



Relative survival analysis in R

Maja Pohar^{a,*}, Janez Stare^b

^a Department of Medical Informatics, University of Ljubljana, Vrazov trg 2, SI-1000 Ljubljana, Slovenia

^b Department of Medical Informatics, University of Ljubljana, Slovenia

ARTICLE INFO

Article history:

Received 7 March 2005

Received in revised form 12 January 2006

Accepted 12 January 2006

Keywords:

Relative survival

R software

Regression

ABSTRACT

Relative survival techniques are used to compare the survival experience in a study cohort with the one expected should they follow the background population mortality rates. The techniques are especially useful when the cause-specific death information is not accurate or not available since they provide a measure of excess mortality in a group of patients with a certain disease. There are several approaches to modeling relative survival, but there is no widely used statistical package that would incorporate the relevant techniques. The existing software was mostly written by the authors of different methods, in different computer languages and with different requirements for the data input, which makes it almost impossible for a user to choose between available models. We describe our R package `reلسurv` that provides functions for easy and flexible fitting of several relative survival regression models.

© 2006 Elsevier Ireland Ltd. All rights reserved.

1. Introduction

Survival analysis is concerned with studying the random variable T , representing the time between entry to a study and some event of interest (e.g. death, recurrence of disease ...). Instead of the cumulative distribution function $F(t)$, we usually estimate the cumulative survival function $S(t)$, which is defined as $S(t) = P(T > t) = 1 - F(t)$. In the case of the final event being death, $S(t)$ is the probability of being alive at time t . When the final event of interest is death due to a certain disease, but the causes of death are not known, it is not possible to directly estimate the proportion of dead due to the disease in question. We then resort to the methods of relative survival. The cumulative relative survival function $r(t)$ is defined [1] as

$$r(t) := \frac{S_O(t)}{S_P(t)} \quad (1)$$

where $S_O(t)$ denotes observed survival and $S_P(t)$ stands for population or expected survival, which is estimated on the basis of

population life tables. Obviously, $r(t)$ can be any non-negative number, although the methods are most often applied to data where $r(t)$ is less than 1.

Several approaches to modelling relative survival exist, but all of the existing programmes (for example: `Surv3` [2], SAS macros and Stata functions [3], `RSurv` R function [4]) focus on only one of the models and use specifically organised general population tables, making it difficult for the users to compare different methods.

We present three R [5] functions organised as a package called `reلسurv` that largely simplify the usage of relative survival regression models. All the functions require the same basic organisation of the data and can be used with any format of the population mortality data.

Section 2 briefly describes the three most commonly used regression approaches and gives an outline of the theory for the five fitting methods used in the `reلسurv` functions. Section 3 describes the functions, their arguments, the preparation of the data and the returned values. The usage is further explained through an example in Section 4. In case the user does not yet have the relevant population mortality tables organ-

* Corresponding author. Tel.: +386 1 543 7785; fax: +386 1 543 7771.

E-mail address: maja.pohar@mf.uni-lj.si (M. Pohar).

ised in the required `ratetable` format, the appendix provides some quick help.

2. Relative survival regression models

The relative survival literature most frequently refers to the *additive models*, dominating especially in cancer research. The main assumption is that the hazard of every individual can be split into two additive components:

$$\lambda_O(t) = \lambda_P(t) + \lambda_E(t), \tag{2}$$

where $\lambda_P(t)$ is the hazard every patient takes because of his age, sex and cohort year (or any other combination of covariates included in the population mortality data). $\lambda_E(t)$ denotes the excess hazard, specific for the disease in question and $\lambda_O(t)$ is used for the observed hazard - the one we can estimate from our data. Using the equality

$$S(t) = e^{-\int \lambda(t) dt}, \tag{3}$$

we can write

$$S_O(t) = S_P(t) \times S_E(t) \implies S_E(t) = \frac{S_O(t)}{S_P(t)}, \tag{4}$$

giving the same expression as 1. Several fitting approaches for the additive model exist and a review of the three methods described in this paper is given in [6].

