Package 'relsurv'

July 23, 2025

Type Package

Title Relative Survival

Version 2.3-2

Date 2025-01-28

Author Maja Pohar Perme [aut],
Damjan Manevski [aut, cre]

Maintainer Damjan Manevski <damjan.manevski@mf.uni-lj.si>

Description Contains functions for analysing relative survival data, including nonparametric estimators of net (marginal relative) survival, relative survival ratio, crude mortality, methods for fitting and checking additive and multiplicative regression models, transformation approach, methods for dealing with population mortality tables. Work has been described in Pohar Perme, Pavlic (2018) <doi:10.18637/jss.v087.i08>.

Depends R (>= 4.1.0), survival (>= 3.1)

Imports splines, ggplot2, pammtools, scales, Rcpp (>= 1.0.10)

License GPL (>= 2)

LinkingTo Rcpp, RcppArmadillo

RoxygenNote 7.3.2

LazyData true

NeedsCompilation yes

Repository CRAN

Date/Publication 2025-01-28 15:50:02 UTC

Contents

---

cmp.rel	*Compute crude probability of death*

Description

Estimates the crude probability of death due to disease and due to population reasons

Usage

```
cmp.rel(
  formula = formula(data),
  data = parent.frame(),
  ratetable = relsurv::slopop,
  na.action,
  tau,
  conf.int = 0.95,
  precision = 1,
  add.times,
```

```
    rmap
)
```

Arguments

formula	a formula object, with the response as a Surv object on the left of a ~ operator, and, if desired, terms separated by the + operator on the right. If no strata are used, ~1 should be specified.
	NOTE: The follow-up time must be in days.
data	a data.frame in which to interpret the variables named in the formula.
ratetable	a table of event rates, organized as a ratetable object, such as slopop.
na.action	a missing-data filter function, applied to the model.frame, after any subset argument has been used. Default is options()$na.action.
tau	the maximum follow-up time of interest, all times larger than tau shall be censored. Equals maximum observed time by default
conf.int	the level for a two-sided confidence interval on the survival curve(s). Default is 0.95.
precision	the level of precision used in the numerical integration of variance. Default is 1, which means that daily intervals are taken, the value may be decreased to get a higher precision or increased to achieve a faster calculation. The calculation intervals always include at least all times of event and censoring as border points.
add.times	specific times at which the value of estimator and its variance should be evaluated. Default is all the event and censoring times.
rmap	an optional list to be used if the variables are not organized and named in the same way as in the ratetable object. See details below.

Details

NOTE: The follow-up time must be specified in days. The ratetable being used may have different variable names and formats than the user's data set, this is dealt with by the rmap argument. For example, if age is in years in the data set but in days in the ratetable object, age=age*365.241 should be used. The calendar year can be in any date format (Date and POSIXt are allowed), the date formats in the ratetable and in the data may differ.

Note that numerical integration is required to calculate the variance estimator. The integration precision is set with argument precision, which defaults to daily intervals, a default that should give enough precision for any practical purpose.

The area under the curve is calculated on the interval [0,tau].

Function summary may be used to get the output at specific points in time.

Value

An object of class cmp.rel. Objects of this class have methods for the functions print and plot. The summary function can be used for printing output at required time points. An object of class cmp.rel is composed of several lists, each pertaining the cumulative hazard function for one risk and one strata. Each of the lists contains the following objects:

time	the time-points at which the curves are estimated
est	the estimate
var	the variance of the estimate
lower	the lower limit of the confidence interval
upper	the upper limit of the confidence interval
area	the area under the curve calculated on the interval [0,tau]
index	indicator of event and censoring times among all the times in the output. The times added via paramater add.times are also included
add.times	the times added via parameter add.times

References

Package: Pohar Perme, M., Pavlic, K. (2018) "Nonparametric Relative Survival Analysis with the R Package relsurv". Journal of Statistical Software. 87(8), 1-27, doi: "10.18637/jss.v087.i08"

See Also

rs.surv, summary.cmp.rel

Examples

```
data(slopop)
data(rdata)
#calculate the crude probability of death
#note that the variable year must be given in a date format and that
#age must be multiplied by 365.241 in order to be expressed in days.
fit <- cmp.rel(Surv(time,cens)~sex,rmap=list(age=age*365.241),
ratetable=slopop,data=rdata,tau=3652.41)
fit
plot(fit,col=c(1,1,2,2),xscale=365.241,xlab="Time (years)")
#if no strata are desired:
fit <- cmp.rel(Surv(time,cens)~1,rmap=list(age=age*365.241),
ratetable=slopop,data=rdata,tau=3652.41)
```

---

colrec   *Relative Survival Data*

Description

Survival of patients with colon and rectal cancer diagnosed in 1994-2000.

Usage

data(colrec)

Format

A data frame with 5971 observations on the following 7 variables:

sex sex (1=male, 2=female).

age age (in days).

diag date of diagnosis (in date format).

time survival time (in days).

stat censoring indicator (0=censoring, 1=death).

stage cancer stage. Values 1-3, code 99 stands for unknown.

site cancer site.

References

Provided by Slovene Cancer Registry. The age, time and diag variables are randomly perturbed to make the identification of patients impossible.

---

ebmt1wide   *EBMT Data Taken From Package mstate*

Description

The ebmt1 dataset taken from package mstate transformed in long format.

Usage

```
data(ebmt1wide)
```

---

epa   *Excess hazard function smoothing*

Description

An Epanechnikov kernel function based smoother for smoothing the baseline excess hazard calculated by the rsadd function with the EM method.

Usage

```
epa(fit, bwin, times, n.bwin = 16, left = FALSE)
```

Arguments

`fit`	Fit from the additive relative survival model using the EM method.
`bwin`	The relative width of the smoothing window (default is 1).
`times`	The times at which the smoother is to be evaluated. If missing, it is evaluated at all event times.
`n.bwin`	Number of times that the window width may change.
`left`	If `FALSE` (default) smoothing is performed symmetrically, if `TRUE` only leftside neighbours are considered.

Details

The function performs Epanechnikov kernel smoothing. The follow up time is divided (according to percentiles of event times) into several intervals (number of intervals defined by `n.bwin`) in which the width is calculated as a factor of the maximum span between event times. Boundary effects are also taken into account on both sides.

Value

A list with two components:

`lambda`	the smoothed excess baseline hazard function
`times`	the times at which the smoothed excess baseline hazard is evaluated.

References

Package. Pohar M., Stare J. (2006) "Relative survival analysis in R." Computer Methods and Programs in Biomedicine, **81**: 272–278

Relative survival: Pohar, M., Stare, J. (2007) "Making relative survival analysis relatively easy." Computers in biology and medicine, **37**: 1741–1749.

EM algorithm: Pohar Perme M., Henderson R., Stare, J. (2009) "An approach to estimation in relative survival regression." Biostatistics, **10**: 136–146.

See Also

[rsadd](#),

Examples

```
data(slopop)
data(rdata)
#fit an additive model with the EM method
fit <- rsadd(Surv(time,cens)~sex+age,rmap=list(age=age*365.241),
ratetable=slopop,data=rdata,int=5,method="EM")
sm <- epa(fit)
plot(sm$times,sm$lambda)
```

---

expprep2 *expprep2 function*

Description

Helper calculation function using C code. Saved also as exp_prep (unexported function).

Usage

```
expprep2(
  x,
  y,
  ratetable,
  status,
  times,
  fast = FALSE,
  ys,
  prec,
  cmp = F,
  netweiDM = FALSE
)
```

Arguments

x	matrix of demographic covariates - each individual has one line
y	follow-up time for each individual (same length as nrow(x))
ratetable	rate table used for calculation
status	status for each individual (same length as nrow(x)!), not needed if we only need Spi, status needed for rs.surv
times	times at which we wish to evaluate the quantities, not needed if we only need Spi, times needed for rs.surv
fast	for mpp method only
ys	entry times (if empty, individuals are followed from time 0)
prec	deprecated
cmp	should cmpfast.C be used
netweiDM	should new netwei script be used

Details

Helper function used in rs.surv and other relsurv functions.

Value

List containing the calculated hazards and probabilities using the population mortality tables.

