Package ‘vitality’

October 12, 2022

Title Fitting Routines for the Vitality Family of Mortality Models

Version 1.3

Date 2018-05-13

Author Gregor Passolt, James J. Anderson, Ting Li, David H. Salinger, David J. Sharrow

Maintainer David J. Sharrow <`dsharrow@uw.edu`>

Depends stats

Description Provides fitting routines for four versions of the Vitality family of mortality models.

License GPL (>= 2)

Collate 'dataPrep.R' 'vitality.4p.R' 'vitality.6p.R' 'vitality.k.R' 'vitality.ku.R' 'vitality.utils.R' 'data_documentation.R' 'density.R' 'mortality_rate.R' 'package_documentation.R'

NeedsCompilation no

Repository CRAN

Date/Publication 2018-05-13 21:26:30 UTC

R topics documented:

`daphnia` *Sample Daphnia Data*

Description

Sample survival data for daphnia. Columns include "days" and "lx" (cumulative survival proportion by day).

Format

data frame

Source

http://cbr.washington.edu/analysis/vitality

Anderson, J.J. (2000). "A vitality-based model relating stressors and environmental properties to organism survival." Ecological Monographs 70(3):445-470 (Figure 5)

`dataPrep` *Function for data preparation*

Description

Function to deal with NAs, right truncated data, and datatype (i.e. cumulative survival or incremental mortality).

Usage

```
dataPrep(time, sdata, datatype, rc.data,
  returnMatrix = FALSE)
```

Arguments

`time`	A vector of observation dates
`sdata`	A vector of survival data of the same length as `time`
`datatype`	either `"CUM"` for cumulative or `"INC"` for incremental
`rc.data`	Boolean. Is data right-censored?
`returnMatrix`	Boolean. False returns a data frame, true returns a matrix. (as in the original), if "matrix" returns a matrix instead, with the "rc.data" column being 0 for FALSE, 1 for TRUE, or 2 for TF

Details

This function is designed for use in the primary vitality model fitting functions in this package. See package documentation.

Value

Returns a data.frame or matrix with columns time, sfract, x1, x2, Ni (incremental survival fraction), rc.data.

`ft.4p` *Density function for 3-parameter (r, s, u)*

Description

This function is used in the calculation of the fitted intrinsic (`mu.vd1.4p`) and total (`mu.vd.4p`) mortality rate in the 4-parameter model.

Usage

```
ft.4p(xx, r, s, u)
```

Arguments

`xx`	age
`r`	r value
`s`	s value
`u`	u value

Value

density

See Also

`vft.4p`, `ft.6p`

`ft.6p` *Density function for 2-parameters (r, s)*

Description

This function is used in the calculation of the fitted intrinsic (`mu.vd1.6p`) and total (`mu.vd.6p`) mortality rate in the 6-parameter model.

Usage

```
ft.6p(xx, r, s)
```

Arguments

`xx`	age
`r`	r value
`s`	s value

Value

density

See Also

`vft.6p`

`indexFinder` *Finds the first value of a vector that is less than a value.*

Description

For use in the primary vitality model fitting functions in this package. See package documentation.

Usage

```
indexFinder(x, val)
```

Arguments

`x`	Vector to search
`val`	Threshold

Value

Gives the index of the first value of x that is <= val. returns -1 if no value satisfies the condition

---

`logLikelihood.4p` *Log likelihood of 2-process 4-parameter model*

Description

Gives the log likelihood of 2-process 6 parameter vitality model.

Usage

```
logLikelihood.4p(par, xx1, xx2, NNi)
```

Arguments

`par`	vector of parameter(r, s, lambda, beta)
`xx1`	xx1 vector
`xx2`	xx2 vector
`NNi`	survival fractions

Details

For use in `vitality.4p`.

Value

log likelihood

See Also

`logLikelihood.6p`

`logLikelihood.6p` *Log likelihood of 2-process 6-parameter vitality model*

Description

Gives the log likelihood of 6-parameter vitality model.

Usage

```
logLikelihood.6p(par, xx1, xx2, NNi)
```

Arguments

`par`	vector of parameters (r, s, lambda, beta, alpha, gamma)
`xx1`	xx1 vector
`xx2`	xx2 vector
`NNi`	survival fractions

Details

For use in `vitality.6p`.

Value

log likelihood

See Also

`vitality.6p`

`logLikelihood.k` *Log likelihood of 3-parameter (r,s,k) model*

Description

Gives the log likelihood of 3-parameter vitality model.

Usage

```
logLikelihood.k(par, xx1, xx2, NNi)
```

Arguments

`par`	vector of parameters (r, s, k)
`xx1`	xx1 vector
`xx2`	xx2 vector
`NNi`	survival fractions

Details

For use in `vitality.k`.

Value

log likelihood

See Also

`vitality.k`

`logLikelihood.ku` *Log likelihood of 4-parameter (r,s,k,u) model*

Description

Gives the log likelihood of 4-parameter vitality model.

Usage

```
logLikelihood.ku(par, xx1, xx2, NNi)
```

Arguments

`par`	vector of parameters (r, s, k, u)
`xx1`	xx1 vector
`xx2`	xx2 vector
`NNi`	survival fractions

Details

For use in `vitality.ku`.

Value

log likelihood

See Also

`vitality.ku`

`mu.vd.4p` *Total mortality rate for the 2-process 4-parameter vitality model*

Description

Gives the total age-specific mortality rates for a given set of the four parameters. See `mu.vd1.4p` for calculation of intrinsic age-specific mortality rates. See `mu.vd2.4p` for calculation of extrinsic age-specific mortality rates.

Usage

```
mu.vd.4p(t, r, s, lambda, beta)
```

Arguments

`t`	age
`r`	r value
`s`	s value
`lambda`	lambda value
`beta`	beta value

Value

Total age-specific mortality rates

See Also

`mu.vd1.4p`, `mu.vd2.4p`

`mu.vd.6p` *Total mortality rate for the 2-process 6-parameter vitality model*

Description

Gives the total age-specific mortality rates for a given set of the six parameters. See `mu.vd1.6p` for calculation of intrinsic age-specific mortality rates. See `mu.vd2.6p` for calculation of extrinsic age-specific mortality rates.

Usage

```
mu.vd.6p(t, r, s, lambda, beta, gamma, alpha)
```

Arguments

`t`	age
`r`	r value
`s`	s value
`lambda`	lambda value
`beta`	beta value
`gamma`	gamma value
`alpha`	alpha value

Value

Total age-specific mortality rates

See Also

`mu.vd1.6p`, `mu.vd2.6p`

`mu.vd1.4p` *Intrinsic mortality rate for the 2-process 4-parameter vitality model*

Description

Gives the intrinsic age-specific mortality rates for a given set of `r` and `s`, the intrinsic parameters.