The *multiplicative models* assume the hazard components to be multiplicative:

$$\lambda_O(t) = \lambda_P(t) \cdot \lambda_E(t). \tag{5}$$

Such a model does not assume that the observed hazard is always greater than the population hazard, but has a less obvious interpretation as the additive model.

The third option are the *transformation models* [7] that make no assumption about the relationship between the observed, the population and the excess hazard. All the individual survival times are first transformed to a different scale (by taking into account the general population mortality), where they can be further analysed by any of the ordinary survival models.

Our programs provide for fitting the transformation model, the Andersen multiplicative model [8], and three different approaches to fitting the additive model. Only the last four have been widely used in the past, while the transformation approach has only recently been published and given its simplicity, it might become popular in the future, especially in studies where long-term observations are common.

- (i) Hakulinen–Tenkanen additive survival method [9]: in order to fit the model, the patients must first be grouped into K strata, indexed by k , with one stratum for each combination of the relevant predictor variables (age, sex, cohort year, stage ...) and a life table must be estimated for each stratum, with intervals indexed by i . The $\lambda_E(t)$ is assumed to be a multiplicative function of the covariates and constant within each time interval i . We write it in

the form $\lambda_E(t) = \exp\{\beta'z + \gamma'fu(t)\}$, where z is the vector of the predictor variables, fu the vector of the follow-up interval indicators (with components fu_i , where $fu_i(t) = 1$ on interval i and 0 elsewhere) and β and γ are the vectors of regression coefficients. If we denote the interval-specific observed and expected survival proportions by p_{ki} and p_{ki}^* respectively, Eqs. (2)–(4) lead us to

$$\ln \left(-\ln \left(\frac{p_{ki}}{p_{ki}^*} \right) \right) = \beta'z + \gamma_i, \tag{6}$$

which is a generalized linear model with a binomial error structure and a complementary log–log link function combined with a division by p_{ki}^* .

- (ii) The glm model with the Poisson error structure [6]: the method is similar to Hakulinen–Tenkanen and it requires the same grouping of the data. Again the λ_E is assumed to be of the $\lambda_E(t) = \exp\{\beta'z + \gamma'fu(t)\}$ form. The number of deaths d_{ki} is assumed to be distributed as $\text{Poisson}(\mu_{ki})$, where $\mu_{ki} = \lambda_{E,ki} \cdot y_{ki}$ and y_{ki} is the person-time at risk for group k of observations on time interval i . Denoting the expected number of deaths as d_{ki}^* , Eq. (2) can thus be written as

$$\frac{\mu_{ki}}{y_{ki}} = \frac{d_{ki}^*}{y_{ki}} + e^{\beta'z + \gamma_i} \tag{7}$$

$$\ln(\mu_{ki} - d_{ki}^*) = \beta'z + \gamma_i + \ln(y_{ki}). \tag{8}$$

This implies a generalized linear model with outcome d_{ki} , Poisson error structure, link $\ln(\mu_{ki} - d_{ki}^*)$ and offset $\ln(y_{ki})$.

The model can be rewritten [6] to yield the same likelihood function as the Estève model, meaning that the grouping of the variables is not crucial. The idea of splitting the observations into subject-band observations and taking d_{ki}^*/y_{ki} as a kind of an average of individual λ_P values however causes the estimates to slightly differ from those in the Estève model.

- (iii) The Estève additive survival model [10]: the method uses individual data and estimates the coefficients using the maximum likelihood approach. The likelihood for the additive model is

$$L(\beta) = \prod_{j=1}^n (\lambda_O(t_j))^{d_j} \exp \left(-\int_0^{t_j} \lambda_O(s) ds \right) = \prod_{j=1}^n (\lambda_P(t_j) + e^{\beta'z + \gamma'fu(t_j)})^{d_j} \exp \left(-\int_0^{t_j} (\lambda_P(s) + e^{\beta'z + \gamma'fu(s)}) ds \right), \tag{9}$$

where t_j is the event time for person j and d_j the status indicator. The log-likelihood function is then

$$l(\beta) = \sum_{j=1}^n d_j \ln(\lambda_P(t_j) + e^{\beta'z + \gamma'fu(t_j)}) - \sum_{j=1}^n \int_0^{t_j} e^{\beta'z + \gamma'fu(s)} ds - \sum_{j=1}^n \int_0^{t_j} \lambda_P(s) ds. \tag{10}$$

The last term on the right-hand side in Eq. (10) does not depend on β or γ and can be omitted. It follows that only the expected hazards at time of event are needed for each individual.