See Also

rs.surv

---

invtime   *Inverse transforming of time in Relative Survival*

Description

This function can be used when predicting in Relative Survival using the transformed time regression model (using rstrans function). It inverses the time from Y to T in relative survival using the given ratetable. The times Y can be produced with the rstrans function, in which case, this is the reverse function. This function does the transformation for one person at a time.

Usage

```
invtime(
  y = 0.1,
  age = 23011,
  sex = "male",
  year = as.Date("1986-01-01"),
  scale = 1,
  ratetable = relsurv::slopop,
  lower,
  upper
)
```

Arguments

y	time in Y.
age	age of the individual. Must be in days.
sex	sex of the individual. Must be coded in the same way as in the ratetable.
year	date of diagnosis. Must be in a Date or POSIXt format.
scale	numeric value to scale the results. If ratetable is in units/day, scale = 365.241 causes the output to be reported in years.
ratetable	a table of event rates, such as survexp.us.
lower	the lower bound of interval where the result is expected. This argument is optional, but, if given, can shorten the time the function needs to calculate the result.
upper	the upper bound of interval where the result is expected. See lower.

Details

Works only with ratetables that are split by age, sex and year. Transforming can be computationally intensive, use lower and/or upper to guess the interval of the result and thus speed up the function.

Value

A list of values

T	the original time
Y	the transformed time
.	

References

Package: Pohar M., Stare J. (2006) "Relative survival analysis in R." Computer Methods and Programs in Biomedicine, **81**: 272-278.

Relative survival: Pohar, M., Stare, J. (2007) "Making relative survival analysis relatively easy." Computers in biology and medicine, **37**: 1741-1749.

See Also

[rstrans](#)

Examples

```
data(slopop)
invtime(y = 0.1, age = 23011, sex = 1, year = as.Date('1986-01-01'), ratetable = slopop)
```

---

joinrate   *Join ratetables*

Description

The function joins two or more objects organized as `ratetable` by adding a new dimension.

Usage

```
joinrate(tables, dim.name = "country", merge = FALSE)
```

Arguments

tables	a list of ratetables. If names are given, they are included as `dimnames`.
dim.name	the name of the added dimension.
merge	if FALSE (default) only the intersect of all years/ages is taken. If TRUE all possible years/ages are included (NOTE: in this case hazards are extrapolated from earlier or later years/ages). This option only works for ratetables with dimensions

Details

This function joins two or more `ratetable` objects by adding a new dimension. The cutpoints of all the rate tables are compared; if merge=FALSE (default) only the common intervals are kept, otherwise if merge=TRUE all intervals are added (and hazards are extrapolated). If the intervals defined by the cutpoints are not of the same length, a warning message is displayed. Each rate table must have the same dimensions.

Value

An object of class `ratetable`.

References

Package: Pohar M., Stare J. (2006) "Relative survival analysis in R." Computer Methods and Programs in Biomedicine, **81**: 272-278.

Relative survival: Pohar, M., Stare, J. (2007) "Making relative survival analysis relatively easy." Computers in biology and medicine, **37**: 1741-1749.

See Also

`survival::ratetable`, `transrate.hld`, `transrate.hmd`, `transrate`.

Examples

```
#newpop <- joinrate(list(Arizona=survexp.az,Florida=survexp.fl,
#                    Minnesota=survexp.mn),dim.name="state")
```

---

nessie   *Net Expected Sample Size Is Estimated*

Description

Calculates how the sample size decreases in time due to population mortality

Usage

```
nessie(
  formula = formula(data),
  data = parent.frame(),
  ratetable = relsurv::slopop,
  times,
  rmap
)
```

Arguments

formula	a formula object, same as in rs.surv. The right-hand side of the formula object includes the variable that defines the subgroups (a variable of type factor) by which the expected sample size is to be calculated.
data	a data.frame in which to interpret the variables named in the formula.
ratetable	a table of event rates, organized as a ratetable object, such as slopop.
times	Times at which the calculation should be evaluated - in years!
rmap	an optional list to be used if the variables are not organized and named in the same way as in the ratetable object. See details of the rs.surv function.

Details

The function calculates the sample size we can expect at a certain time point if the patients die only due to population causes (population survival * initial sample size in a certain category), i.e. the number of individuals that remains at risk at given timepoints after the individuals who die due to population causes are removed. The result should be used as a guideline for the sensible length of follow-up interval when calculating the net survival.

The first column of the output reports the number of individuals at time 0. The last column of the output reports the conditional expected (population) survival time for each subgroup.

Value

A list of values.

References

Pohar Perme, M., Pavlic, K. (2018) "Nonparametric Relative Survival Analysis with the R Package relsurv". Journal of Statistical Software. 87(8), 1-27, doi: "10.18637/jss.v087.i08"

See Also

rs.surv

Examples

```
data(slopop)
data(rdata)
rdata$agegr <-cut(rdata$age,seq(40,95,by=5))
nessie(Surv(time,cens)~agegr,rmap=list(age=age*365.241),
ratetable=slopop,data=rdata,times=c(1,3,5,10,15))
```

---

plot.cmp.rel   *Plot the crude probability of death*

Description

Plot method for cmp.rel. Plots the cumulative probability of death due to disease and due to population reasons

Usage

```
## S3 method for class 'cmp.rel'
plot(
  x,
  main = " ",
  curvlab,
  ylim = c(0, 1),
  xlim,
  wh = 2,
  xlab = "Time (days)",
  ylab = "Probability",
  lty = 1:length(x),
  xscale = 1,
  col = 1,
  lwd = par("lwd"),
  curves,
  conf.int,
  all.times = FALSE,
  ...
)
```

Arguments

x	a list, with each component representing one curve in the plot, output of the function `cmp.rel`.
main	the main title for the plot.
curvlab	Curve labels for the plot. Default is `names(x)`, or if that is missing, `1:nc`, where `nc` is the number of curves in `x`.
ylim	yaxis limits for plot.
xlim	xaxis limits for plot (default is 0 to the largest time in any of the curves).
wh	if a vector of length 2, then the upper right coordinates of the legend; otherwise the legend is placed in the upper right corner of the plot.
xlab	X axis label.
ylab	y axis label.

lty	vector of line types. Default 1:nc (nc is the number of curves in x). For color displays, lty=1, color=1:nc, might be more appropriate. If length(lty)<nc, then lty[1] is used for all.
xscale	Scale of the X axis. Default is in days (1).
col	vector of colors. If length(col)<nc, then the col[1] is used for all.
lwd	vector of line widths. If length(lwd)<nc, then lwd[1] is used for all.
curves	Vector if integers, specifies which curves should be plotted. May take values 1:nc, where nc is the number of curves in x. By default, all of the curves are plotted.
conf.int	Vector if integers, specifies which confidence intervals should be plotted. May take values 1:nc, where nc is the number of curves in x. By default, no confidence intervals are plotted.
all.times	By default, the disease specific mortality estimate is plotted as a step function between event or censoring times. If set to TRUE, the graph is evaluated at all estimated times.
...	additional arguments passed to the initial call of the plot function.

Details

By default, the graph is plotted as a step function for the cause specific mortality and as a piecewise linear function for the population mortality. It is evaluated at all event and censoring times even though it constantly changes also between these time points.

If the argument all.times is set to TRUE, the plot is evaluated at all times that were used for numerical integration in the cmp.rel function (there, the default is set to daily intervals). If only specific time points are to be added, this should be done via argument add.times in cmp.rel.

Value

No value is returned.

See Also

rs.surv

Examples

```
data(slopop)
data(rdata)
fit <- cmp.rel(Surv(time,cens)~sex,rmap=list(age=age*365.241),
ratetable=slopop,data=rdata,tau=3652.41)
plot(fit,col=c(1,1,2,2),xscale=365.241,conf.int=c(1,3))
```

plot.rs.zph	*Graphical Inspection of Proportional Hazards Assumption in Relative Survival Models*

Description

Displays a graph of the scaled partial residuals, along with a smooth curve.