Usage

```
mu.vd1.4p(x, r, s)
```

Arguments

`x`	age
`r`	r value
`s`	s value

Value

Intrinsic age-specific mortality rates

See Also

`mu.vd.4p`, `mu.vd2.4p`

`mu.vd1.6p` *Intrinsic mortality rate for the 2-process 6-parameter vitality model*

Description

Gives the intrinsic age-specific mortality rates for a given set of `r` and `s`, the intrinsic parameters.

Usage

```
mu.vd1.6p(x, r, s)
```

Arguments

`x`	age
`r`	r value
`s`	s value

Value

Vector of intrinsic age-specific mortality rates at age `x`

See Also

`mu.vd.6p`, `mu.vd2.6p`

`mu.vd2.4p` *Extrinsic mortality rate for the 2-process 4-parameter vitality model*

Description

Gives the extrinsic age-specific mortality rates for a given set of `r` and the extrinsic parameters.

Usage

```
mu.vd2.4p(x, r, lambda, beta)
```

Arguments

`x`	age
`r`	r value
`lambda`	lambda value
`beta`	beta value

Value

Extrinsic age-specific mortality rates

See Also

`mu.vd.4p`, `mu.vd1.4p`

`mu.vd2.6p` *Extrinsic mortality rate for the 2-process 6-parameter vitality model*

Description

Gives the extrinsic age-specific mortality rates for a given set of `r` and the extrinsic parameters.

Usage

```
mu.vd2.6p(x, r, lambda, beta, gamma, alpha)
```

Arguments

`x`	age
`r`	r value
`lambda`	lambda value
`beta`	beta value
`gamma`	gamma value
`alpha`	alpha value

Value

Vector of extrinsic age-specific mortality rates at ages `x`

See Also

`mu.vd.6p`, `mu.vd1.6p`, `mu.vd3.6p`, `mu.vd4.6p`

`mu.vd3.6p` *Adult extrinsic mortality rate for the 2-process 6-parameter vitality model*

Description

Gives the extrinsic age-specific mortality rates for a given set of `r` and the adult extrinsic parameters.

Usage

```
mu.vd3.6p(x, r, lambda, beta)
```

Arguments

`x`	age
`r`	r value
`lambda`	lambda value
`beta`	beta value

Value

Vector of adult extrinsic age-specific mortality rates at ages `x`

See Also

`mu.vd.6p`, `mu.vd1.6p`, `mu.vd2.6p`, `mu.vd4.6p`

`mu.vd4.6p` *Childhood extrinsic mortality rate for the 2-process 6-parameter vitality model*

Description

Gives the childhood extrinsic age-specific mortality rates for a given set of the childhood extrinsic parameters.

Usage

```
mu.vd4.6p(x, gamma, alpha)
```

Arguments

`x`	age
`gamma`	gamma value
`alpha`	alpha value

Value

Vector of childhood extrinsic age-specific mortality rates at ages `x`

See Also

`mu.vd.6p`, `mu.vd1.6p`, `mu.vd2.6p`, `mu.vd3.6p`

`plotting.4p` *Plotting function for 2-process 4-parameter vitality model*

Description

This function plots the estimated results from the 4 parameter vitality model. It is used within the function `vitality.4p`.

Usage

```
plotting.4p(r.final, s.final, lambda.final, beta.final,
  mlv, time, sfract, x1, x2, Ni, pplot, Iplot, Mplot,
  tlab, rc.data)
```

Arguments

`r.final`	r estimate
`s.final`	s estimate
`lambda.final`	lambda estimate
`beta.final`	beta estimate
`mlv`	TODO mlv
`time`	time vector
`sfract`	survival fraction
`x1`	Time 1
`x2`	Time 2
`Ni`	Initial population
`pplot`	Boolean. Plot cumulative survival fraction?
`Iplot`	Boolean. Plot incremental survival?
`Mplot`	Boolean. Plot mortality rate? Note: plotted mortality data is derived from survival data and is approximate
`tlab`	Character, label for time axis
`rc.data`	Boolean, right-censored data?

Details

See `vitality.4p` for further description of function arguments.

Only one of `Iplot` or `Mplot` should be set to `TRUE` at once.

See Also

`vitality.4p`, `mu.vd.4p`, `mu.vd1.4p`, `mu.vd2.4p`

`plotting.6p` *Plotting function for 2-process 6-parameter vitality model*

Description

This function plots the estimated results from the 6 parameter vitality model. It is used within the function `vitality.6p`.

Usage

```
plotting.6p(r.final, s.final, lambda.final, beta.final,
  gamma.final, alpha.final, mlv, time, sfract, x1, x2, Ni, pplot, Iplot, Mplot,
   tlab, rc.data)
```

Arguments

`r.final`	r estimate
`s.final`	s estimate
`lambda.final`	lambda estimate
`beta.final`	beta estimate
`gamma.final`	gamma estimate
`alpha.final`	alpha estimate
`mlv`	TODO mlv
`time`	time vector giving the ages at which calculate the estimated value
`sfract`	observed survival fraction
`x1`	Time 1
`x2`	Time 2
`Ni`	Initial population
`pplot`	Boolean. Plot cumulative survival fraction?
`Iplot`	Boolean. Plot incremental survival?
`Mplot`	Boolean. Plot mortality rate? Note: plotted mortality data is derived from survival data and is approximate
`tlab`	Character, label for time axis
`rc.data`	Boolean, right-censored data?

Details

See `vitality.6p` for further description of function arguments.

Only one of `Iplot` or `Mplot` should be set to `TRUE` at once.

See Also

`vitality.6p`, `mu.vd.6p`, `mu.vd1.6p`, `mu.vd2.6p`, `mu.vd3.6p`, `mu.vd4.6p`

`plotting.k` *Plotting function for 3-parameter vitality model*

Description

This function plots the estimated results from the 3 parameter vitality model. It is used within the function `vitality.k`.

Usage

```
plotting.k(r.final,s.final,k.final,mlv,time,sfract,x1,x2,Ni,
pplot,tlab,lplot,cplot,Iplot,gfit,rc.data)
```

Arguments

`r.final`	r estimate
`s.final`	s estimate
`k.final`	k estimate
`mlv`	TODO mlv
`time`	time vector giving the ages at which calculate the estimated value
`sfract`	observed survival fraction
`x1`	Time 1
`x2`	Time 2
`Ni`	Initial population
`pplot`	Boolean. Plot cumulative survival fraction?
`Iplot`	Boolean. Plot incremental survival?
`lplot`	Boolean. Plot likelihood functions? Provides likelihood function plotting. Defaults to `FALSE`. Note: these plots are not "likelihood profiles" in that while one parameter is varied, the others are held fixed, rather than re-optimized. (must also have pplot=T.)
`cplot`	Boolean. Plot likelihood contour plot? Provides a likelihood contour plot for a range of r and s values (can be slow so default is `FALSE`). Must also have `lplot=T` and `pplot=T` to get contour plots.
`tlab`	Character, label for time axis. Defaults to "days".
`gfit`	Provides a Pearson C type test for goodness of fit. Default is `FALSE`. Must provide the initial study population to compute goodness of fit.
`rc.data`	Boolean, right-censored data?