- (iv) The Andersen multiplicative model [8]: the model assumes $\lambda_E(t) = \lambda_0(t) \exp(\beta'z)$, which is the same as in the Cox model. The observed hazard thus becomes

$$\lambda_0(t) = \lambda_P(t) \lambda_0(t) e^{\beta'z}. \quad (11)$$

This model can be rewritten in the form

$$\lambda_0(t) = \lambda_0(t) e^{\beta'z + 1 \ln(\lambda_P(t))}, \quad (12)$$

from which it is obvious that this is a Cox model with an additional time-dependent variable with the coefficient fixed to 1. Though $\lambda_P(t)$ is a continuous process, the population mortality tables are usually organised in yearly intervals. In order to fit the model using the Cox model procedures, data can be split in yearly intervals, with the $\lambda_P(t)$ updated on each of the intervals.

- (v) The transformation approach [7]: the individual survival times t are first transformed as

$$y = F_P(t), \quad (13)$$

where F_P is the cumulative distribution function of a person of a certain age, sex and cohort year (or any other combination included in the population tables) that would apply if the person is a typical representative of the general population. This distribution function is calculated from the general population mortality data. The values y can thus be interpreted as the achieved values on the expected cumulative distribution function for each individual. By transforming to the new scale, the population hazard is automatically taken into account, consequently all what is left is precisely the disease-specific hazard, which we can thus directly model. One of the possibilities is to use the Cox model

$$\lambda(y) = \lambda_0(y) e^{\beta'z}. \quad (14)$$

3. The relsurv package

The core of the package are three functions that fit the models described in the previous section:

- `rsadd` fits an additive model. As described in Section 2, different methods of estimation exist, and the user can choose among them through the `method` argument. The default is the Estève method (iii) which is specified by `"max.lik"`, the other two options are `method="glm.bin"` and `method="glm.poi"` for models described in (i) and (ii) of Section 2. When using one of the glm methods, the observed and expected survival proportions for each group are returned as object `groups`.
- `rsmul` fits the multiplicative model as described in (iv). A more computationally intensive alternative is chosen by specifying `method="mul1"` that splits the intervals at every time point in which the population mortality

changes—for example on 1 January and on the individual's birthday.

- `rstrans` fits the transformation model as described in (v). If only the transformation times are needed, this can be done directly by the `survexp` function (`survival` package) or by function `rstrans`, where the transformed times are returned in output value `y(fit<-rstrans(...), y<-fit$y)`.

The package also includes the functions for testing goodness-of-fit and plotting relevant graphs for all the above models with methods described in [11]. Additionally, two data sets are included in the package. One is called `rdata` and contains survival data that can be used as an example (see Section 4 for more details). The `slopop` data set contains the population mortality tables for Slovenia.

3.1. Usage

All the functions are called in the same way, for example:

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

Two data sets are required for any relative survival model. One is our observed data set, which we will pass to the function as argument `data`. The other is the population mortality table that we want to compare our observed data to, and this is specified in the argument `ratetable`. The population mortality tables have to be organised as an object of class `ratetable` (defined in package `survival`), default is the `survexp.us` table that contains the US data (also in the `survival` package).

The model is specified within the argument `formula`. It is organised following the syntactic rules of an R language formula and thus similar to any other `survival` package formula. The left-hand side must be a `Surv` object. For example, if `time` and `status` are the survival time variable and the censoring indicator, and `x` is a covariate, then the command may be

```
rstrans(Surv(time, status)~x)
```

If the variables that are used for the expected survival calculation (for example age, sex and year) are not organised and not named in the same way as in the population tables, a special term `ratetable` is to be included in the formula, for example:

```
Surv(time, status)~x+ratetable(age=age, sex="male",
                              year=diag)
```

Argument `int` specifies the number of yearly intervals to be used in `rsadd` function. `int` can either be a single number or a vector. The latter is used when the intervals are not all one year long, for example `int=c(0, 0.5, 1, 5, 10)` means a couple of short intervals at the beginning and longer intervals thereafter with the maximum follow-up time of 10 years. Each interval includes the right endpoint, but not the left one, except for the first interval which also includes the left endpoint. In this case the third interval would therefore be (1, 5].

