

Package ‘cuRe’

March 16, 2018

Type Package

Title Generalized parametric cure models and loss of lifetime

Version 0.1.0

Date 2017-12-01

Author Lasse Hjort Jakobsen

Maintainer Lasse Hjort Jakobsen <lahja@dcm.aau.dk>

Description The package contains functions for estimating generalized parametric mixture and non-mixture cure models, loss of lifetime, and mean residual lifetime, crude event probabilities.

URL <http://github.com/LasseHjort/cuRe>

BugReports <http://github.com/LasseHjort/cuRe/issues>

License GPL (>=2)

Encoding UTF-8

Depends R (>= 3.0.2), survival, rstpm2

Imports numDeriv, relsurv, rootSolve, statmod

LazyData yes

RoxygenNote 6.0.1

R topics documented:

calc.Crude	2
calc.Crude.quantile	4
calc.cure.quantile	5
calc.LL	6
calc.LL.quantile	8
colonDC	9
fit.cure.model	10
general.haz	12
GenFlexCureModel	13
lts	15
plot.crude	17
plot.cuRe	17
plot.le	18
plot.lts	19
predict.cm	19
predict.gfcm	20
survexp.dk	22

calc.Crude	<i>Crude event probabilities</i>
------------	----------------------------------

Description

Function for computing crude event probabilities based on relative survival models.

Usage

```
calc.Crude(object, newdata = NULL, type = c("cancer", "other", "condother"),
  time = NULL, tau = 100, reverse = FALSE, var.type = c("ci", "se",
    "n"), exp.fun = NULL, ratetable = survexp.dk, rmap, link = "loglog",
  n = 100)
```

Arguments

object	Fitted model to do predictions from. Possible classes are gfc, cm, stpm2, and pstpm2.
newdata	Data frame from which to compute predictions. If empty, predictions are made on the the data which the model was fitted on.
type	Probability to compute. Possible values are cancer (default), other, and condother (see details).
time	Optional time points at which to compute predictions. If empty, a grid of 100 time points between 0 and tau is selected.
tau	Upper bound of the cancer related death integral (see details). The argument is only used for type = condother. Default is 100.
reverse	Logical. If TRUE, 1 - probability is provided (default is FALSE). Only applicable for type = condother.
var.type	Character. Possible values are "ci" (default) for confidence intervals, "se" for standard errors, and "n" for neither.
exp.fun	Object of class list containing functions for the expected survival of each row in newdata. If not specified, the function computes the expected survival using the survival::survexp function and smoothing by smooth.spline.
ratetable	Object of class ratetable used to compute the general population survival. Default is survexp.dk.
rmap	List to be passed to survexp from the survival package if exp.fun = NULL. Detailed documentation on this argument can be found by ?survexp.
n	Number of knots used for the Gauss-Legendre quadrature.
Link	Link function for computing variance in order to bound confidence intervals. Default is loglog.

Details

The function estimates crude probabilities by using the relative survival, expected survival, and the cause-specific hazard function. The crude cumulative incidence of cancer related death (type = "cancer") is

$$P(T \leq t, D = cancer) = \int_0^t S^*(u)R(u)\lambda(u)du.$$

The crude cumulative incidence of death from other causes (type = "other") is

$$P(T \leq t, D = other) = \int_0^t S^*(u)R(u)h^*(u)du.$$

The conditional probability of eventually dying from other causes than cancer (type = "condother") is

$$P(D = other|T > t) = \frac{P(D = cancer) - P(T \leq t, D = cancer)}{P(T > t)}.$$

The proportion of patients bound to die from the disease ($P(D = cancer)$) can be computed by using type = "cancer" and choosing a sufficiently large time point (e.g., 100 years).

Value

A list containing the crude probability estimates of each individual in newdata.

References

Eloranta, S., et al. (2014) The application of cure models in the presence of competing risks: a tool for improved risk communication in population-based cancer patient survival. *Epidemiology*, 12:86.