Details

See `vitality.k` for further description of function arguments.

See Also

`vitality.k`

`plotting.ku` *Plotting function for 4-parameter vitality model*

Description

This function plots the estimated results from the 4 parameter vitality model. It is used within the function `vitality.ku`.

Usage

```
plotting.ku(r.final,s.final,k.final,u.final,mlv,time,sfract,x1,x2,Ni,
  pplot,tlab,lplot,cplot,Iplot,gfit)
```

Arguments

`r.final`	r estimate
`s.final`	s estimate
`k.final`	k estimate
`u.final`	u estimate
`mlv`	TODO mlv
`time`	time vector giving the ages at which calculate the estimated value
`sfract`	observed survival fraction
`x1`	Time 1
`x2`	Time 2
`Ni`	Initial population
`pplot`	Boolean. Plot cumulative survival fraction?
`Iplot`	Boolean. Plot incremental survival?
`lplot`	Boolean. Plot likelihood functions? Provides likelihood function plotting. Defaults to `FALSE`. Note: these plots are not "likelihood profiles" in that while one parameter is varied, the others are held fixed, rather than re-optimized. (must also have pplot=T.)
`cplot`	Boolean. Plot likelihood contour plot? Provides a likelihood contour plot for a range of r and s values (can be slow so default is `FALSE`). Must also have `lplot=T` and `pplot=T` to get contour plots.
`tlab`	Character, label for time axis. Defaults to "days".
`gfit`	Provides a Pearson C type test for goodness of fit. Default is `FALSE`. Must provide the initial study population to compute goodness of fit.

Details

See `vitality.ku` for further description of function arguments.

See Also

`vitality.ku`

`rainbow_trout_for_k` *Sample Rainbow Trout Data*

Description

Sample survival data for rainbow trout. Columns include "days" and "survival" (cumulative survival proportion by day).

Format

matrix

Source

http://cbr.washington.edu/analysis/vitality

`stdErr.4p` *Standard errors for 4-parameters: r, s, lambda, beta*

Description

Gives the standard errors for the 4 parameter model. Primarily used within `vitality.4p`.

Usage

```
stdErr.4p(r, s, k, u, x1, x2, Ni, pop)
```

Arguments

`r`	r value
`s`	s value
`k`	lambda value
`u`	beta value
`x1`	age 1 (corresponding 1:(t-1) and 2:t)
`x2`	age 2
`Ni`	survival fraction
`pop`	initial population (total population of the study)

Value

standard error for r, s, lambda, beta

Note

if k <= 0, cannot find standard error for k

See Also

`vitality.4p`

`stdErr.6p` *Standard errors for 6-parameters: r, s, lambda, beta, gamma, alpha*

Description

Gives the standard errors for the 6 parameter model. Primarily used within `vitality.6p`.

Usage

```
stdErr.6p(r, s, k, u, g, a, x1, x2, Ni, pop)
```

Arguments

`r`	r value
`s`	s value
`k`	lambda value
`u`	beta value
`g`	gamma value
`a`	alpha value
`x1`	age 1 (corresponding 1:(t-1))
`x2`	age 2 (corresponding 2:t
`Ni`	age-specific survival fractions
`pop`	initial population (total population of the study)

Value

standard error for r, s, lambda, beta, gamma, and alpha.

Note

if k <= 0, cannot find standard error for k

See Also

`vitality.6p`

`stdErr.k` *Standard errors for 3-parameters: r, s, k*

Description

Gives the standard errors for the 3 parameter model. Primarily used within `vitality.k`.

Usage

```
stdErr.k(r, s, k, x1, x2, Ni, pop)
```

Arguments

`r`	r value
`s`	s value
`k`	k value
`x1`	age 1 (corresponding 1:(t-1))
`x2`	age 2 (corresponding 2:t
`Ni`	age-specific survival fractions
`pop`	initial population (total population of the study)

Value

standard error for r, s, k.

Note

k is restricted to be >0.

See Also

`vitality.k`

`stdErr.ku` *Standard errors for 4-parameters: r, s, k, u*

Description

Gives the standard errors for the 4 parameter model. Primarily used within `vitality.ku`.

Usage

```
stdErr.ku(r, s, k, u, x1, x2, Ni, pop)
```

Arguments

`r`	r value
`s`	s value
`k`	k value
`u`	u value
`x1`	age 1 (corresponding 1:(t-1))
`x2`	age 2 (corresponding 2:t
`Ni`	age-specific survival fractions
`pop`	initial population (total population of the study)

Value

standard error for r, s, k, u.

Note

k is restricted to be >0.

See Also

`vitality.ku`

`SurvFn.4p` *The cumulative survival distribution function for 2-process 4-parameter vitality model*

Description

Gives the cumulative survival proportions at `xx` from all processes for a given set of parameter values.

Usage

```
SurvFn.4p(xx, r, s, lambda, beta)
```

Arguments

`xx`	vector of ages
`r`	r value
`s`	s value
`lambda`	lambda value
`beta`	beta value

Details

Used within `vitality.4p` for estimating model parameters based on the observed cumulative survival function.

Value

vector of cumulative survival proportions at `xx` from all processes

See Also

`vitality.4p`, `survProbInc.4p`, `SurvFn.in.4p`, `SurvFn.ex.4p`

`SurvFn.6p` *The cumulative survival distribution function for 2-process 6-parameter vitality model*

Description

Gives the cumulative survival proportions at `xx` from all processes for a given set of parameter values.

Usage

```
SurvFn.6p(xx, r, s, lambda, beta, gamma, alpha)
```

Arguments

`xx`	vector of ages
`r`	r value
`s`	s value
`lambda`	lambda value
`beta`	beta value
`gamma`	gamma value
`alpha`	alpha value

Details

Used within `vitality.6p` for estimating model parameters based on the observed cumulative survival function.

Value

vector of cumulative survival proportions at `xx` from all processes

See Also

`vitality.6p`, `survProbInc.6p`, `SurvFn.in.6p`, `SurvFn.ex.6p`

`SurvFn.ex.4p` *The extrinsic cumulative survival distribution function for 2-process 4-parameter vitality model*

Description

Gives the cumulative survival proportions at `xx` from extrinsic process for a given set of parameter values.