This option can also be used in `rsmul` and `rstrans` functions, serving simply as the maximum follow-up time after

which all the data are censored, enabling in this way a more straightforward comparison of the models.

Missing data are handled in the usual way through the argument `na.action` and the initial values for the model can be specified via the argument `init`. The number of iterations and other fitting specifications can be specified through the `control` argument, which follows the `glm.control` function in the `rsadd` function and the `coxph.control` function in `rsmul` and `rstrans`.

3.2. Preparing data

The functions follow syntactic conventions of the R package `survival` and the data should be organised as required by the `survival` functions. In particular, this means that we need a time and a censoring value for each subject and any additional number of covariates we would like to include in the model. All the functions also support time-dependent data, with times and events organised in the standard way in a `Surv` object.

The main feature of any relative survival method is the comparison between the observed and population mortality data. The latter have to be organized as a `ratetable` object (see Appendix A for details) and specified in the `ratetable` argument of the used function. The only thing that remains to be done is matching the variables in our data set with the names of the variables that the `ratetable` is grouped by. Age in the `ratetable` object is always expressed in days and therefore the same must be done with the age in the observed data set. The calendar year must be specified in the `date` format and the coding of the sex variable must be done in the matching way (in the US and Slovene tables sex is defined as a factor variable with levels “male” and “female”, so either a factor variable or an integer with values 1 and 2 will work).

The user, however, does not have to change the data set—all the matching can be done by the help of a special term in the formula named `ratetable`. In order to construct this term, first check the names of the `ratetable` variables. For the `slopop` rate table the names are

```
> attributes(slopop)$dimid
"age" "year" "sex"
```

These names are now to be matched with the variable names (or any function of them), for example:

```
Surv(time, status)~x+ratetable(age=age*365.24,
sex="male", year=diag)
```

The names on the left of each equality sign are the dimension names of the `ratetable` names, while the names on the right are the variable names in the observed data set. This example is for a data set in which no variable sex is included as all the patients are male, the calendar year variable is stored as `diag` (diagnosis year), and age is in years and must therefore be multiplied by 365.24 to match the data in the rate table (note that .24 is used in all R calculations as there are 24 leap years per century).

If the variables in the observed data set are of the same type and have the same names as the `ratetable` variables, the term `ratetable` can be omitted from the formula, i.e. the call

```
rsadd(Surv(time, status)~x, ratetable=slopop,
data=my.data)
```

assumes that the data set `my.data` has variables `age`, `sex` and `year` organised in the format that matches `slopop` rate table.

3.3. Values

The `rsadd` function returns an object of class `rsadd`, while `rsmul` and `rstrans` return a `coxph` object. All the returned values can be viewed as such or by the help of `summary` function. Apart from the coefficients, their standardized values and the other usually returned values, the common object returned by all the functions is named `y` and contains the survival and censoring times for each individual, organised as a `Surv` object. In the case of `rstrans` function, these are already the transformed times and this function can thus also be used just for making the transformation.

The `max.lik` method of `rsadd` function is based on maximum likelihood and therefore the output also contains the log-likelihood values stored as `loglik`. The other two methods are based on the `glm` function and the returned objects preserve all its values. The original and the grouped data are both stored in the output object as `data` and `groups`, respectively. The original data set contains the demographic variables in the format that matches the rate table (stored as `X1`, `X2`, ...) and the covariates in the format in which they were used in the model. For each of the groups in the model, object `groups` contains the number at risk (`ld`), number of events (`nd`), and the values p_{ki}^* (`ps`), d_{ki}^* (`dstar`) and $\ln(y_{ki})$ (`lny`) defined in Section 2. This object can be useful in a more detailed analysis of the data. Additionally, a note about the number of the groups in which the observed number of deaths is smaller than the expected is reported (this gives a very basic indication about goodness of fit).