Usage

```
## S3 method for class 'rs.zph'
plot(
  x,
  resid = TRUE,
  df = 4,
  nsmo = 40,
  var,
  cex = 1,
  add = FALSE,
  col = 1,
  lty = 1,
  xlab,
  ylab,
  xscale = 1,
  ...
)
```

Arguments

x	result of the `rs.zph` function.
resid	a logical value, if `TRUE` the residuals are included on the plot, as well as the smooth fit.
df	the degrees of freedom for the fitted natural spline, `df=2` leads to a linear fit.
nsmo	number of points used to plot the fitted spline.
var	the set of variables for which plots are desired. By default, plots are produced in turn for each variable of a model. Selection of a single variable allows other features to be added to the plot, e.g., a horizontal line at zero or a main title.
cex	a numerical value giving the amount by which plotting text and symbols should be scaled relative to the default.
add	logical, if `TRUE` the plot is added to an existing plot
col	a specification for the default plotting color.
lty	the line type.
xlab	x axis label.
ylab	y axis label.
xscale	units for x axis, default is 1, i.e. days.
...	Additional arguments passed to the `plot` function.

References

Goodness of fit: Stare J.,Pohar Perme M., Henderson R. (2005) "Goodness of fit of relative survival models." Statistics in Medicine, **24**: 3911-3925.

Package: Pohar M., Stare J. (2006) "Relative survival analysis in R." Computer Methods and Programs in Biomedicine, **81**: 272-278.

Relative survival: Pohar, M., Stare, J. (2007) "Making relative survival analysis relatively easy." Computers in biology and medicine, **37**: 1741-1749, 2007.

See Also

`rs.zph`, `survival::plot.cox.zph`

Examples

```
data(slopop)
data(rdata)
fit <- rsadd(Surv(time,cens)~sex+as.factor(agegr),rmap=list(age=age*365.241),
             ratetable=slopop,data=rdata,int=5)
rszph <- rs.zph(fit)
plot(rszph)
```

---

plot_f   *Plot the absolute risk (observed and population curve)*

Description

Plots the estimated observed and population curve for the life years difference (Manevski, Ruzic Gorenjec, Andersen, Pohar Perme, 2022).

Usage

```
plot_f(
  years,
  xlab = "Time interval",
  ylab = "Absolute risk",
  xbreak,
  ybreak,
  xlimits,
  ylimits,
  show.legend = TRUE
)
```

Arguments

`years`	the object obtained using function `years`.
`xlab`	a title for the x axis.
`ylab`	a title for the y axis.
`xbreak`	the breaks on the x axis (this is supplied to `scale_x_continuous`).
`ybreak`	the breaks on the y axis (this is supplied to `scale_y_continuous`).
`xlimits`	define the limits on the x axis (this is supplied to `scale_x_continuous`).
`ylimits`	define the limits on the y axis (this is supplied to `scale_y_continuous`).
`show.legend`	if TRUE, the legend is shown on the graph.

Details

A ggplot2 implementation for plotting the observed and population curves. The type of curves is dependent upon the measure calculated using `years` function (argument `measure`).

Value

A ggplot object

See Also

[years](#), [plot_years](#)

`plot_years`	*Plot the years measure*

Description

Plot the years measure obtained from the `years` function.

Usage

```
plot_years(
  years,
  xlab = "Time interval",
  ylab = "Years",
  xbreak,
  ybreak,
  xlimits,
  ylimits,
  conf.int = FALSE,
  ymirror = FALSE,
  yminus = FALSE
)
```

Arguments

years	the object obtained using function `years`.
xlab	a title for the x axis.
ylab	a title for the y axis.
xbreak	the breaks on the x axis (this is supplied to `scale_x_continuous`).
ybreak	the breaks on the y axis (this is supplied to `scale_y_continuous`).
xlimits	define the limits on the x axis (this is supplied to `scale_x_continuous`).
ylimits	define the limits on the y axis (this is supplied to `scale_y_continuous`).
conf.int	if TRUE, the confidence interval is plotted.
ymirror	mirror the y values (w.r.t. the x axis).
yminus	use function y -> -y when plotting.

Details

A ggplot2 implementation for plotting the years measure. The type of curve is dependent upon the measure calculated using the `years` function (argument `measure`).

Value

A ggplot object

See Also

[years](), [plot_f]()

---

popsurv *Calculate the expected (population) survival*

Description

For a given individual with sex, year, and age, calculate the expected (population) survival at the supplied time points based on the mortality tables.

Usage

```
popsurv(sex, year = as.Date("1970-01-01"), age = 0, ratetable, times)
```

Arguments

sex	Either character ('male'/'female'), or integer (1/2).
year	The year from which the individual is followed. Either a Date or POSIXt object. Default is as.Date('1970-01-01').
age	The age from which the individual is followed. Must be in days.
ratetable	a table of event rates, organized as a `ratetable` object, such as `slopop`.
times	The times at which the expected (population) survival should be calculated. Must be in days.

Details

The follow-up time and age must be specified in days. The calendar year can be in any date format (Date and POSIXt are allowed)

Value

A vector containing the survival estimate at the supplied times.

See Also

[expprep2](#)

Examples

```
library(relsurv)
# Estimate P(T>2000 days) for a newborn:
popsurv(sex='male', year=as.Date('1970-01-01'), age=0, ratetable=slopop, times=2000)
# P(T>300 days) for a 50-year old:
popsurv(sex='male', year=as.Date('1970-01-01'), age=50*365.241, ratetable=slopop, times=300)
```

---

predict.rsadd   *Subject-specific prediction from rsadd*

Description

Function

Usage

```
## S3 method for class 'rsadd'
predict(object, newdata, ...)
```

Arguments

object	An rsadd object
newdata	A data.frame with one row (add covariate values in columns)
...	Not used for now

Value

A data.frame with times, excess and population hazard.

Author(s)

Damjan Manevski <damjan.manevski@mf.uni-lj.si>

---

rdata	*Survival Data*

Description

Survival data.

Usage

```
data(rdata)
```

Format

A data frame with 1040 observations on the following 6 variables:

time survival time (in days).

cens censoring indicator (0=censoring, 1=death).

age age (in years).

sex sex (1=male, 2=female).

year date of diagnosis (in date format).

agegr age group.

References

Pohar M., Stare J. (2006) "Relative survival analysis in R." Computer Methods and Programs in Biomedicine, **81**: 272-278.

---

residuals.rsadd	*Calculate Residuals for a "rsadd" Fit*

Description

Calculates partial residuals for an additive relative survival model.

Usage

```
## S3 method for class 'rsadd'
residuals(object, type = "schoenfeld", ...)
```

Arguments

object	an object inheriting from class rsadd, representing a fitted additive relative survival model. Typically this is the output from the rsadd function.
type	character string indicating the type of residual desired. Currently only Schoenfeld residuals are implemented.
...	other arguments.

Value

A list of the following values is returned:

res   a matrix containing the residuals for each variable.

varr   the variance for each residual

varr1   the sum of `varr`.

kvarr   the derivative of each residual, to be used in `rs.zph` function.

kvarr1   the sum of `kvarr`.

References

Package. Pohar M., Stare J. (2006) "Relative survival analysis in R." Computer Methods and Programs in Biomedicine, **81**: 272–278

Relative survival: Pohar, M., Stare, J. (2007) "Making relative survival analysis relatively easy." Computers in biology and medicine, **37**: 1741–1749.

Goodness of fit: Stare J.,Pohar Perme M., Henderson R. (2005) "Goodness of fit of relative survival models." Statistics in Medicine, **24**: 3911–3925.

See Also

[rsadd](#).

Examples

```
data(slopop)
data(rdata)
fit <- rsadd(Surv(time,cens)~sex,rmap=list(age=age*365.241),
        ratetable=slopop,data=rdata,int=5)
sresid <- residuals.rsadd(fit)
```

---

rs.br   *Test the Proportional Hazards Assumption for Relative Survival Re-*
                                gression Models

Description

Test the proportional hazards assumption for relative survival models (rsadd, rsmul or rstrans) by forming a Brownian Bridge.