Usage

```
SurvFn.ex.4p(xx, r, s, lambda, beta)
```

Arguments

`xx`	vector of ages
`r`	r value
`s`	s value
`lambda`	lambda value
`beta`	beta value

Details

Used within `vitality.4p` for estimating model parameters based on the observed cumulative survival function.

Value

vector of cumulative survival proportions at `xx` from extrinsic process

See Also

`vitality.4p`, `survProbInc.4p`, `SurvFn.4p`, `SurvFn.in.4p`

`SurvFn.ex.6p` *The extrinsic cumulative survival distribution function for 2-process 6-parameter vitality model*

Description

Gives the cumulative survival proportions at `xx` from the extrinsic processes for a given set of parameter values.

Usage

```
SurvFn.ex.6p(xx, r, s, lambda, beta, gamma, alpha)
```

Arguments

`xx`	vector of ages
`r`	r value
`s`	s value
`lambda`	lambda value
`beta`	beta value
`gamma`	gamma value
`alpha`	alpha value

Details

Used within `vitality.6p` for estimating model parameters based on the observed cumulative survival function.

Value

vector cumulative survival proportions at `xx` from extrinsic processes

See Also

`vitality.6p`, `survProbInc.6p`, `SurvFn.6p`

`SurvFn.h.4p` *Intrinsic cumulative survival distribution for 4 parameter model*

Description

Gives the intrinsic cumulative survival distribution at `xx`.

Usage

```
SurvFn.h.4p(xx, r, s, u)
```

Arguments

`xx`	vector of ages
`r`	r value
`s`	s value
`u`	u value

Details

For use in `vitality.4p`.

Value

intrinsic cumulative survival distribution

See Also

`vitality.4p`

`SurvFn.h.6p` *Intrinsic cumulative survival distribution for 6 parameter model*

Description

Gives the intrinsic cumulative survival distribution at `xx`.

Usage

```
SurvFn.h.6p(xx, r, s)
```

Arguments

`xx`	vector of ages
`r`	r value
`s`	s value

Details

For use in `vitality.6p`.

Value

intrinsic cumulative survival distribution

See Also

`vitality.6p`

`SurvFn.in.4p` *The intrinsic cumulative survival distribution function for 2-process 4-parameter vitality model*

Description

Gives the cumulative survival proportions at `xx` from intrinsic process for a given set of parameter values.

Usage

```
SurvFn.in.4p(xx, r, s)
```

Arguments

`xx`	vector of ages
`r`	r value
`s`	s value

Details

Used within `vitality.4p` for estimating model parameters based on the observed cumulative survival function.

Value

vector of cumulative survival proportions at `xx` from intrinsic process

See Also

`vitality.4p`, `survProbInc.4p`, `SurvFn.4p`, , `SurvFn.ex.4p`

`SurvFn.in.6p` *The intrinsic cumulative survival distribution function for 2-process 6-parameter vitality model*

Description

Gives the cumulative survival proportions at `xx` from the intrinsic process for a given set of parameter values.

Usage

```
SurvFn.in.6p(xx, r, s)
```

Arguments

`xx`	vector of ages
`r`	r value
`s`	s value

Details

Used within `vitality.6p` for estimating model parameters based on the observed cumulative survival function.

Value

vector of cumulative survival proportions at `xx` from intrinsic process

See Also

`vitality.6p`, `survProbInc.6p`, `SurvFn.6p`

`SurvFn.k` *The cumulative survival distribution function for 3-parameter vitality model*

Description

Gives the cumulative survival proportions at `xx` for a given set of parameter values.

Usage

```
SurvFn.k(xx, r, s, k)
```

Arguments

`xx`	vector of ages
`r`	r value
`s`	s value
`k`	k value

Details

Used within `vitality.k` for estimating model parameters based on the observed cumulative survival function.

Value

cumulative survival proportions at `xx`

See Also

`vitality.k`, `survProbInc.k`

`SurvFn.ku` *The cumulative survival distribution function for 4-parameter vitality model*

Description

Gives the cumulative survival proportions at `xx` for a given set of parameter values.

Usage

```
SurvFn.ku(xx, r, s, k, u)
```

Arguments

`xx`	vector of ages
`r`	r value
`s`	s value
`k`	k value
`u`	u value

Details

Used within `vitality.ku` for estimating model parameters based on the observed cumulative survival function.

Value

cumulative survival proportions at `xx`

See Also

`vitality.ku`, `survProbInc.ku`

`survProbInc.4p` *Incremental survival probability for 2-process 4-parameter model*

Description

Calculates the incremental survival probabilities (between xx1 and xx2) for 2-process 4-parameter model.

Usage

```
survProbInc.4p(r, s, lambda, beta, xx1, xx2)
```

Arguments

`r`	r value
`s`	s value
`lambda`	lambda value
`beta`	beta value
`xx1`	xx1 vector
`xx2`	xx2 vector

Details

For use in `vitality.4p`.

Value

incremental survival probabilities

See Also

`vitality.4p`, `logLikelihood.4p`

`survProbInc.6p` *Incremental survival probability for 2-process 6-parameter model*

Description

Calculates the incremental survival probabilities (between xx1 and xx2) for 2-process 6-parameter model.

Usage

```
survProbInc.6p(r, s, lambda, beta, gamma, alpha, xx1, xx2)
```

Arguments

`r`	r value
`s`	s value
`lambda`	lambda value
`beta`	beta value
`gamma`	gamma value
`alpha`	alpha value
`xx1`	xx1 vector
`xx2`	xx2 vector

Details

For use in `vitality.6p`.

Value

incremental survival probabilities

See Also

`vitality.6p`, `logLikelihood.6p`

`survProbInc.k` *Incremental survival probability for 3-parameter model*

Description

Calculates the incremental survival probabilities (between xx1 and xx2) for 3-parameter model.

Usage

```
survProbInc.k(r, s, k, xx1, xx2)
```

Arguments

`r`	r value
`s`	s value
`k`	k value
`xx1`	xx1 vector
`xx2`	xx2 vector

Details

For use in `vitality.k`.

Value

incremental survival probabilities

See Also

`vitality.k`, `logLikelihood.k`

`survProbInc.ku` *Incremental survival probability for 4-parameter model*

Description

Calculates the incremental survival probabilities (between xx1 and xx2) for 4-parameter model.

Usage

```
survProbInc.ku(r, s, k, u, xx1, xx2)
```

Arguments

`r`	r value
`s`	s value
`k`	k value
`u`	u value
`xx1`	xx1 vector
`xx2`	xx2 vector

Details

For use in `vitality.ku`.

Value

incremental survival probabilities

See Also

`vitality.ku`, `logLikelihood.ku`

`swedish_females` *Swedish Female Mortality Data*

Description

Period life table for Swedish females in the year 2000. Columns follow standard life-table naming conventions.