Examples

```
##Use data cleaned version of the colon cancer data from the rstpm2 package
data("colonDC")

##Extract general population hazards
colonDC$bhaz <- general.haz(time = "FU", age = "agedays", sex = "sex", year = "dx",
                           data = colonDC, ratetable = survexp.dk)

##Spline-base cure model
#Fit cure model
fit <- stpm2(Surv(FUyear, status) ~ 1, data = colonDC, df = 6, bhazard = colonDC$bhaz, cure = T)

#Compute the probability of cancer related death
res <- calc.Crude(fit, time = seq(0, 20, length.out = 50),
                 rmap = list(age = agedays, sex = sex, year = dx))
plot(res)

#Compute the conditional probability of dying from other causes than cancer
res <- calc.Crude(fit, time = seq(0, 20, length.out = 50), type = "condother",
                 rmap = list(age = agedays, sex = sex, year = dx))
plot(res)

#Simple parametric cure model
#Fit cure model
fit <- fit.cure.model(Surv(FUyear, status) ~ 1, data = colonDC, bhazard = "bhaz",
```

```

type = "mixture", dist = "weibull", link = "logit")

#Compute the probability of cancer related death
res <- calc.Crude(fit, time = seq(0, 20, length.out = 50),
                 rmap = list(age = agedays, sex = sex, year = dx))
plot(res)

#Compute the conditional probability of cancer-related death
res <- calc.Crude(fit, time = seq(0, 20, length.out = 50), type = "condother",
                 rmap = list(age = agedays, sex = sex, year = dx), reverse = T)
plot(res)

```

calc.Crude.quantile	<i>Compute the time to statistical cure using the conditional probability of cancer-related death</i>
---------------------	---

Description

The following function estimates the time to statistical cure using the conditional probability of cancer-related death.

Usage

```

calc.Crude.quantile(fit, q = 0.05, newdata = NULL, max.time = 20,
  exp.fun = NULL, var.type = c("ci", "se", "n"), rmap,
  ratetable = survexp.dk, tau = 100, reverse = TRUE)

```

Arguments

fit	Fitted model to do predictions from. Possible classes are fmc, cm, stpm2, and pstpm2.
q	Threshold to estimate statistical cure according to.
newdata	Data frame from which to compute predictions. If empty, predictions are made on the the data which the model was fitted on.
max.time	Upper boundary of the interval [0, max.time] in which to search for solution.
exp.fun	Object of class list containing functions for the expected survival of each row in newdata. If not specified, the function computes the expected survival using the survival::survexp function and smoothing by smooth.spline.
var.type	Character. Possible values are "ci" (default) for confidence intervals, "se" for standard errors, and "n" for neither.
rmap	List to be passed to survexp from the survival package if exp.fun = NULL. Detailed documentation on this argument can be found by ?survexp.
ratetable	Object of class ratetable used to compute the general population survival. Default is survexp.dk
tau	Constant at which the bound to tie probability is calculated. Default is 100.
reverse	Logical passed on to calc.Crude. If TRUE (default), 1 - probability is provided. Only applicable for type = condother.

Value

The estimated cure point.

Examples

```
##Use data cleaned version of the colon cancer data from the rstpm2 package
data("colonDC")

##Extract general population hazards
colonDC$bhaz <- general.haz(time = "FU", age = "agedays", sex = "sex", year = "dx",
                           data = colonDC, ratetable = survexp.dk)

#Fit cure model and estimate cure point
fit <- stpm2(Surv(FUyear, status) ~ 1, data = colonDC, df = 6, bhazard = colonDC$bhaz, cure = T)
calc.Crude.quantile(fit, q = 0.05, rmap = list(age = agedays, sex = sex, year = dx))
```

calc.cure.quantile	<i>Compute the time to statistical cure using the conditional probability of cure</i>
--------------------	---

Description

The following function estimates the time to statistical cure using the conditional probability of cure.

