

Package ‘relsurv’

January 2, 2012

Title Relative survival

Version 1.6.4

Author Maja Pohar Perme

Description Various functions for relative survival analysis.

Maintainer Maja Pohar Perme <maja.pohar@mf.uni-lj.si>

Depends survival, splines, date

License GPL

Repository CRAN

Date/Publication 2011-06-02 13:31:21

R topics documented:

epa	2
invtime	3
joinrate	4
plot.rs.zph	5
rdata	6
residuals.rsadd	7
rs.br	8
rs.surv	9
rs.surv.rsadd	11
rs.zph	12
rsadd	13
rsmul	16
rstrans	18
rsurvaux	19
slopop	20
srvxp.fit	20
survfit.rsadd	21
survsplit	22

transrate	23
transrate.hld	24
transrate.hmd	25

Index	27
--------------	-----------

epa	<i>Excess hazard function smoothing</i>
-----	---

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

<code>fit</code>	Fit from the additive relative survival model using the EM method.
<code>bwin</code>	The relative width of the smoothing window (default is 1).
<code>times</code>	The times at which the smoother is to be evaluated. If missing, it is evaluated at all event times.
<code>n.bwin</code>	Number of times that the window width may change.
<code>left</code>	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:

<code>lambda</code>	the smoothed excess baseline hazard function
<code>times</code>	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+ratetable(age=age*365,
sex=sex,year=year), ratetable=slopop,data=rdata,int=5,method="EM")
sm <- epa(fit)
plot(sm$times,sm$lambda)
```

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, age, sex, year, scale, ratetable, lower, upper)
```

Arguments

<code>y</code>	time in Y.
<code>age</code>	age of the individual. Must be in days.
<code>sex</code>	sex of the individual. Must be coded in the same way as in the <code>ratetable</code> .
<code>year</code>	date of diagnosis. Must be in date format
<code>scale</code>	numeric value to scale the results. If <code>ratetable</code> is in units/day, <code>scale = 365.25</code> causes the output to be reported in years.
<code>ratetable</code>	a table of event rates, such as <code>survexp.us</code> .

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 = 9497, 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")
```

Arguments

tables	a list of ratetables. If names are given, they are included as dimnames.
dim.name	the name of the added dimension.

Details

This function joins two or more `ratetable` objects by adding a new dimension. The cutpoints of all the rate tables are compared and only the common intervals kept. If the intervals defined by the cutpoints are not of the same length, a warning message is displayed. Each rate table must have 3 dimensions, i.e. age, sex and year (the order is not important).

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

[ratetable](#), [transrate.hld](#), [transrate.hmd](#), [transrate](#).

Examples

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

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, scale=1, ...)
```

Arguments

<code>x</code>	result of the <code>rs.zph</code> function.
<code>resid</code>	a logical value, if <code>TRUE</code> the residuals are included on the plot, as well as the smooth fit.
<code>df</code>	the degrees of freedom for the fitted natural spline, <code>df=2</code> leads to a linear fit.
<code>nsmo</code>	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.
scale	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](#), [plot.cox.zph](#).

Examples

```
data(slopop)
data(rdata)
fit <- rsadd(Surv(time,cens)~sex+as.factor(agegr)+ratetable(age=age*365,
sex=sex,year=year),ratetable=slopop,data=rdata,int=5)
rszph <- rs.zph(fit)
plot(rszph)
```

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	<i>Calculate Residuals for a "rsadd" Fit</i>
-----------------	--

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+ratetable(age=age*365,sex=sex,year=year),
             ratetable=slopop,data=rdata,int=5)
sresid <- residuals.rsadd(fit)
```

rs.br

Test the Proportional Hazards Assumption for Relative Survival Regression 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 weights 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 weights 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 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+ratetable(age=age*365,sex=sex,
      year=year),ratetable=slopop,data=rdata,int=5)
rsbr <- rs.br(fit)
print(rsbr)
plot(rsbr)
```

rs.surv

Compute a Relative Survival Curve

Description

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

Usage

```
rs.surv(formula, data, ratetable=survexp.us, na.action, fin.date,
method="pohar-perme", conf.type="log", conf.int=0.95)
```