Format

A `data.frame` object

Source

Human Mortality Database

`vft.4p` *Vectorized density function*

Description

This function is used in the calculation of the fitted intrinsic (`mu.vd1.4p`) and total (`mu.vd.4p`) mortality rate in the 4-parameter model.

Usage

```
vft.4p(xx, r, s, u)
```

Arguments

`xx`	vector of ages
`r`	r value
`s`	s value
`u`	u value

Value

vector of densities

See Also

`ft.4p`

`vft.6p` *Vectorized density function*

Description

This function is used in the calculation of the fitted intrinsic (`mu.vd1.6p`) and total (`mu.vd.6p`) mortality rate in the 6-parameter model.

Usage

```
vft.6p(xx, r, s)
```

Arguments

`xx`	vector of ages
`r`	r value
`s`	s value

Value

vector of densities

See Also

`ft.6p`

`vitality` *Fitting routines for the Vitality family of mortality models.*

Description

This package provides support for fitting the vitality family of mortality models that characterize mortality in terms of the loss vitality, an abstract measure of survival capacity. Mortality occurs by two processes. Intrinsic mortality occurs when vitality is depleted by stochastic losses. Extrinsic mortality occurs when a random external challenge exceeds the available vitality. The package contains four model versions:

- `vitality.k` is a 3-parameter model. Intrinsic mortality is characterized by the mean (r) and variability (s) in vitality loss rate. Extrinsic mortality is characterized by the frequency (k) of lethal random challenges. Model is appropriate to animal mortality data (e.g. Anderson 2000).
- `vitality.ku` is a 4-parameter model. Intrinsic mortality is characterized by the mean (r) and variability (s) in the vitality loss rate and the standard deviation of initial vitality (u). Extrinsic mortality is characterized by the frequency (k) of lethal random challenges. Model is appropriate to animal mortality data (e.g. Li and Anderson 2009).

- `vitality.4p` is a 4-parameter model. Intrinsic mortality is characterized by the mean (r) and variability (s) in the vitality loss rate. Extrinsic mortality is characterized by random challenges of frequency (lambda) and random magnitude (beta) exceeding the remaining average vitality. Model is appropriate to adult human mortality data (e.g. Li and Anderson 2013).
- `vitality.6p` is a 6-parameter model. Intrinsic mortality is characterized by the mean (r) and variability (s) in the vitality loss rate. Adult extrinsic mortality is characterized by random challenges of frequency (lambda) and random magnitude (beta) exceeding the remaining average vitality. Child extrinsic mortality is characterized by childhood challenges of frequency (gamma) exceeding childhood vitality development rate (alpha). Model is appropriate to full lifespan of human mortality data (e.g. Anderson and Li 2015).

Model parameters are estimated on survival or mortality rate data using maximum log likelihood methods based on Salinger et al. (2003).

Version 1.1 adds the versions `vitality.k`, `vitality.ku` and `vitality.6p` to the original code for the `vitality.4p` model previously designated vitality.2ps.

Version 1.2 makes previously invisible functions to produce the survival and mortality rate functions from a given set of parameters (e.g. `SurvFn.4p`, `SurvFn.6p`, `mu.vd.4p`, `mu.vd.6p`) usable. The child mortality rate formula in `mu.vd.6p`, `mu.vd2.6p`, and `mu.vd4.6p` has been updated to improve fit.

Details

Support for package development was provided by the National Institute of Ageing Grant 1R21AG046760-01, the Bonneville Power Administration, and the University of Washington Center for Statistics and the Social Sciences and Center for Studies in Demography and Ecology.

References

- Anderson, J.J. (2000). "A vitality-based model relating stressors and environmental properties to organism survival." Ecological Monographs 70(3):445-470.
- Anderson, J.J. and Li, T. (2015). "A two-process mortality model with extensions to juvenile mortality, populations and evolution." Population Association of America Annual Meeting 2015 http://paa2015.princeton.edu/abstracts/153144
- Li, T. and J.J. Anderson. (2009). "The vitality model: A way to understand population survival and demographic heterogeneity." Theoretical Population Biology 76: 118-131.
- Li, T. and J.J. Anderson (2013). "Shaping human mortality patterns through intrinsic and extrinsiv vitality processes." Demographic Research 28(12): 341-372.
- Salinger, D.H., J.J. Anderson, and O.S. Hamel. (2003). "A parameter estimation routine for the vitality-based survival model." Ecological Modelling 166 (3): 287-29

Examples

```
# vitality.k
data(daphnia)
time <- daphnia$days
survival_fraction <- daphnia$lx

results.modk <- vitality.k(time = time,
```

```
                           sdata = survival_fraction,
                           rc.data=TRUE,
                           se=FALSE,
                           gfit=FALSE,
                           datatype="CUM",
                           ttol=.000001,
                           init.params=FALSE,
                           lower=c(0,-1,0), upper=c(100,50,50),
                           pplot=TRUE,
                           tlab="days",
                           lplot=TRUE,
                           cplot=TRUE,
                           Iplot=TRUE,
                           silent=TRUE)

# vitality.ku
data(rainbow_trout_for_k)
time <- rainbow_trout_for_k$days
survival_fraction <- rainbow_trout_for_k$survival

results.modku <- vitality.ku(time = time,
                             sdata = survival_fraction,
                             rc.data=TRUE,
                             se=FALSE,
                             gfit=FALSE,
                             datatype="CUM",
                             ttol=.000001,
                             init.params=FALSE,
                             lower=c(0,-1,0,0),upper=c(100,100,50,50),
                             pplot=TRUE,
                             tlab="days",
                             lplot=TRUE,
                             cplot=TRUE,
                             Iplot=TRUE,
                             silent=TRUE,
                             L=0)
# vitality.4p
data(swedish_females)
swe <- swedish_females
initial_age <- 20 # Could be adjusted
time <- initial_age:max(swedish_females$age)
survival_fraction <- swe$lx / swe$lx[1]
survival_fraction <- survival_fraction[time] # when first element <1 data is adjusted
sample_size <- swe$Lx[initial_age] #sample size

results.4par <- vitality.4p(time = time,
                            sdata = survival_fraction,
                            #init.params=FALSE,
                            init.params=c(0.012, 0.01, 0.1, 0.1),
                            lower = c(0, 0, 0, 0), upper = c(100,50,1,50),
                            rc.data = TRUE,
                            se = sample_size,
                            datatype = "CUM",
```

```
                            ttol = 1e-06,
                            pplot = TRUE,
                            Iplot = TRUE,
                            Mplot = TRUE,
                            tlab = "years",
                            silent = FALSE)

# vitality.6p
data(swedish_females)
swe <- swedish_females
initial_age <- 0
time <- swedish_females$age
survival_fraction <- swe$lx / swe$lx[1]
sample_size <- swe$Lx[1] #sample size

results.6par <- vitality.6p(time = time,
                            sdata = survival_fraction,
                            #init.params=FALSE,
                            init.params=c(0.012, 0.01, 0.1, 0.1, 0.1, 1),
                            lower = c(0, 0, 0, 0, 0, 0), upper = c(100,50,1,50,50,50),
                            rc.data = TRUE,
                            se=FALSE,
                            #se = sample_size,
                            datatype = "CUM",
                            ttol = 1e-06,
                            pplot = TRUE,
                            Iplot = TRUE,
                            Mplot = TRUE,
                            tlab = "years",
                            silent = FALSE)
```