Usage

```
rs.br(fit, sc, rho = 0, test = "max", global = TRUE)
```

Arguments

fit	the result of fitting a relative survival model, using the rsadd, rsmul or rstrans function.
sc	partial residuals calculated by the resid function. This is used to save time if several tests are to be calculated on these residuals and can otherwise be omitted.
rho	a number controlling the weigths of residuals. The weights are the number of individuals at risk at each event time to the power rho. The default is rho=0, which sets all weigths to 1.
test	a character string specifying the test to be performed on Brownian bridge. Possible values are "max" (default), which tests the maximum absolute value of the bridge, and cvm, which calculates the Cramer Von Mises statistic.
global	should a global Brownian bridge test be performed, in addition to the per-variable tests

Value

an object of class rs.br. This function would usually be followed by both a print and a plot of the result. The plot gives a Brownian bridge for each of the variables. The horizontal lines are the 95 confidence intervals for the maximum absolute value of the Brownian bridge

References

Goodness of fit: Stare J.,Pohar Perme M., Henderson R. (2005) "Goodness of fit of relative survival models." Statistics in Medicine, **24**: 3911–3925.

Package. Pohar M., Stare J. (2006) "Relative survival analysis in R." Computer Methods and Programs in Biomedicine, **81**: 272–278

Relative survival: Pohar, M., Stare, J. (2007) "Making relative survival analysis relatively easy." Computers in biology and medicine, **37**: 1741–1749.

See Also

rsadd, rsmul, rstrans, resid.

Examples

```
data(slopop)
data(rdata)
fit <- rsadd(Surv(time,cens)~sex,rmap=list(age=age*365.241),
ratetable=slopop,data=rdata,int=5)
rsbr <- rs.br(fit)
rsbr
plot(rsbr)
```

| rs.diff | *Test Net Survival Curve Differences* |

Description

Tests if there is a difference between two or more net survival curves using a log-rank type test.

Usage

```
rs.diff(
  formula = formula(data),
  data = parent.frame(),
  ratetable = relsurv::slopop,
  na.action,
  precision = 1,
  rmap
)
```

Arguments

formula	A formula expression as for other survival models, of the form `Surv(time, status) ~ predictors`. Each combination of predictor values defines a subgroup. A `strata` term may be used to produce a stratified test. NOTE: The follow-up time must be in days.
data	a data.frame in which to interpret the variables named in the `formula`.
ratetable	a table of event rates, organized as a `ratetable` object, such as `slopop`.
na.action	a missing-data filter function, applied to the model.frame, after any subset argument has been used. Default is `options()$na.action`.
precision	Precision for numerical integration. Default is 1, which means that daily intervals are taken, the value may be decreased to get a higher precision or increased to achieve a faster calculation. The calculation intervals always include at least all times of event and censoring as border points.
rmap	an optional list to be used if the variables are not organized and named in the same way as in the `ratetable` object. See details below.

Details

NOTE: The follow-up time must be specified in days. The `ratetable` being used may have different variable names and formats than the user's data set, this is dealt with by the `rmap` argument. For example, if age is in years in the data set but in days in the `ratetable` object, age=age*365.241 should be used. The calendar year can be in any date format (Date and POSIXt are allowed), the date formats in the `ratetable` and in the data may differ.

Value

a `rsdiff` object; can be printed with `print`.

References

Package: Pohar Perme, M., Pavlic, K. (2018) "Nonparametric Relative Survival Analysis with the R Package relsurv". Journal of Statistical Software. 87(8), 1-27, doi: "10.18637/jss.v087.i08" Theory: Graffeo, N., Castell, F., Belot, A. and Giorgi, R. (2016) "A log-rank-type test to compare net survival distributions. Biometrics. doi: 10.1111/biom.12477" Theory: Pavlic, K., Pohar Perme, M. (2017) "On comparison of net survival curves. BMC Med Res Meth. doi: 10.1186/s12874-017-0351-3"

See Also

`rs.surv`, `survdiff`

Examples

```
data(slopop)
data(rdata)
#calculate the relative survival curve
#note that the variable year is given in days since 01.01.1960 and that
#age must be multiplied by 365.241 in order to be expressed in days.
rs.diff(Surv(time,cens)~sex,rmap=list(age=age*365.241),
ratetable=slopop,data=rdata)
```

---

rs.surv   *Compute a Relative Survival Curve*

Description

Computes an estimate of the relative survival curve using the Ederer I, Ederer II method, Pohar-Perme method or the Hakulinen method

Usage

```
rs.surv(
  formula = formula(data),
  data = parent.frame(),
  ratetable = relsurv::slopop,
  na.action,
  fin.date,
  method = "pohar-perme",
  conf.type = "log",
  conf.int = 0.95,
  type = "kaplan-meier",
  add.times,
  precision = 1,
  rmap,
  weight.table = NULL,
  weight.names = NULL
)
```

Arguments

formula	a formula object, with the response as a Surv object on the left of a ~ operator, and, if desired, terms separated by the + operator on the right. If no strata are used, ~1 should be specified.
	NOTE: The follow-up time must be in days.
data	a data.frame in which to interpret the variables named in the formula.
ratetable	a table of event rates, organized as a ratetable object, such as slopop.
na.action	a missing-data filter function, applied to the model.frame, after any subset argument has been used. Default is options()$na.action.
fin.date	the date of the study ending, used for calculating the potential follow-up times in the Hakulinen method. If missing, it is calculated as max(year+time).
method	the method for calculating the relative survival. The options are pohar-perme(default), ederer1, ederer2 and hakulinen.
conf.type	one of plain, log (the default), or log-log. The first option causes the standard intervals curve +- k *se(curve), where k is determined from conf.int. The log option calculates intervals based on the cumulative hazard or log(survival). The last option bases intervals on the log hazard or log(-log(survival)).
conf.int	the level for a two-sided confidence interval on the survival curve(s). Default is 0.95.
type	defines how survival estimates are to be calculated given the hazards. The default (kaplan-meier) calculates the product integral, whereas the option fleming-harrington exponentiates the negative cumulative hazard. Analogous to the usage in survfit.
add.times	specific times at which the curve should be evaluated.
precision	Precision for numerical integration. Default is 1, which means that daily intervals are taken, the value may be decreased to get a higher precision or increased to achieve a faster calculation. The calculation intervals always include at least all times of event and censoring as border points.
rmap	an optional list to be used if the variables are not organized and named in the same way as in the ratetable object. See details below.
weight.table	Default is NULL. If defined it is used for calculating standardized net survival. Supply a data.frame that contains the weights and group names for calculating the standardized net survival. The group column also has to be included in the data argument. For the theory see the details below.
weight.names	Default is NULL. If defined it is used for calculating standardized net survival. Supply a character vector of length two with the names of the group and weight columns in weight.table.

Details

NOTE: The follow-up time must be specified in days. The ratetable being used may have different variable names and formats than the user's data set, this is dealt with by the rmap argument. For example, if age is in years in the data set but in days in the ratetable object, age=age*365.241 should be used. The calendar year can be in any date format (Date and POSIXt are allowed), the date formats in the ratetable and in the data may differ.

The potential censoring times needed for the calculation of the expected survival by the Hakulinen method are calculated automatically. The times of censoring are left as they are, the times of events are replaced with `fin.date - year`.

The calculation of the Pohar-Perme estimate is more time consuming since more data are needed from the population tables. The old version of the function, now named `rs.survo` can be used as a faster version for the Hakulinen and Ederer II estimate.

Numerical integration is required for Pohar-Perme estimate. The integration precision is set with argument `precision`, which defaults to daily intervals, a default that should give enough precision for any practical purpose.

Note that even though the estimate is always calculated using numerical integration, only the values at event and censoring times are reported. Hence, the function `plot` draws a step function in between and the function `summary` reports the value at the last event or censoring time before the specified time. If the output of the estimated values at other points is required, this should be specified with argument `add.times`.

Standardized net survival can be also calculated:

$SNS(t) = \sum_j w_j NS_j(t)$

where NS_j is the net survival in the j-th group, w_j is the weight for the j-th group and SNS is the standardized net survival. SNS can be calculated by using the `weight.table` and `weight.names` arguments. The function also returns the corresponding standard error and confidence interval.

Value

a `survfit` object; see the help on `survfit.object` for details. The `survfit` methods are used for `print`, `summary`, `plot`, `lines`, and `points`.