Arguments

formula	a formula object, with the response as a <code>Surv</code> object on the left of a <code>~</code> operator, and, if desired, terms separated by the <code>+</code> operator on the right. If the variables are not organized and named in the same way as in the population tables, a <code>ratetable</code> term must be added to match each subject to his/her expected cohort. For a single survival curve the <code>~ 1</code> part of the formula is not required. NOTE: The time must be in days, and the same is required for the <code>ratetable</code> variables (the variables used in the population tables), for example age and year (year must be given in the date format, i.e. in number of days since 01.01.1960).
data	a <code>data.frame</code> in which to interpret the variables named in the formula.
ratetable	a table of event rates, organized as a <code>ratetable</code> object, such as <code>survexp.us</code> .
na.action	a missing-data filter function, applied to the <code>model.frame</code> , after any subset argument has been used. Default is <code>options()\$na.action</code> .
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 <code>max(year+time)</code> .
method	the method for calculating the relative survival. The options are <code>pohar-perme</code> (default), <code>ederer</code> and <code>hakulinen</code> .
conf.type	one of <code>plain</code> , <code>log</code> (the default), or <code>log-log</code> . The first option causes the standard intervals curve $\pm k * se(\text{curve})$, where k is determined from <code>conf.int</code> . The <code>log</code> option calculates intervals based on the cumulative hazard or <code>log(survival)</code> . The last option bases intervals on the log hazard or <code>log(-log(survival))</code> .
conf.int	the level for a two-sided confidence interval on the survival curve(s). Default is 0.95.

Details

NOTE: All times used in the formula argument must be specified in days. This is true for the follow-up time as well as for any variables needed `ratetable` object, like age and year. On the contrary, the `int` argument requires interval specification in years.

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.

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

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 in order to be expressed in days.
rs.surv(Surv(time,cens)~sex+ratetable(age=age*365,sex=sex,
  year=year),ratetable=slopop,data=rdata)
```

rs.surv.rsadd	<i>Compute a Relative Survival Curve from an additive relative survival model</i>
---------------	---

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

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+ratetable(age=age*365,sex=sex,year=year),
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("1Jul1982")))
#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](#), [cox.zph](#).

Examples

```
data(slopop)
data(rdata)
fit <- rsadd(Surv(time,cens)~sex+ratetable(age=age*365,sex=sex,
      year=year),ratetable=slopop,data=rdata,int=5)
rszph <- rs.zph(fit)
```

 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, data=parent.frame(), ratetable = survexp.us,
      int, na.action, method, init,bwin,centered,cause,control,...)
```

Arguments

`formula` a formula object, with the response on the left of a `~` operator, and the terms on the right. The terms consist of predictor variables separated by the `+` operator, along with a `ratetable` term. The `ratetable` term matches each subject to his/her expected cohort. If the variables are organized and named in the same

way as in the population tables, the `ratetable` term can be omitted. The response must be a survival object as returned by the `Surv` function.

NOTE: The time must be in days, and the same is required for the `ratetable` variables (the variables used in the population tables), for example age and year (year must be given in the date format, i.e. in number of days since 01.01.1960).

<code>data</code>	a <code>data.frame</code> in which to interpret the variables named in the formula.
<code>ratetable</code>	a table of event rates, organized as a <code>ratetable</code> object, such as <code>survexp.us</code> .
<code>int</code>	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 <code>max(int)</code> 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).
<code>na.action</code>	a missing-data filter function, applied to the <code>model.frame</code> , after any subset argument has been used. Default is <code>options()\$na.action</code> .
<code>method</code>	<code>glm.bin</code> or <code>glm.poi</code> for a <code>glm</code> model, EM for the EM algorithm and <code>max.lik</code> for the maximum likelihood model (default).
<code>init</code>	vector of initial values of the iteration. Default initial value is zero for all variables.
<code>bwin</code>	controls the bandwidth used for smoothing in the EM algorithm. The follow-up time is divided into quartiles and <code>bwin</code> specifies a factor by which the maximum between events time length on each interval is multiplied. The default <code>bwin=-1</code> lets the function find an appropriate value. If <code>bwin=0</code> , no smoothing is applied.
<code>centered</code>	if TRUE, all the variables are centered before fitting and the baseline excess hazard is calculated accordingly. Default is FALSE.
<code>cause</code>	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.
<code>control</code>	a list of parameters for controlling the fitting process. See the documentation for <code>glm.control</code> for details.
<code>...</code>	other arguments will be passed to <code>glm.control</code> .