Usage

```
calc.cure.quantile(fit, q = 0.05, newdata = NULL, max.time = 20,
                  var.type = c("ci", "se", "n"), reverse = TRUE)
```

Arguments

fit	Fitted model to do predictions from. Possible classes are gfcM, cm, stpm2, and pstpm2.
q	Threshold to estimate statistical cure according to.
newdata	Data frame from which to compute predictions. If empty, predictions are made on the the data which the model was fitted on.
max.time	Upper boundary of the interval [0, max.time] in which to search for solution.
var.type	Character. Possible values are "ci" (default) for confidence intervals, "se" for standard errors, and "n" for neither.
reverse	Logical. Whether to use 1 - prob (default) or prob as measure.

Value

The estimated cure point.

Examples

```
##Use data cleaned version of the colon cancer data from the rstpm2 package
data("colonDC")

##Extract general population hazards
colonDC$bhaz <- general.haz(time = "FU", age = "agedays", sex = "sex", year = "dx",
                           data = colonDC, ratetable = survexp.dk)

#Fit cure model and estimate cure point
fit <- GenFlexCureModel(Surv(FUyear, status) ~ 1, data = colonDC, df = 5, bhazard = "bhaz")
calc.cure.quantile(fit, q = 0.05)
```

calc.LL

Loss of lifetime estimation

Description

Function for computing loss of lifetime function based on relative survival models.

Usage

```
calc.LL(object, newdata = NULL, time = NULL, type = c("l1", "mr1"),
        tau = 100, var.type = c("ci", "se", "n"), exp.fun = NULL,
        ratetable = survexp.dk, rmap, pars = NULL, n = 100)
```

Arguments

object	Fitted model to do predictions from. Possible classes are gfcM, cm, stpm2, and pstpm2.
newdata	Data frame from which to compute predictions. If empty, predictions are made on the the data which the model was fitted on.
time	Optional time points at which to compute predictions. If empty, a grid of 100 time points between 0 and tau is selected.
type	Type of life expectation estimate. Possible values are l1 (default) which gives the loss of lifetime, and mr1 which gives the mean residual lifetime.
tau	The upper limit of the integral. Default is 100.
var.type	Character. Possible values are "ci" (default) for confidence intervals, "se" for standard errors, and "n" for neither.
exp.fun	Object of class list containing functions for the expected survival of each row in newdata. If not specified, the function computes the expected survival using the survival::survexp function and smoothing by smooth.spline.
ratetable	Object of class ratetable used to compute the general population survival. Default is survexp.dk.
rmap	List to be passed to survexp from the survival package if exp.fun = NULL. Detailed documentation on this argument can be found by ?survexp.
n	Number of knots used for the Gauss-Legendre quadrature.

Details

If type = "ll", the function computes

$$\frac{\int_t^\infty S^*(u)}{S^*(t)} - \frac{\int_t^\infty S(u)}{S(t)}.$$

If type = "mrl", the function computes

$$\frac{\int_t^\infty S(u)}{S(t)},$$

for a given t. The function $S^*(t)$ is the general population survival function and $S(t)$ is the patient survival function. The integral is computed by Gauss-Legendre quadrature and the point wise variance is estimated using the delta method and numerical differentiation.

Value

An object of class `le` containing the life expectancy estimates of each individual in `newdata`.