References

Package: Pohar Perme, M., Pavlic, K. (2018) "Nonparametric Relative Survival Analysis with the R Package relsurv". Journal of Statistical Software. 87(8), 1-27, doi: "10.18637/jss.v087.i08" Theory: Pohar Perme, M., Esteve, J., Rachet, B. (2016) "Analysing Population-Based Cancer Survival - Settling the Controversies." BMC Cancer, 16 (933), 1-8. doi:10.1186/s12885-016-2967-9. Theory: Pohar Perme, M., Stare, J., Esteve, J. (2012) "On Estimation in Relative Survival", Biometrics, 68(1), 113-120. doi:10.1111/j.1541-0420.2011.01640.x.

See Also

`survfit`, `survexp`

Examples

```
data(slopop)
data(rdata)
#calculate the relative survival curve
#note that the variable year must be given in a date format and that
#age must be multiplied by 365.241 in order to be expressed in days.
rs.surv(Surv(time,cens)~sex,rmap=list(age=age*365.241), ratetable=slopop,data=rdata)

# SNS:
```

```
wei <- data.frame(agegr=c('<54', '54-61', '62-70', '71-95'),
                  weight=c(0.2, 0.2, 0.3, 0.3))
rs.surv(Surv(time,cens)~1, rmap=list(age=age*365.241), ratetable=slopop, data=rdata,
    weight.table=wei, weight.names = c('agegr', 'weight'))
```

---

rs.surv.rsadd   *Compute a Relative Survival Curve from an additive relative survival*
                                  model

Description

Computes the predicted relative survival function for an additive relative survival model fitted with maximum likelihood.

Usage

```
rs.surv.rsadd(formula, newdata)
```

Arguments

formula	a rsadd object (Implemented only for models fitted with the codemax.lik (default) option.)
newdata	a data frame with the same variable names as those that appear in the rsadd formula. a predicted curve for each individual in this data frame shall be calculated

Details

Does not work with factor variables - you have to form dummy variables before calling the rsadd function.

Value

a survfit object; see the help on survfit.object for details. The survfit methods are used for print, plot, lines, and points.

References

Package. Pohar M., Stare J. (2006) "Relative survival analysis in R." Computer Methods and Programs in Biomedicine, **81**: 272–278

See Also

survfit, survexp

Examples

```
data(slopop)
data(rdata)
#fit a relative survival model
fit <- rsadd(Surv(time,cens)~sex+age+year,rmap=list(age=age*365.241),
ratetable=slopop,data=rdata,int=c(0:10,15))

#calculate the predicted curve for a male individual, aged 65, diagnosed in 1982
d <- rs.surv.rsadd(fit,newdata=data.frame(sex=1,age=65,year=as.Date("1982-01-01")))
#plot the curve (will result in a step function since the baseline is assumed piecewise constant)
plot(d,xscale=365.241)

#calculate the predicted survival curves for each individual in the data set
d <- rs.surv.rsadd(fit,newdata=rdata)
#calculate the average over all predicted survival curves
p.surv <- apply(d$surv,1,mean)
#plot the relative survival curve
plot(d$time/365.241,p.surv,type="b",ylim=c(0,1),xlab="Time",ylab="Relative survival")
```

---

rs.zph	*Behaviour of Covariates in Time for Relative Survival Regression Models*

Description

Calculates the scaled partial residuals of a relative survival model (rsadd, rsmul or rstrans)

Usage

```
rs.zph(fit, sc, transform = "identity", var.type = "sum")
```

Arguments

fit	the result of fitting an additive relative survival model, using the rsadd, rsmul or rstrans function.
	In the case of multiplicative and transformation models the output is identical to cox.zph function, except no test is performed.
sc	partial residuals calculated by the resid function. This is used to save time if several tests are to be calculated on these residuals and can otherwise be omitted.
transform	a character string specifying how the survival times should be transformed. Possible values are "km", "rank", "identity" and log. The default is "identity".
var.type	a character string specifying the variance used to scale the residuals. Possible values are "each", which estimates the variance for each residual separately, and sum(default), which assumes the same variance for all the residuals.

Value

an object of class `rs.zph`. This function would usually be followed by a plot of the result. The plot gives an estimate of the time-dependent coefficient beta(t). If the proportional hazards assumption is true, beta(t) will be a horizontal line.

References

Goodness of fit: Stare J.,Pohar Perme M., Henderson R. (2005) "Goodness of fit of relative survival models." Statistics in Medicine, **24**: 3911–3925.

Package. Pohar M., Stare J. (2006) "Relative survival analysis in R." Computer Methods and Programs in Biomedicine, **81**: 272–278

Relative survival: Pohar, M., Stare, J. (2007) "Making relative survival analysis relatively easy." Computers in biology and medicine, **37**: 1741–1749.

See Also

rsadd, rsmul, rstrans, resid, survival::cox.zph.

Examples

```
data(slopop)
data(rdata)
fit <- rsadd(Surv(time,cens)~sex,rmap=list(age=age*365.241),
ratetable=slopop,data=rdata,int=5)
rszph <- rs.zph(fit)
plot(rszph)
```

---

rsaalen *Fit an extended additive hazards model using relative survival*

Description

Fits the Aalen additive hazard model using relative survival. The function can be used for multi-state model data (as in the package mstate; class msdata) by supplying the start and stop times in the Surv object and adding a strata(trans) object in the formula (where trans denotes the transition in the multi-state model).

Usage

```
rsaalen(
  formula,
  data,
  variance = FALSE,
  var_estimator = "dN",
  ratetable = relsurv::slopop,
  rmap,
  split.transitions
)
```

Arguments

`formula`	a formula object, with the response as a `Surv` object on the left of a ~ operator, and, if desired, terms separated by the + operator on the right.
`data`	a data.frame in which to interpret the variables named in the `formula`.
`variance`	a logical value indicating whether the variances of the hazards should be computed. Default is FALSE.
`var_estimator`	Choose variance estimator, in the same way as in survaalen. The default option 'dN' uses dN(t)-dLambda_P(t) in the variance estimator, equivalent to formula 4.63 in Aalen et al. (2008). Option 'XdB' uses X*dB(t), see formula 4.64 in Aalen et al. (2008).
`ratetable`	a table of event rates, organized as a `ratetable` object, such as `slopop`.
`rmap`	an optional list to be used if the variables are not organized and named in the same way as in the `ratetable` object.
`split.transitions`	only relevant if a multi-state model is fitted. An integer vector containing the numbered transitions that should be split. Use same numbering as in the given transition matrix.

Value

An object of class `aalen.model`.

Author(s)

Damjan Manevski

See Also

`survaalen`

Examples

```
# Survival:
data(rdata)
mod <- rsaalen(Surv(time, cens)~sex+age, data=rdata, ratetable=slopop,
     rmap=list(age=age*365.241))
head(mod$coefficients)
tail(mod$coefficients)

# Multi-state model:
data(ebmt1wide)
# Generate sex and year data (for illustrative purposes since it is not given in the data):
ebmt1wide$sex <- sample(1:2, size = nrow(ebmt1wide), replace = TRUE)
ebmt1wide$year <- as.Date('2010-01-01')

mod <- rsaalen(Surv(Tstart, Tstop, status)~age.1+age.2+age.3+strata(trans), data=ebmt1wide,
               ratetable = slopop, rmap = list(age=age*365.241), split.transitions = 2:3)
head(mod$coefficients$trans1)
```

```
head(mod$coefficients$trans2)
head(mod$coefficients$trans3)
```

---

rsadd   *Fit an Additive model for Relative Survival*

Description

The function fits an additive model to the data. The methods implemented are the maximum likelihood method, the semiparametric method, a glm model with a `binomial` error and a glm model with a `poisson` error.