Details

NOTE: All times used in the formula argument must be specified in days. This is true for the follow-up time as well as for any variables needed `ratetable` object, like age and year. On the contrary, the `int` argument requires interval specification in years.

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 λ_0 , the cumulative baseline excess hazard Λ_0 and times at which they are estimated. The individual probabilities 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:

<code>data</code>	the data as used in the model, along with the variables defined in the rate table
<code>ratetable</code>	the ratetable used.
<code>int</code>	the maximum time (in years) used. All the events at and after this value are censored.
<code>method</code>	the fitting method that was used.
<code>linear.predictors</code>	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 in order to be expressed in days.
fit <- rsadd(Surv(time,cens)~sex+as.factor(agegr)+ratetable(age=age*365,
  sex=sex,year=year), 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+ratetable(age=age*365,
  sex=sex,year=year), ratetable=slopop,data=rdata,int=5,method="EM")
sm <- epa(fit)
plot(sm$times,sm$lambda,type="l")
```

rsmul	<i>Fit Andersen et al Multiplicative Regression Model for Relative Survival</i>
-------	---

Description

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

Usage

```
rsmul(formula, data, ratetable = survexp.us, int,na.action,init,
  method,control,...)
```

Arguments

formula	a formula object, with the response on the left of a \sim operator, and the terms on the right. The terms consist of predictor variables separated by the + operator, along with a ratetable term. The ratetable term matches each subject to his/her expected cohort. If the variables are organized and named in the same way as in the population tables, the ratetable term can be omitted. The response must be a survival object as returned by the Surv function. NOTE: The time must be in days, and the same is required for the ratetable variables (the variables used in the population tables), for example age and year (year must be given in the date format, i.e. in number of days since 01.01.1960).
data	a data.frame in which to interpret the variables named in the formula.
ratetable	a table of event rates, such as survexp.us.
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.

... Other arguments will be passed to `coxph.control`.

Details

NOTE: All times used in the formula argument must be specified in days. This is true for the follow-up time as well as for any variables needed `ratetable` object, like `age` and `year`. On the contrary, the `int` argument requires interval specification in years.

Value

an object of class `coxph`.

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 in order to be expressed in days.
fit <- rsmul(Surv(time,cens)~sex+as.factor(agegr)+ratetable(age=age*365,
  sex=sex,year=year),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 calculating the transformed times (no covariates are needed in the formula for that purpose).

Usage

```
rstrans(formula, data, ratetable, int, na.action, init, control, ...)
```

Arguments

formula	a formula object, with the response on the left of a \sim operator, and the terms on the right. The terms consist of predictor variables separated by the + operator, along with a ratetable term. The ratetable term matches each subject to his/her expected cohort. If the variables are organized and named in the same way as in the population tables, the ratetable term can be omitted. The response must be a survival object as returned by the Surv function. NOTE: The time must be in days, and the same is required for the ratetable variables (the variables used in the population tables), for example age and year (year must be given in the date format, i.e. in number of days since 01.01.1960).
data	a data.frame in which to interpret the variables named in the formula.
ratetable	a table of event rates, such as survexp.us.
int	the number of follow-up years used for calculating survival (the rest is censored). If missing, it is set to 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.
...	other arguments will be passed to coxph.control.

Details

NOTE: All times used in the formula argument must be specified in days. This is true for the follow-up time as well as for any variables needed in the ratetable object, like age and year. On the contrary, the int argument requires interval specification in years.

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](#), [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 in order to be expressed in days.
fit <- rstrans(Surv(time,cens)~sex+as.factor(agegr)+ratetable(age=age*365,
  sex=sex,year=year),ratetable=slopop,data=rdata)

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

rsurvaux

Auxiliary Function for relsurv

Description

This function is not called by the user.

`slopop`*Census Data Set for the Slovene Population*

Description

Census data set for the Slovene population.

Usage