`vitality.4p` *Fitting routine for the 2-process, 4-parameter vitality model (no childhood hook).*

Description

This function provides the fitting routine for the 4-parameter 2-process vitality model. Intrinsic mortality is characterized by the mean (r) and variability (s) in the vitality loss rate. Extrinsic mortality is characterized by random challenges of frequency (lambda) and random magnitude (beta) exceeding the remaining average vitality. Model is appropriate to adult human mortality data (e.g. Li and Anderson 2013).

Usage

```
vitality.4p(time = 0:(length(sdata)-1), sdata, init.params = FALSE,
  lower = c(0, 0, 0, 0), upper = c(100,50,100,50),rc.data = FALSE,
  se = FALSE, datatype = c("CUM", "INC"), ttol = 1e-06, pplot = TRUE,
  Iplot = FALSE, Mplot = FALSE, tlab = "years", silent = FALSE)
```

Arguments

`time`	Vector. Time component of data: Defaults to `0:(length(sdata)-1)`.
`sdata`	Required. Survival or mortality data. The default expects cumulative survival fraction. If providing incremental mortality fraction instead, use option: datatype = "INC". The default also expects the data to represent full mortality. Otherwise, use option: rc.data = T to indicate right censored data. If first element of `sdata` <1 data will be rescaled.
`rc.data`	Optional, Boolean. Specifies Right Censored data. If the data does not represent full mortality, it is probably right censored. The default is rc.data = F. A third option is rc.data = "TF". Use this case to add a near-term zero survival point to data which displays nearly full mortality (<.01 survival at end). If rc.data = F but the data does not show full mortality, rc.data = "TF" will be invoked automatically.
`se`	Optional, Boolean. Calculates the standard errors for the MLE parameters. Default is FALSE. Set equal to the initial study population to compute standard errors.
`datatype`	Optional. Defaults to `"CUM"` for cumulative survival fraction data. Use `"INC"` - for incremental mortality fraction data.
`ttol`	Optional. Stopping criteria tolerance. Default is 1e-6. Specify as ttol = .0001. If one of the likelihood plots (esp. for "k") does not look optimal, try decreasing ttol. If the program crashes, try increasing ttol.
`init.params`	Optional. Please specify the initial param values. specify `init.params = c(r, s, lambda, beta)` in that order (e.g.. init.params = c(.1, .02, .3, 0.12)).
`lower`	vector of lower parameter bounds in order of `c(r, s, lambda, beta)`. see `nlminb`
`upper`	vector of upper parameter bounds in order of `c(r, s, lambda, beta)`. see `nlminb`
`pplot`	Optional, Boolean. Plots of cumulative survival for both data and fitted curves? Default `TRUE`. `FALSE` Produce no plots. A A third option: `pplot = n` (n >= 1) extends the time axis of the fitting plots (beyond the max time in data). For example: `pplot = 1.2` extends the time axis by 20 Note: the incremental mortality plot is a continuous representation of the appropriately- binned histogram of incremental mortalities.
`Iplot`	Optional, Boolean. Incremental mortality for both data and fitted curves? Default: `FALSE`.
`Mplot`	Optional, Boolean. Plot fitted mortality curve? Default is `FALSE`. If `TRUE` 'observed' mortality rates for plotting are approximate and estimated from the input survival data. Depending on time (ages) and the change in mortality over age, these may be unrealistic but the fitted lines are derived from the parameter estimates themselves. For human data, the approximated mortality rates diverge at ages >80. `pplot` must be set to TRUE and `Iplot=FALSE`.
`tlab`	Optional, character. specifies units for x-axis of plots. Default is "days".
`silent`	Optional, Boolean. Stops all print and plot options (still get most warning and all error messages) Default is `FALSE`. A third option, `"verbose"` also enables the trace setting in the ms (minimum sum) S-Plus routine.

Value

vector of final MLE r, s, lambda, beta parameter estimates. standard errors of MLE parameter estimates (if se = <population> is specified).

References

D.H. Salinger, J.J. Anderson and O. Hamel (2003). "A parameter fitting routine for the vitality based survival model." Ecological Modeling 166(3): 287–294.

Li, T. and J.J. Anderson (2013). "Shaping human mortality patterns through intrinsic and extrinsic vitality processes." Demographic Research 28(12): 341-372.

Examples

```
data(swedish_females)
swe <- swedish_females
initial_age <- 20 # Could be adjusted
time <- initial_age:max(swedish_females$age)
survival_fraction <- swe$lx / swe$lx[1]
survival_fraction <- survival_fraction[time] # when first element <1 data is adjusted
sample_size <- swe$Lx[initial_age] #sample size

results.4par <- vitality.4p(time = time,
                            sdata = survival_fraction,
                            #init.params=FALSE,
                            init.params=c(0.012, 0.01, 0.1, 0.1),
                            lower = c(0, 0, 0, 0), upper = c(100,50,1,50),
                            rc.data = TRUE,
                            se = sample_size,
                            datatype = "CUM",
                            ttol = 1e-06,
                            pplot = TRUE,
                            Iplot = TRUE,
                            Mplot = TRUE,
                            tlab = "years",
                            silent = FALSE)
```

`vitality.6p` *Fitting routine for the 2-process, 6-parameter vitality model (with childhood hook).*

Description

This function provides the fitting routine for the 6-parameter 2-process vitality model. Intrinsic mortality is characterized by the mean (r) and variability (s) in the vitality loss rate. Adult extrinsic mortality is characterized by random challenges of frequency (lambda) and random magnitude (beta) exceeding the remaining average vitality. Child extrinsic mortality is characterized by childhood challenges of frequency (gamma) exceeding childhood vitality development rate (alpha). Model is appropriate to full lifespan of human mortality data (e.g. Anderson and Li 2015).