Usage

```
rsadd(
  formula = formula(data),
  data = parent.frame(),
  ratetable = relsurv::slopop,
  int,
  na.action,
  method = "max.lik",
  init,
  bwin,
  centered = FALSE,
  cause,
  control,
  rmap,
  ...
)
```

Arguments

formula	a formula object, with the response as a `Surv` object on the left of a ~ operator, and, if desired, terms separated by the + operator on the right. `Surv(start,stop,event)` outcomes are also possible for time-dependent covariates and left-truncation for `method='EM'`.
	NOTE: The follow-up time must be in days.
data	a data.frame in which to interpret the variables named in the `formula`.
ratetable	a table of event rates, organized as a `ratetable` object, such as `slopop`.
int	either a single value denoting the number of follow-up years or a vector specifying the intervals (in years) in which the hazard is constant (the times that are bigger than `max(int)` are censored. If missing, only one interval (from time 0 to maximum observation time) is assumed. The EM method does not need the intervals, only the maximum time can be specified (all times are censored after this time point).

na.action	a missing-data filter function, applied to the model.frame, after any subset argument has been used. Default is options()$na.action.
method	glm.bin or glm.poi for a glm model, EM for the EM algorithm and max.lik for the maximum likelihood model (default).
init	vector of initial values of the iteration. Default initial value is zero for all variables.
bwin	controls the bandwidth used for smoothing in the EM algorithm. The follow-up time is divided into quartiles and bwin specifies a factor by which the maximum between events time length on each interval is multiplied. The default bwin=-1 lets the function find an appropriate value. If bwin=0, no smoothing is applied.
centered	if TRUE, all the variables are centered before fitting and the baseline excess hazard is calculated accordingly. Default is FALSE.
cause	A vector of the same length as the number of cases. 0 for population deaths, 1 for disease specific deaths, 2 (default) for unknown. Can only be used with the EM method.
control	a list of parameters for controlling the fitting process. See the documentation for glm.control for details.
rmap	an optional list to be used if the variables are not organized and named in the same way as in the ratetable object. See details below.
...	other arguments will be passed to glm.control.

Details

NOTE: The follow-up time must be specified in days. The ratetable being used may have different variable names and formats than the user's data set, this is dealt with by the rmap argument. For example, if age is in years in the data set but in days in the ratetable object, age=age*365.241 should be used. The calendar year can be in any date format (Date and POSIXt are allowed), the date formats in the ratetable and in the data may differ.

The maximum likelihood method and both glm methods assume a fully parametric model with a piecewise constant baseline excess hazard function. The intervals on which the baseline is assumed constant should be passed via argument int. The EM method is semiparametric, i.e. no assumptions are made for the baseline hazard and therefore no intervals need to be specified.

The methods using glm are methods for grouped data. The groups are formed according to the covariate values. This should be taken into account when fitting a model. The glm method returns life tables for groups specified by the covariates in groups.

The EM method output includes the smoothed baseline excess hazard lambda0, the cumulative baseline excess hazard Lambda0 and times at which they are estimated. The individual probabilites of dying due to the excess risk are returned as Nie. The EM method fitting procedure requires some local smoothing of the baseline excess hazard. The default bwin=-1 value lets the function find an appropriate value for the smoothing band width. While this ensures an unbiased estimate, the procedure time is much longer. As the value found by the function is independent of the covariates in the model, the value can be read from the output (bwinfac) and used for refitting different models to the same data to save time.

Value

An object of class `rsadd`. In the case of `method="glm.bin"` and `method="glm.poi"` the class also inherits from `glm` which inherits from the class `lm`. Objects of this class have methods for the functions `print` and `summary`. An object of class `rsadd` is a list containing at least the following components:

data   the data as used in the model, along with the variables defined in the rate table

ratetable   the ratetable used.

int   the maximum time (in years) used. All the events at and after this value are
                censored.

method   the fitting method that was used.

linear.predictors
                the vector of linear predictors, one per subject.

References

Package. Pohar M., Stare J. (2006) "Relative survival analysis in R." Computer Methods and Programs in Biomedicine, **81**: 272–278

Relative survival: Pohar, M., Stare, J. (2007) "Making relative survival analysis relatively easy." Computers in biology and medicine, **37**: 1741–1749.

EM algorithm: Pohar Perme M., Henderson R., Stare, J. (2009) "An approach to estimation in relative survival regression." Biostatistics, **10**: 136–146.

See Also

rstrans, rsmul

Examples

```
data(slopop)
data(rdata)
#fit an additive model
#note that the variable year is given in days since 01.01.1960 and that
#age must be multiplied by 365.241 in order to be expressed in days.
fit <- rsadd(Surv(time,cens)~sex+as.factor(agegr)+ratetable(age=age*365.241),
    ratetable=slopop,data=rdata,int=5)

#check the goodness of fit
rs.br(fit)

#use the EM method and plot the smoothed baseline excess hazard
fit <- rsadd(Surv(time,cens)~sex+age,rmap=list(age=age*365.241),
 ratetable=slopop,data=rdata,int=5,method="EM")
sm <- epa(fit)
plot(sm$times,sm$lambda,type="l")
```

---

rsmul   *Fit Andersen et al Multiplicative Regression Model for Relative Sur-*
                         vival

Description

Fits the Andersen et al multiplicative regression model in relative survival. An extension of the
coxph function using relative survival.

Usage

```
rsmul(
  formula = formula(data),
  data = parent.frame(),
  ratetable = relsurv::slopop,
  int,
  na.action,
  init,
  method = "mul",
  control,
  rmap,
  ...
)
```

Arguments

formula	a formula object, with the response as a Surv object on the left of a ~ operator, and, if desired, terms separated by the + operator on the right. NOTE: The follow-up time must be in days.
data	a data.frame in which to interpret the variables named in the formula.
ratetable	a table of event rates, such as slopop.
int	the number of follow-up years used for calculating survival(the data are censored after this time-point). If missing, it is set the the maximum observed follow-up time.
na.action	a missing-data filter function, applied to the model.frame, after any subset argument has been used. Default is options()$na.action.
init	vector of initial values of the iteration. Default initial value is zero for all variables.
method	the default method mul assumes hazard to be constant on yearly intervals. Method mul1 uses the ratetable to determine the time points when hazard changes. The mul1 method is therefore more accurate, but at the same time can be more computationally intensive.
control	a list of parameters for controlling the fitting process. See the documentation for coxph.control for details.

rmap	an optional list to be used if the variables are not organized and named in the same way as in the `ratetable` object. See details below.
...	Other arguments will be passed to `coxph.control`.

Details

NOTE: The follow-up time must be specified in days. The `ratetable` being used may have different variable names and formats than the user's data set, this is dealt with by the `rmap` argument. For example, if age is in years in the data set but in days in the `ratetable` object, age=age*365.241 should be used. The calendar year can be in any date format (Date and POSIXt are allowed), the date formats in the `ratetable` and in the data may differ.

Value

an object of class `coxph` with an additional item:

basehaz	Cumulative baseline hazard (population values are seen as offset) at centered values of covariates.

References

Method: Andersen, P.K., Borch-Johnsen, K., Deckert, T., Green, A., Hougaard, P., Keiding, N. and Kreiner, S. (1985) "A Cox regression model for relative mortality and its application to diabetes mellitus survival data.", Biometrics, **41**: 921–932.

Package. Pohar M., Stare J. (2006) "Relative survival analysis in R." Computer Methods and Programs in Biomedicine, **81**: 272–278

Relative survival: Pohar, M., Stare, J. (2007) "Making relative survival analysis relatively easy." Computers in biology and medicine, **37**: 1741–1749.

See Also

rsadd, rstrans.

Examples

```
data(slopop)
data(rdata)
#fit a multiplicative model
#note that the variable year is given in days since 01.01.1960 and that
#age must be multiplied by 365.241 in order to be expressed in days.
fit <- rsmul(Surv(time,cens)~sex+as.factor(agegr),rmap=list(age=age*365.241),
             ratetable=slopop,data=rdata)


#check the goodness of fit
rs.br(fit)
```

---

rstrans *Fit Cox Proportional Hazards Model in Transformed Time*

Description

The function transforms each person's time to his/her probability of dying at that time according to the ratetable. It then fits the Cox proportional hazards model with the transformed times as a response. It can also be used for calculatin the transformed times (no covariates are needed in the formula for that purpose).