```
data(slopop)
```

Examples

```
data(slopop)
```

`srvxp.fit`*Work Function to Compute Expected Survival*

Description

Compute expected survival for a group of people. This function is not to be called by the user.

Usage

```
srvxp.fit(x, y, ratetable)
```

Arguments

<code>x</code>	a data.frame or a matrix. Each row represents one person. The columns must match the dimensions of the <code>ratetable</code> , in the correct order.
<code>y</code>	a vector of survival times of the same length as the number of rows of <code>x</code> .
<code>ratetable</code>	a table of event rates, organized as a <code>ratetable</code> object, such as <code>survexp.us</code> .

Value

A vector of survival times.

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 $\text{curve} \pm k * \text{se}(\text{curve})$, where k is determined from conf.int. The log option calculates intervals based on the cumulative hazard or $\log(\text{survival})$. The last option bases intervals on the log hazard or $\log(-\log(\text{survival}))$.
...	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)
fit <- rsadd(Surv(time,cens)~sex+ratetable(age=age*365,sex=sex,
      year=year),ratetable=slopop,data=rdata,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

[survSplit](#).

transrate	<i>Reorganize Data into a Ratetable Object</i>
-----------	--

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

[ratetable](#).

Examples

```
men <- cbind(exp(-365*exp(-14.5+.08*(0:100))),exp(-365*exp(-14.7+.085*(0:100))))
women <- cbind(exp(-365*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	<i>Reorganize Data obtained from Human Life-Table Database into a Ratetable Object</i>
---------------	--

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

[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:
nzipop <- 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	<i>Reorganize Data obtained from Human Mortality Database into a Ratetable Object</i>
---------------	---

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

[ratetable](#), [transrate.hld](#), [joinrate](#), [transrate](#).

Examples

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

Index

*Topic **datasets**

rdata, 6
slopop, 20

*Topic **survival**

epa, 2
invtime, 3
joinrate, 4
plot.rs.zph, 5
residuals.rsadd, 7
rs.br, 8
rs.surv, 9
rs.surv.rsadd, 11
rs.zph, 12
rsadd, 13
rsmul, 16
rstrans, 18
rsurvaux, 19
srvxp.fit, 20
survfit.rsadd, 21
survsplit, 22
transrate, 23
transrate.hld, 24
transrate.hmd, 25

cox.zph, 13

em(rsadd), 13
epa, 2

glmxp(rsadd), 13

invtime, 3, 19

joinrate, 4, 24, 26

Kern(epa), 2
kern(epa), 2
kernerleftch(rsurvaux), 19
Kernmatch(rsurvaux), 19

lik.fit(rsadd), 13

maxlik(rsadd), 13
mlfit(rsurvaux), 19

plot.cox.zph, 6
plot.rs.br(rs.br), 8
plot.rs.zph, 5
print.rs.br(rs.br), 8
print.rsadd(rsadd), 13
print.summary.rsadd(rsadd), 13

ratetable, 5, 23, 24, 26
rdata, 6
resid, 9, 13
resid.com(residuals.rsadd), 7
residuals.rsadd, 7
rformulate(rsurvaux), 19
rs.br, 8
rs.surv, 9, 22
rs.surv.rsadd, 11
rs.survo(rs.surv), 9
rs.zph, 6, 12
rsadd, 3, 8, 9, 13, 13, 17, 19
rsfitterem(rsadd), 13
rsmul, 15, 16, 19
rstrans, 4, 15, 17, 18
rsurvaux, 19

slopop, 20
srvxp.fit, 20
summary.rsadd(rsadd), 13
survexp, 19
survfit.rsadd, 21
survSplit, 23
survsplit, 22

transrate, 5, 23, 24, 26
transrate.hld, 5, 24, 26
transrate.hmd, 5, 24, 25