Usage

```
vitality.6p(time = 0:(length(sdata)-1), sdata, init.params = FALSE,
  lower = c(0, 0, 0, 0, 0, 0),upper = c(100,50,100,50,50,10), rc.data = FALSE,
  se = FALSE, datatype = c("CUM", "INC"), ttol = 1e-06, pplot = TRUE,
  Iplot = FALSE, Mplot = FALSE, tlab = "years", silent = FALSE)
```

Arguments

`time`	Vector. Time component of data: Defaults to `0:(length(sdata)-1)`. Typically this refers to ages.
`sdata`	Required. Survival or mortality data. The default expects cumulative survival fraction. If providing incremental mortality fraction instead, use option: datatype = "INC". The default also expects the data to represent full mortality. Otherwise, use option: rc.data = T to indicate right censored data.
`rc.data`	Optional, Boolean. Specifies Right Censored data. If the data does not represent full mortality, it is probably right censored. The default is rc.data = F. A third option is rc.data = "TF". Use this case to add a near-term zero survival point to data which displays nearly full mortality (<.01 survival at end). If rc.data = F but the data does not show full mortality, rc.data = "TF" will be invoked automatically.
`se`	Optional, Boolean. Calculates the standard errors for the MLE parameters. Default is FALSE. Set equal to the initial study population to compute standard errors.
`datatype`	Optional. Defaults to `"CUM"` for cumulative survival fraction data. Use `"INC"` - for incremental mortality fraction data.
`ttol`	Optional. Stopping criteria tolerance. Default is 1e-6. Specify as ttol = .0001. If one of the likelihood plots (esp. for "k") does not look optimal, try decreasing ttol. If the program crashes, try increasing ttol.
`init.params`	Optional. Please specify the initial param values. specify `init.params` = `c(r, s, lambda, beta, gamma, alpha)` in that order (e.g.. init.params = c(.1, .02, .3, 0.12, 0.1, 1)).
`lower`	vector of lower parameter bounds in order of `c(r, s, lambda, beta, gamma, alpha)`. see `nlminb`
`upper`	vector of upper parameter bounds in order of `c(r, s, lambda, beta, gamma, alpha)`. see `nlminb`
`pplot`	Optional, Boolean. Plots of cumulative survival for both data and fitted curves? Default: `TRUE`. `FALSE` produces no plots. A third option: `pplot` = `n` (n >= 1) extends the time axis of the fitting plots (beyond the max time in data). For example: `pplot` = `1.2` extends the time axis by 20 Note: the incremental mortality plot is a continuous representation of the appropriately binned histogram of incremental mortalities.
`Iplot`	Optional, Boolean. Incremental mortality for both data and fitted curves? Default: `FALSE`.
`Mplot`	Optional, Boolean. Plot fitted total, intrinsic, extrinsic (child), and extrinsic (adult) mortality curves? Default: `FALSE`. If `TRUE` 'observed' mortality rates for

	plotting are approximate and estimated from the input survival data. Depending on time (ages) and the change in mortality over age, the approximated, plotted mortality rates may be inaccurate (e.g. For human mortality data, mortality rates at the oldest ages are underestimated because nax is assumed to be half the length of the age interval but the true nax is likely larger). This caveat applies only to the plotted mortality rates (black circles in plot). The plotted, fitted lines are calculated with the parameter estimates themselves and are unaffected by the estimation of the mortality rates for this plot. `pplot` must be set to TRUE and `Iplot=FALSE`.
`tlab`	Optional, character. specifies units for x-axis of plots. Default is "years".
`silent`	Optional, Boolean. Stops all print and plot options (still get most warning and all error messages) Default is `FALSE`. A third option, `"verbose"` also enables the trace setting in the ms (minimum sum) S-Plus routine.

Value

vector of final MLE r, s, lambda, beta, gamma and alpha parameter estimates. standard errors of MLE parameter estimates (if se = <population> is specified).

References

D.H. Salinger, J.J. Anderson and O. Hamel (2003). "A parameter fitting routine for the vitality based survival model." Ecological Modeling 166(3): 287–294.

Anderson, J.J. and T. Li. (2015). "A two-process mortality model with extensions to juvenile mortality, populations and evolution." Population Association of America Annual Meeting 2015 http://paa2015.princeton.edu/abstracts/153144

Examples

```
data(swedish_females)
swe <- swedish_females
initial_age <- 0
time <- swedish_females$age
survival_fraction <- swe$lx / swe$lx[1]
sample_size <- swe$Lx[1] #sample size

results.6par <- vitality.6p(time = time,
                            sdata = survival_fraction,
                            #init.params=FALSE,
                            init.params=c(0.012, 0.01, 0.1, 0.1, 0.1, 1),
                            lower = c(0, 0, 0, 0, 0, 0), upper = c(100,50,1,50,50,50),
                            rc.data = TRUE,
                            se=FALSE,
                            #se = sample_size,
                            datatype = "CUM",
                            ttol = 1e-06,
                            pplot = TRUE,
                            Iplot = TRUE,
                            Mplot = TRUE,
                            tlab = "years",
```

```
                        silent = FALSE)
```

`vitality.k` *Fitting routine for the 3-parameter vitality model.*

Description

This function provides the fitting routine for the 3-parameter vitality model. Intrinsic mortality is characterized by the mean (r) and variability (s) in vitality loss rate. Extrinsic mortality is characterized by the frequency (k) of lethal random challenges. Model is appropriate to animal mortality data (e.g. Anderson 2000).

Usage

```
vitality.k(time, sdata, rc.data=F, se=F, gfit=F, datatype="CUM", ttol=.000001,
    init.params=F, lower=c(0,-1,0), upper=c(100,50,50), pplot=T, tlab="days",
    lplot=F, cplot=F, Iplot=F, silent=F)
```

Arguments

`time`	Vector. Time component of data: Defaults to `0:(length(sdata)-1)`. Typically this refers to ages.
`sdata`	Required. Survival or mortality data. The default expects cumulative survival fraction. If providing incremental mortality fraction instead, use option: datatype = "INC". The default also expects the data to represent full mortality. Otherwise, use option: rc.data = T to indicate right censored data.
`rc.data`	Optional, Boolean. Specifies Right Censored data. If the data does not represent full mortality, it is probably right censored. The default is rc.data = F. A third option is rc.data = "TF". Use this case to add a near-term zero survival point to data which displays nearly full mortality (<.01 survival at end). If rc.data = F but the data does not show full mortality, rc.data = "TF" will be invoked automatically.
`se`	Optional, Boolean. Calculates the standard errors for the MLE parameters. Default is FALSE. Set equal to the initial study population to compute standard errors.
`datatype`	Optional. Defaults to `"CUM"` for cumulative survival fraction data. Use `"INC"` - for incremental mortality fraction data.
`ttol`	Optional. Stopping criteria tolerance. Default is 1e-6. Specify as ttol = .0001. If one of the likelihood plots (esp. for "k") does not look optimal, try decreasing ttol. If the program crashes, try increasing ttol.
`init.params`	Optional. Please specify the initial param values. specify `init.params` = `c(r, s, k)` in that order (e.g.. init.params = c(.1, .02, .3)).
`lower`	vector of lower parameter bounds in order of `c(r, s, k)`. see `nlminb`
`upper`	vector of upper parameter bounds in order of `c(r, s, k)`. see `nlminb`