Usage

```
rstrans(
  formula = formula(data),
  data = parent.frame(),
  ratetable = relsurv::slopop,
  int,
  na.action,
  init,
  control,
  rmap,
  ...
)
```

Arguments

formula	a formula object, with the response as a `Surv` object on the left of a ~ operator, and, if desired, terms separated by the + operator on the right.
	NOTE: The follow-up time must be in days.
data	a data.frame in which to interpret the variables named in the `formula`.
ratetable	a table of event rates, such as `slopop`.
int	the number of follow-up years used for calculating survival(the rest is censored). If missing, it is set the the maximum observed follow-up time.
na.action	a missing-data filter function, applied to the model.frame, after any subset argument has been used. Default is `options()$na.action`.
init	vector of initial values of the iteration. Default initial value is zero for all variables.
control	a list of parameters for controlling the fitting process. See the documentation for `coxph.control` for details.
rmap	an optional list to be used if the variables are not organized and named in the same way as in the `ratetable` object. See details below.
...	other arguments will be passed to `coxph.control`.

Details

NOTE: The follow-up time must be specified in days. The `ratetable` being used may have different variable names and formats than the user's data set, this is dealt with by the `rmap` argument. For example, if age is in years in the data set but in days in the `ratetable` object, age=age*365.241 should be used. The calendar year can be in any date format (Date and POSIXt are allowed), the date formats in the `ratetable` and in the data may differ. A side product of this function are the transformed times - stored in teh y object of the output. To get these times, covariates are of course irrelevant.

Value

an object of class `coxph`. See `coxph.object` and `coxph.detail` for details.

y   an object of class `Surv` containing the transformed times (these times do not depend on covariates).

References

Method: Stare J., Henderson R., Pohar M. (2005) "An individual measure for relative survival." Journal of the Royal Statistical Society: Series C, **54** 115–126.

Package. Pohar M., Stare J. (2006) "Relative survival analysis in R." Computer Methods and Programs in Biomedicine, **81**: 272–278

Relative survival: Pohar, M., Stare, J. (2007) "Making relative survival analysis relatively easy." Computers in biology and medicine, **37**: 1741–1749.

See Also

`rsmul`, `invtime`, `rsadd`, `survival::survexp`.

Examples

```
data(slopop)
data(rdata)

#fit a Cox model using the transformed times
#note that the variable year is given in days since 01.01.1960 and that
#age must be multiplied by 365.241 in order to be expressed in days.
fit <- rstrans(Surv(time,cens)~sex+as.factor(agegr),rmap=list(age=age*365.241,
        sex=sex,year=year),ratetable=slopop,data=rdata)


#check the goodness of fit
rs.br(fit)
```

---

slopop   *Census Data Set for the Slovene Population*

Description

Census data set for the Slovene population.

Usage

```
data(slopop)
```

Examples

```
data(slopop)
```

---

summary.cmp.rel   *Summary of the crude probability of death*

Description

Returns a list containing the estimated values at required times.

Usage

```
## S3 method for class 'cmp.rel'
summary(object, times, scale = 365.241, area = FALSE, ...)
```

Arguments

object	output of the function `cmp.rel`.
times	the times at which the output is required.
scale	The time scale in which the times are specified. The default value is 1, i.e. days.
area	Should area under the curves at time `tau` be printed out? Default is `FALSE`.
...	Additional arguments, currently not implemented

Details

The variance is calculated using numerical integration. If the required time is not a time at which the value was estimated, the value at the last time before it is reported. The density of the time points is set by the `precision` argument in the `cmp.rel` function.

Value

A list of values is returned.

See Also

cmp.rel

Examples

```
data(slopop)
data(rdata)
#calculate the crude probability of death and summarize it
fit <- cmp.rel(Surv(time,cens)~sex,rmap=list(age=age*365),
     ratetable=slopop,data=rdata,tau=3652.41)
summary(fit,c(1,3),scale=365.241)
```

---

survaalen	*Fit an additive hazards model*

Description

Fits the Aalen additive hazard model. The function can be used for multi-state model data (as in the package mstate; class msdata) by supplying the start and stop times in the Surv object and adding a strata(trans) object in the formula (where trans denotes the transition in the multi-state model).

Usage

```
survaalen(formula, data, variance = FALSE, var_estimator = "dN")
```

Arguments

formula	a formula object, with the response as a Surv object on the left of a ~ operator, and, if desired, terms separated by the + operator on the right.
data	a data.frame in which to interpret the variables named in the formula.
variance	a logical value indicating whether the variances of the hazards should be computed. Default is FALSE.
var_estimator	Choose variance estimator. The default option 'dN' uses dN(t) in the variance formula, see formula 4.63 in Aalen et al. (2008). Option 'XdB' uses X*dB(t), see formula 4.64 in Aalen et al. (2008).

Value

An object of class aalen.model.

Author(s)

Damjan Manevski

See Also

rsaalen

Examples

```
# Survival:
data(rdata)
mod <- survaalen(Surv(time, cens)~sex+age, data=rdata)
head(mod$coefficients)
tail(mod$coefficients)

# Multi-state model:
data(ebmt1wide)
mod <- survaalen(Surv(Tstart, Tstop, status)~age.1+age.2+age.3+strata(trans), data=ebmt1wide)
head(mod$coefficients$trans1)
head(mod$coefficients$trans2)
head(mod$coefficients$trans3)
```

---

survfit.rsadd *Compute a Predicted Survival Curve*

Description

Computes a predicted survival curve based on the additive model estimated by rsadd function.

Usage

```
## S3 method for class 'rsadd'
survfit(
  formula,
  newdata,
  se.fit = TRUE,
  conf.int = 0.95,
  individual = FALSE,
  conf.type = c("log", "log-log", "plain", "none"),
  ...
)
```

Arguments

formula	a rsadd object
newdata	a data frame with the same variable names as those that appear in the rsadd formula. The curve(s) produced will be representative of a cohort who's covariates correspond to the values in newdata.
se.fit	a logical value indicating whether standard errors should be computed. Default is TRUE.
conf.int	the level for a two-sided confidence interval on the survival curve(s). Default is 0.95.

individual	a logical value indicating whether the data frame represents different time epochs for only one individual (T), or whether multiple rows indicate multiple individuals (F, the default). If the former only one curve will be produced; if the latter there will be one curve per row in newdata.
conf.type	One of none, plain, log (the default), or log-log. The first option causes confidence intervals not to be generated. The second causes the standard intervals curve +- k *se(curve), where k is determined from conf.int. The log option calculates intervals based on the cumulative hazard or log(survival). The last option bases intervals on the log hazard or log(-log(survival)).
...	Currently not implemented

Details

When predicting the survival curve, the ratetable values for future years will be equal to those of the last given year. The same ratetables will be used for fitting and predicting. To predict a relative survival curve, use rs.surv.rsadd.

Value

a survfit object; see the help on survfit.object for details. The survfit methods are used for print, plot, lines, and points.

References

Package: Pohar M., Stare J. (2006) "Relative survival analysis in R." Computer Methods and Programs in Biomedicine,**81**: 272–278.

Relative survival: Pohar, M., Stare, J. (2007) "Making relative survival analysis relatively easy." Computers in biology and medicine, **37**: 1741–1749.

See Also

survfit, survexp, rs.surv

Examples

```
data(slopop)
data(rdata)
#BTW: work on a smaller dataset here to run the example faster
fit <- rsadd(Surv(time,cens)~sex,rmap=list(age=age*365.241),
ratetable=slopop,data=rdata[1:500,],method="EM")
survfit.rsadd(fit,newdata=data.frame(sex=1,age=60,year=17000))
```

---

survsplit   *Split a Survival Data Set at Specified Times*

Description

Given a survival data set and a set of specified cut times, the function splits each record into multiple records at each cut time. The new data set is be in counting process format, with a start time, stop time, and event status for each record. More general than survSplit as it also works with the data already in the counting process format.

Usage

```
survsplit(
  data,
  cut,
  end,
  event,
  start,
  id = NULL,
  zero = 0,
  episode = NULL,
  interval = NULL
)
```

Arguments

data	data frame.
cut	vector of timepoints to cut at.
end	character string with name of event time variable.
event	character string with name of censoring indicator.
start	character string with name of start variable (will be created if it does not exist).
id	character string with name of new id variable to create (optional).
zero	If start doesn't already exist, this is the time that the original records start. May be a vector or single value.
episode	character string with name of new episode variable (optional).
interval	this argument is used by max.lik function

Value

New, longer, data frame.