4. Example

To illustrate the usage of the program, we use a subset of data from the study of survival of patients after acute myocardial infarction that is included in the package as the file `rdata`. The data were collected in the study carried out at the University Clinical Center in Ljubljana and contain 1040 patients diagnosed between 1982 and 1986 and followed up until 1997. During this time 547 deaths occurred and as the causes of death are not given, this is a good example of the need of the relative survival methodology. The organisation of the data is as follows:

```
>rdata[1:2,]
  time   cens age sex  year   agegr
1 2657     1  68  2 24Jun82 62-70
2 1097     1  63  2 31Aug82 62-70
```

Time is measured in days and year of infarction is expressed in R date format. Age is measured in years and a categorical variable `agegr` containing four age categories (“under 54”, “54–61”, “62–70”, “71–95”) is formed. The censoring indicator is specified in variable `cens` and is coded 0 (censoring) and 1 (event).

Table 1 – Comparison of coefficient estimates (coef) and their standard errors (S.E.) for the three additive model fitting methods

	coef (S.E.)		
	Estève	Poisson	Hakulinen
Sex	0.90 (0.22)	0.86 (0.22)	0.89 (0.22)
Age 54–61	0.14 (0.32)	0.15 (0.31)	0.14 (0.31)
Age 62–70	0.54 (0.29)	0.52 (0.29)	0.55 (0.29)
Age 71–95	0.61 (0.31)	0.54 (0.31)	0.53 (0.31)
fu [0,1]	-4.22 (0.39)	-4.09 (0.37)	-4.11 (0.37)
fu (1,2]	-4.98 (0.43)	-4.89 (0.42)	-4.93 (0.42)
fu (2,3]	-5.02 (0.44)	-4.98 (0.44)	-5.01 (0.44)
fu (3,4]	-5.46 (0.56)	-5.42 (0.53)	-5.45 (0.53)
fu (4,5]	-4.98 (0.44)	-4.96 (0.46)	-5.00 (0.46)

We fit all the models to the data in the same way:

- Time and censoring are already in the right format, so we put them in the formula as `Surv(time, cens)`;
- `sex` and `agegr` (“under 54” is the reference group) are taken as covariates; `agegr` is forced to be a factor variable \implies `sex+as.factor(agegr)`;
- We concentrate on the first 5 years of follow-up (and assume the hazard is constant in yearly intervals for all the additive models) \implies `int=5`;
- The name of the observed data set is `rdata`, the study was performed in Slovenia, therefore we use Slovenian population tables \implies `data=rdata, ratetable=slopop`;
- Variables `sex` and `year` are in the same format as in the rate table `slopop`, age must be put into days \implies `ratetable(age=age*365.24, sex=sex, year=year)`.

The command lines for our five models therefore look very similar:

```
> rsadd(Surv(time, cens)~sex+as.factor(agegr) +
+   ratetable(age=age*365.24, sex=sex, year=year) ,
+   data=rdata, ratetable=slopop, int=5, method="glm.bin")
> rsadd(Surv(time, cens)~sex+as.factor(agegr) +
+   ratetable(age=age*365.24, sex=sex, year=year) ,
+   data=rdata, ratetable=slopop, int=5, method="glm.poi")
> rsadd(Surv(time, cens)~sex+as.factor(agegr) +
+   ratetable(age=age*365.24, sex=sex, year=year) ,
+   data=rdata, ratetable=slopop, int=5)
> rsmul(Surv(time, cens)~sex+as.factor(agegr) +
+   ratetable(age=age*365.24, sex=sex, year=year) ,
+   data=rdata, ratetable=slopop, int=5)
> rstrans(Surv(time, cens)~sex+as.factor(agegr) +
+   ratetable(age=age*365.24, sex=sex, year=year) ,
+   data=rdata, ratetable=slopop, int=5)
```