`pplot`	Optional, Boolean. Plots of cumulative survival for both data and fitted curves? Default: `TRUE`. `FALSE` produces no plots.Note: the incremental mortality plot is a continuous representation of the appropriately binned histogram of incremental mortalities.
`Iplot`	Boolean. Plot incremental survival? Must have `pplot=TRUE`
`lplot`	Boolean. Plot likelihood functions? Provides likelihood function plotting. Defaults to `FALSE`. Note: these plots are not "likelihood profiles" in that while one parameter is varied, the others are held fixed, rather than re-optimized. (must also have `pplot=T` and `Iplot=F`.)
`cplot`	Boolean. Plot likelihood contour plot? Provides a likelihood contour plot for a range of r and s values (can be slow so default is `FALSE`). Must also have `lplot=T` and `pplot=T` to get contour plots.
`tlab`	Character, label for time axis. Defaults to "days".
`gfit`	Provides a Pearson C type test for goodness of fit. Default is `FALSE`. Must provide the initial study population to compute goodness of fit.
`silent`	Optional, Boolean. Stops all print and plot options (still get most warning and all error messages) Default is `FALSE`. A third option, `"verbose"` also enables the trace setting in the ms (minimum sum) S-Plus routine.

Value

vector of final MLE r, s, k parameter estimates. standard errors of MLE parameter estimates (if se = <population> is specified).

References

- Anderson, J.J. (2000). "A vitality-based model relating stressors and environmental properties to organism survival." Ecological Monographs 70(3):445-470.

Examples

```
data(daphnia)
time <- daphnia$days
survival_fraction <- daphnia$lx

results.modk <- vitality.k(time = time,
                           sdata = survival_fraction,
                           rc.data=TRUE,
                           se=FALSE,
                           gfit=FALSE,
                           datatype="CUM",
                           ttol=.000001,
                           init.params=FALSE,
                           #init.params=c(0.075, 0.15, 0.001),
                           lower=c(0,-1,0), upper=c(100,50,50),
                           pplot=TRUE,
                           tlab="days",
                           lplot=TRUE,
                           cplot=TRUE,
```

```
                        Iplot=TRUE,
                        silent=FALSE)
```

`vitality.ku` *Fitting routine for the 4-parameter vitality model.*

Description

This function provides the fitting routine for the 4-parameter vitality model. Intrinsic mortality is characterized by the mean (r) and variability (s) in the vitality loss rate and the standard deviation of initial vitality (u). Extrinsic mortality is characterized by the frequency (k) of lethal random challenges. Model is appropriate to animal mortality data (e.g. Li and Anderson 2009)

Usage

```
vitality.ku(time, sdata, rc.data=F, se=F, gfit=F, datatype="CUM", ttol=.000001,
  init.params=F, lower=c(0,-1,0,0), upper=c(100,100,50,50), pplot=T, tlab="days",
   lplot=F, cplot=F, Iplot=F, silent=F, L=0)
```

Arguments

`time`	Vector. Time component of data. Typically this refers to ages.
`sdata`	Required. Survival or mortality data. The default expects cumulative survival fraction. If providing incremental mortality fraction instead, use option: datatype = "INC". The default also expects the data to represent full mortality. Otherwise, use option: rc.data = T to indicate right censored data.
`rc.data`	Optional, Boolean. Specifies Right Censored data. If the data does not represent full mortality, it is probably right censored. The default is rc.data = F. A third option is rc.data = "TF". Use this case to add a near-term zero survival point to data which displays nearly full mortality (<.01 survival at end). If rc.data = F but the data does not show full mortality, rc.data = "TF" will be invoked automatically.
`se`	Optional, Boolean. Calculates the standard errors for the MLE parameters. Default is FALSE. Set equal to the initial study population to compute standard errors.
`datatype`	Optional. Defaults to `"CUM"` for cumulative survival fraction data. Use `"INC"` - for incremental mortality fraction data.
`ttol`	Optional. Stopping criteria tolerance. Default is 1e-6. Specify as ttol = .0001. If one of the likelihood plots (esp. for "k") does not look optimal, try decreasing ttol. If the program crashes, try increasing ttol.
`init.params`	Optional. Please specify the initial param values. specify `init.params` = `c(r, s, k, u)` in that order (e.g.. init.params = c(.1, .02, .3, .25)).
`lower`	vector of lower parameter bounds in order of `c(r, s, k, u)`. see `nlminb`
`upper`	vector of upper parameter bounds in order of `c(r, s, k, u)`. see `nlminb`

`pplot`	Optional, Boolean. Plots of cumulative survival for both data and fitted curves? Default: `TRUE`. `FALSE` produces no plots. Note: the incremental mortality plot is a continuous representation of the appropriately binned histogram of incremental mortalities.
`Iplot`	Boolean. Plot incremental survival?
`lplot`	Boolean. Plot likelihood functions? Provides likelihood function plotting. Defaults to `FALSE`. Note: these plots are not "likelihood profiles" in that while one parameter is varied, the others are held fixed, rather than re-optimized. (must also have pplot=T.)
`cplot`	Boolean. Plot likelihood contour plot? Provides a likelihood contour plot for a range of parameter values (can be slow so default is `FALSE`). Must also have `lplot=T` and `pplot=T` to get contour plots.
`tlab`	Character, label for time axis. Defaults to "days".
`gfit`	Provides a Pearson C type test for goodness of fit. Default is `FALSE`. Must provide the initial study population to compute goodness of fit.
`silent`	Optional, Boolean. Stops all print and plot options (still get most warning and all error messages) Default is `FALSE`.
`L`	Number of times of running simulated annealing. Default is 0, use Newton-Ralphson method only.

Value

vector of final MLE r, s, k, u parameter estimates. standard errors of MLE parameter estimates (if se = <population> is specified).

References

- Li, T. and J.J. Anderson. (2009). "The vitality model: A way to understand population survival and demographic heterogeneity." Theoretical Population Biology 76: 118-131.

Examples

```
data(rainbow_trout_for_k)
time <- rainbow_trout_for_k$days
survival_fraction <- rainbow_trout_for_k$survival

results.modku <- vitality.ku(time = time,
                              sdata = survival_fraction,
                              rc.data=TRUE,
                              se=FALSE,
                              gfit=FALSE,
                              datatype="CUM",
                              ttol=.000001,
                              init.params=FALSE,
                              lower=c(0,-1,0,0),upper=c(100,100,50,50),
                              pplot=TRUE,
                              tlab="days",
                              lplot=TRUE,
```

```
          cplot=TRUE,
          Iplot=TRUE,
          silent=FALSE,
          L=0)
```

Index