See Also

survival::survSplit.

transrate	*Reorganize Data into a Ratetable Object*

Description

The function assists in reorganizing certain types of data into a ratetable object.

Usage

```
transrate(men, women, yearlim, int.length = 1)
```

Arguments

men	a matrix containing the yearly (conditional) probabilities of one year survival for men. Rows represent age (increasing 1 year per line,starting with 0), the columns represent cohort years (the limits are in `yearlim`, the increase is in `int.length`.
women	a matrix containing the yearly (conditional) probabilities of one year survival for women.
yearlim	the first and last cohort year given in the tables.
int.length	the length of intervals in which cohort years are given.

Details

This function only applies for ratetables that are organized by age, sex and year.

Value

An object of class `ratetable`.

References

Package. Pohar M., Stare J. (2006) "Relative survival analysis in R." Computer Methods and Programs in Biomedicine, **81**: 272–278

Relative survival: Pohar, M., Stare, J. (2007) "Making relative survival analysis relatively easy." Computers in biology and medicine, **37**: 1741–1749.

See Also

survival::ratetable.

Examples

```
men <- cbind(exp(-365.241*exp(-14.5+.08*(0:100))),exp(-365*exp(-14.7+.085*(0:100))))
women <- cbind(exp(-365.241*exp(-15.5+.085*(0:100))),exp(-365*exp(-15.7+.09*(0:100))))
table <- transrate(men,women,yearlim=c(1980,1990),int.length=10)
```

---

transrate.hld	*Reorganize Data obtained from Human Life-Table Database into a Ratetable Object*

Description

The function assists in reorganizing the .txt files obtained from Human Life-Table Database (http://www.lifetable.de -> Data by Country) into a ratetable object.

Usage

```
transrate.hld(file, cut.year, race)
```

Arguments

file	a vector of file names which the data are to be read from. Must be in .tex format and in the same format as the files in Human Life-Table Database.
cut.year	a vector of cutpoints for years. Must be specified when the year spans in the files are not consecutive.
race	a vector of race names for the input files.

Details

This function works with any table organised in the format provided by the Human Life-Table Database, but currently only works with TypeLT 1 (i.e. age intervals of length 1). The age must always start with value 0, but can end at different values (when that happens, the last value is carried forward). The rates between the cutpoints are taken to be constant.

Value

An object of class ratetable.

References

Package. Pohar M., Stare J. (2006) "Relative survival analysis in R." Computer Methods and Programs in Biomedicine, **81**: 272–278

Relative survival: Pohar, M., Stare, J. (2007) "Making relative survival analysis relatively easy." Computers in biology and medicine, **37**: 1741–1749.

See Also

survival::ratetable, transrate.hmd, joinrate, transrate.

Examples

```
## Not run:
finpop <- transrate.hld(c("FIN_1981-85.txt","FIN_1986-90.txt","FIN_1991-95.txt"))

## End(Not run)
## Not run:
nzpop <- transrate.hld(c("NZL_1980-82_Non-maori.txt","NZL_1985-87_Non-maori.txt",
 "NZL_1980-82_Maori.txt","NZL_1985-87_Maori.txt"),
 cut.year=c(1980,1985),race=rep(c("nonmaori","maori"),each=2))

## End(Not run)
```

---

transrate.hmd	*Reorganize Data obtained from Human Mortality Database into a Ratetable Object*

Description

The function assists in reorganizing the .txt files obtained from Human Mortality Database (http://www.mortality.org) into a ratetable object.

Usage

```
transrate.hmd(male, female)
```

Arguments

male   a .txt file, containing the data on males.

female   a .txt file, containing the data on females.

Details

This function works automatically with tables organised in the format provided by the Human Mortality Database. Download Life Tables for Males and Females separately from the column named 1x1 (period life tables, organized by date of death, yearly cutpoints for age as well as calendar year).

If you wish to provide the data in the required format by yourself, note that the only two columns needed are calendar year (Year) and probability of death (qx). Death probabilities must be calculated up to age 110 (in yearly intervals).

Value

An object of class `ratetable`.

References

Package. Pohar M., Stare J. (2006) "Relative survival analysis in R." Computer Methods and Programs in Biomedicine, **81**: 272–278

Relative survival: Pohar, M., Stare, J. (2007) "Making relative survival analysis relatively easy." Computers in biology and medicine, **37**: 1741–1749.

See Also

survival::ratetable, transrate.hld, joinrate, transrate.

Examples

```
## Not run:
auspop <- transrate.hmd("mltper_1x1.txt","fltper_1x1.txt")

## End(Not run)
```

---

years   *Compute one of the life years measures*

Description

Provides an estimate for one of the following measures: years lost (Andersen, 2013), years lost/saved (Andersen, 2017), or life years difference (Manevski, Ruzic Gorenjec, Andersen, Pohar Perme, 2022).

Usage

```
years(
  formula = formula(data),
  data,
  measure = c("yd", "yl2017", "yl2013"),
  ratetable = relsurv::slopop,
  rmap,
  var.estimator = c("none", "bootstrap", "greenwood"),
  B = 100,
  precision = 30,
  add.times,
  na.action = stats::na.omit,
  conf.int = 0.95,
  timefix = FALSE,
  is.boot = FALSE,
  first.boot = FALSE
)
```

Arguments

formula	a formula object, with the response as a Surv object on the left of a ~ operator, and, ~1 specified on the right.
	NOTE: The follow-up time must be in days.
data	a data.frame in which to interpret the variables named in the formula.
measure	choose which measure is used: 'yd' (life years difference; Manevski, Ruzic Gorenjec, Andersen, Pohar Perme, 2022), 'yl2017' (years lost/saved; Andersen 2017), 'yl2013' (years lost/saved; Andersen 2013).
ratetable	a table of event rates, organized as a ratetable object, such as slopop.
rmap	an optional list to be used if the variables are not organized and named in the same way as in the ratetable object. See details below.
var.estimator	Choose the estimator for the variance ('none', 'bootstrap', 'greenwood'). Default is 'none'. The 'greenwood' option is possible only for measure='yd'.
B	if var.estimator is 'bootstrap'. The number of bootstrap replications. Default is 100.
precision	precision for numerical integration of the population curve. Default is 30 (days). The value may be decreased to get a higher precision or increased to achieve a faster calculation.
add.times	specific times at which the curves should be reported.
na.action	a missing-data filter function. Default is na.omit.
conf.int	the confidence level for a two-sided confidence interval. Default is 0.95.
timefix	the timefix argument in survival::survfit.formula. Default is FALSE.
is.boot	if TRUE, the function years has been called during a bootstrap replication.
first.boot	if TRUE, this is the first bootstrap replication.

Details

The life years difference (measure='yd') is taken by default. If other measures are of interest, use the measure argument.

The follow-up time must be specified in days. The ratetable being used may have different variable names and formats than the user's data set, this is dealt with the rmap argument. For example, if age is in years in the data but in days in the ratetable object, age=age*365.241 should be used. The calendar year can be in any date format (Date and POSIXt are allowed), the date formats in the ratetable and in the data may differ.

Numerical integration is performed, argument precision is set with argument precision, which defaults to 30-day intervals for intergration. For higher accuracy take a smaller value (e.g. precision=1 makes the integration on a daily basis).

The observed curves are reported at event and censoring times. The population curves are reported at all times used for the numerical integration. Note that for the years lost (Andersen, 2013) measure, only the excess absolute risk is reported.

Value

A list containing the years measure, the observed and population curves (or the excess curve for Andersen 2013). The values are given as separate data.frames through time. Times are given in days, all areas are given in years. For `measure='yl2017'` values are reported only at the last time point. Functions `plot_f` and `plot_years` can be then used for plotting.

See Also

plot_f, plot_years

Examples

```
library(relsurv)
# Estimate the life years difference for the rdata dataset.
mod <- years(Surv(time, cens)~1, data=rdata, measure='yd', ratetable=slopop,
             rmap=list(age=age*365.241), var.estimator = 'none')
# Plot the absolute risk (observed and population curve):
plot_f(mod)
# Plot the life years difference estimate:
plot_years(mod, conf.int=FALSE)
```

Index