The outputs of all these functions are organized in the same way, the computed values (described in Section (3.3)) are stored in objects that closely match those of other standard regression model functions in R (e.g. `coxph` function). As an example of the most important part of the output (obtained with function `summary`), we report the estimates of the β and γ coefficients (see Eq. (9)), their standard errors and the results of the Wald test for each covariate (z and p values) for the Estève additive model (the default of the `rsadd` function):

```
output1.txt
Coefficients:
      Estimate Std. Error Pr(>|z|)
sex          0.9028   0.2165 3.05e-05 ***
agegr.54.61  0.1421   0.3165  0.6534
agegr.62.70  0.5355   0.2940  0.0685 .
agegr.71.95  0.6110   0.3142  0.0518 .
fu [0, 1]    -4.2187   0.3859 < 2e-16 ***
fu (1, 2]    -4.9838   0.4272 < 2e-16 ***
fu (2, 3]    -5.0188   0.4393 < 2e-16 ***
fu (3, 4]    -5.4598   0.5623 < 2e-16 ***
fu (4, 5]    -4.9762   0.4444 < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*'
                0.05 '.' 0.1 ' ' 1
```

We note that the program's output is presented here as it is, and one would normally change it following medical journals' requirements.

Covariate `agegr` has four possible values, the youngest age group is automatically taken as the reference group. The output therefore consists of nine coefficients, the last five representing the follow-up interval indicators. The positive sex coefficient ($\gamma = 0.903$) implies that the survival of men is relatively better. The age however does not seem to be a very important factor, with even the oldest group coefficient not differing significantly from the youngest ($p = 0.052$). This means that the differences in survival between the age groups that we would get from any classical survival method are nearly wholly attributable to the population risks. The coefficients for the follow-up years are similar as well, only the first year seems to have a larger hazard ($\exp(\text{fu}[0, 1]) = 0.015$ per year).

The estimates obtained by the three additive model fitting options are compared in Table 1. We can see that the results differ slightly but that the interpretations of the model results would be the same in all three cases. There will however always exist some differences as is clear from the theory in Section 2.

All the functions have been thoroughly checked and whenever possible compared to the outputs reported from other programs (e.g. the results in [6]). The details of the methods have been discussed with their authors and it is therefore our hope that the outputs of the functions are reliable.

5. To conclude

The usage of any statistical method depends mainly on the availability of adequate software, and relative survival tech-

niques are no exception. For now, the choice of the model, and the method to fit it, were mostly influenced by software at hand. Different methods were programmed in different languages, and their usage was hindered by considerable effort needed to learn the special requirements of such programs. We believe that by incorporating all the different methods into an R package, now being widely used by statisticians and other professionals, such obstacles are mostly overcome. All the programs use the standard R format for commands, and their usage is as easy as that of any other regression program in R. Users, experienced in relative survival analysis, and in programming in R, will find it easy to adjust the programs to their needs if necessary.

The package was written by the first author and is publicly available at CRAN [5].

Appendix A

The main advantage of using R (or S-plus) for relative survival lies in the way it handles the population mortality data. A special object class named `ratetable` is defined that is designed broadly enough to take into account any format particularity. Once the data are organized in the right format, the `survexp` function (`survival` package) is used to calculate survival probabilities over any period of time and any group of people in the same fashion regardless of the format of the population mortality data. A part of this function is also included in the package `reلسurv` and the same logic and syntax is preserved.

The US mortality tables are already included as part of the `survival` package; the Slovene population tables are included in package `reلسurv`. The Human Life-Table Database [12] contains a collection of population life tables covering a multitude of countries and many years. The function `transrate.hld` in the `reلسurv` package transforms data from their format into a `ratetable` object. When working with other formats, a simpler function `transrate` is provided to assist in the reorganization tasks.

The organisation of a `ratetable` object is described in detail in [13] or [14]. The construction of a `ratetable` object to be used in relative survival is straightforward. Firstly, the data must be organised as a rectangular array, which will usually mean three dimensions—sex, age and calendar year. All the elements in the array must be hazards in unit per day. If the population mortality data are not rectangular (a usual case will be that the rates will be given for women until an older age than for men), simply carry the last value forward as this is also done by all the functions that use the `ratetable` objects.

