level for the pointwise confidence intervals
`plot`	logical; should a plot be returned alongside with the estimates? Default to `TRUE`
`plot.type`	string; either `base` for base R plots or `ggplot` for `ggplot2` plots
`which.plot`	string; which parameters to plot;
`weights`	weights for observations
`arguments`	a named list specifying default arguments of the function that are common to all `elife` calls
`...`	additional arguments for optimization, currently ignored.

Details

The shape estimates are constrained

Value

an invisible list with pointwise estimates and confidence intervals for the scale and shape parameters

See Also

`tstab.gpd` from package `mev`, `gpd.fitrange` from package `ismev` or `tcplot` from package `evd`, among others.

Examples

```
set.seed(1234)
n <- 100L
x <- samp_elife(n = n,
                scale = 2,
                shape = -0.2,
                lower = low <- runif(n),
                upper = upp <- runif(n, min = 3, max = 20),
                type2 = "ltrt",
                family = "gp")
tstab_plot <- tstab(time = x,
                    ltrunc = low,
                   rtrunc = upp,
                   thresh = quantile(x, seq(0, 0.5, length.out = 4)))
plot(tstab_plot, plot.type = "ggplot")
```

Index

∗ **datasets**
    dutch, 6
    ewsim, 8
    idlmetadata, 11
    japanese, 11
    japanese2, 12

autoplot.elife_northropcoleman, 2
autoplot.elife_par, 3
autoplot.elife_profile
        (plot.elife_profile), 21
autoplot.elife_tstab, 5
autoplot.elife_tstab_endpoint
        (plot.elife_profile), 21

dutch, 6

endpoint.profile, 7
ewsim, 8

fit_elife, 9

idlmetadata, 11

japanese, 11
japanese2, 12

lpost_elife, 13

nc_test, 14
nll_elife, 16
np_elife, 19
npsurv, 17

plot.elife_ecdf, 21
plot.elife_northropcoleman
        (autoplot.elife_northropcoleman),
        2
plot.elife_par (autoplot.elife_par), 3
plot.elife_profile, 21
plot.elife_tstab
        (autoplot.elife_tstab), 5
plot.elife_tstab_endpoint, 22

samp_elife, 23
Surv, *17, 18*

test_elife, 24
tstab, 26