The rest of the work means specifying the attributes. The easiest way to do it is to check the attributes of an existing rate table and imitate them (as is done in the following example for an array we named `my.rate`). The compulsory attributes are the following:

- `dimid`. The names of the dimensions. As already seen in Section 3.2, the `dimid` names for the `slopop` `ratetable` are `age`, `year` and `sex`. If we have a four-dimensional array with dimensions `age`, `sex`, `year` and `race`, we can specify this attribute as:


```
> attributes(my.rate)$dimid <- c("age", "sex",
  "year", "race")
```
- `dim`. The dimensions of the array, for example


```
> attributes(my.rate)$dim <- c(100, 2, 4, 3)
```
- `dimnames`. The names of each of the groups, i.e. a vector of names for each of the `dimid` variables. The race names could be specified as (`race` is the fourth dimension of the array)


```
> attributes(my.rate)$dimnames[[4]]
  <- c("white", "black", "other")
```
- `factor`. This attribute specifies which of the variables are constant in time - the values for `sex` and `race` will therefore be equal to 1, the values for `age` and `calendar year` will be 0. Other values are allowed and are used in the US population tables, but special functions must of course be programmed when using those. In our example the value would be


```
> attributes(my.rate)$factor <- c(0, 0, 1, 1)
```
- `cutpoints`. A list of vectors denoting the cut points for each of the `dimid` variables (the constant ones like `sex` must have value `NULL`). If our population tables are estimated every ten years for four decades, vector of cut points will equal to (`calendar year` is the third dimension)


```
> attributes(my.rate)$cutpoints[[3]]
  <- as.date(c("1Jan70",
  "1Jan80", "1Jan90", "1Jan2000"))
```
- `class`. Set the class of the object to


```
> attributes(my.rate)$class <- "ratetable"
```

REFERENCES

- [1] F. Ederer, L.M. Axtell, S.J. Cutler, The relative survival rate: a statistical methodology, *Natl. Cancer Inst. Monogr.* 6 (1961) 101–121.
- [2] Available at: <http://www.cancerregistry.fi/surv3>.
- [3] Available at: <http://www.pauldickman.com>.
- [4] R. Giorgi, J. Payan, J. Gouvernet, *Rsurv*: a function to perform relative survival analysis with S-plus or R, *Comput. Meth. Progr. Biomed.* 78 (2005) 175–178.
- [5] R Development Core Team, R. A Language and Environment for Statistical Computing, R Foundation for Statistical Computing, Vienna, Austria, 2005. Available at: <http://www.R-project.org>.
- [6] P.W. Dickman, A. Sloggett, M. Hills, T. Hakulinen, Regression models for relative survival, *J. R. Stat. Soc. Ser. C* 23 (2003) 51–64.
- [7] J. Stare, R. Henderson, M. Pohar, An individual measure of relative survival, *J. R. Stat. Soc. Ser. C* 54 (2005) 115–126.
- [8] P.K. Andersen, K. Borch-Johnsen, T. Deckert, A. Green, P. Hougaard, N. Keiding, S. Kreiner, A Cox regression model for relative mortality and its application to diabetes mellitus survival data, *Biometrics* 41 (1985) 921–932.
- [9] T. Hakulinen, L. Tenkanen, Regression analysis of relative survival rates, *J. R. Stat. Soc. Ser. C* 36 (1987) 309–317.
- [10] J. Estève, E. Benhamou, M. Croasdale, M. Raymond, Relative survival and the estimation of net survival: elements for further discussion, *Stat. Med.* 9 (1990) 529–538.

- [11] J. Stare, M. Pohar, R. Henderson, Goodness of fit of relative survival models, *Stat. Med.* 24 (2005) 3911-3925.
- [12] Available at: <http://www.lifetable.de>.
- [13] Mathsoft S-Plus 2000, Guide to Statistics 2, Data Analysis Products Division of MathSoft, Seattle, WA, 1999.
- [14] T. Therneau, J. Offord, Expected Survival Based on Hazard Rates (Update), Technical Report, Section of Biostatistics, Mayo Clinic 63, 1999.