Examples

```
##Use data cleaned version of the colon cancer data from the rstpm2 package
data("colonDC")

##Extract general population hazards
colonDC$bhaz <- general.haz(time = "FU", age = "agedays", sex = "sex", year = "dx",
                           data = colonDC, ratetable = survexp.dk)

##Spline-base cure model
#Fit cure model
fit <- stpm2(Surv(FUyear, status) ~ 1, data = colonDC, df = 6, bhazard = colonDC$bhaz, cure = T)

#Compute and plot the loss of lifetime function
res <- calc.LL(fit, time = seq(0, 20, length.out = 50),
              rmap = list(age = agedays, sex = sex, year = dx))
plot(res)

#Compute and plot the mean residual lifetime
res <- calc.LL(fit, time = seq(0, 20, length.out = 50), type = "mrl",
              rmap = list(age = agedays, sex = sex, year = dx))
plot(res)

#Simple parametric cure model
#Fit cure model
fit <- fit.cure.model(Surv(FUyear, status) ~ 1, data = colonDC, bhazard = "bhaz",
                    type = "mixture", dist = "weibull", link = "logit")

#Compute and plot the loss of lifetime function
res <- calc.LL(fit, time = seq(0, 20, length.out = 50),
              rmap = list(age = agedays, sex = sex, year = dx))
plot(res)

#Compute and plot the mean residual lifetime
res <- calc.LL(fit, time = seq(0, 20, length.out = 50), type = "mrl",
              rmap = list(age = agedays, sex = sex, year = dx))
```

```
plot(res)
```

calc.LL.quantile	<i>Compute the time to statistical cure using the loss of lifetime function</i>
------------------	---

Description

The following function estimates the time to statistical cure using the loss of lifetime function.

Usage

```
calc.LL.quantile(fit, q = 1, newdata = NULL, max.time = 20,
  var.type = c("ci", "se", "n"), exp.fun = NULL, rmap = NULL,
  ratetable = survexp.dk, tau = 100, type = "ll")
```

Arguments

fit	Fitted model to do predictions from. Possible classes are gfc, cm, stpm2, and pstpm2.
q	Threshold to estimate statistical cure according to.
newdata	Data frame from which to compute predictions. If empty, predictions are made on the the data which the model was fitted on.
max.time	Upper boundary of the interval [0, max.time] in which to search for solution.
var.type	Character. Possible values are "ci" (default) for confidence intervals, "se" for standard errors, and "n" for neither.
exp.fun	Object of class list containing functions for the expected survival of each row in newdata. If not specified, the function computes the expected survival using the survival::survexp function and smoothing by smooth.spline.
rmap	List to be passed to survexp from the survival package if exp.fun = NULL. Detailed documentation on this argument can be found by ?survexp.
ratetable	Object of class ratetable used to compute the general population survival. Default is survexp.dk
tau	The upper limit of the integral. Default is 100.
type	Type of life expectancy measure. Possible values are "ll" for the loss of lifetime and "mrl" for the mean residual lifetime.

Value

The estimated cure point.

Examples

```
##Use data cleaned version of the colon cancer data from the rstpm2 package
data("colonDC")

##Extract general population hazards
colonDC$bhaz <- general.haz(time = "FU", age = "agedays", sex = "sex", year = "dx",
  data = colonDC, ratetable = survexp.dk)
```



```
#Fit cure model and estimate cure point
fit <- stpm2(Surv(FUyear, status) ~ 1, data = colonDC, df = 6, bhazard = colonDC$bhaz, cure = T)
calc.LL.quantile(fit, q = 1, rmap = list(age = agedays, sex = sex, year = dx))
```

colonDC

Colon cancer data

Description

A dataset containing the individual baseline and follow-up data on more than 15,000 colon cancer patient. The data is a data cleaned version of the `rstpm2::colon` data set.

Usage

```
data(colonDC)
```

Format

A data frame with 15564 rows and 11 variables

Details

- sex. Either female or male
- age. Diagnostic age
- stage. Clinical stage at diagnosis (either unknown, localised, regional, or distant)
- statusDC. Alive indicator with cause of death
- subsite. Anatomical subsite of tumour (either coecum and ascending, transverse, descending and sigmoid, or other and NOS)
- dx. Date of diagnosis
- exit. Date of study exit
- status. Alive indicator (0 = alive, 1 = dead)
- FU. Follow-up time measured in days
- FUyear. Follow-up time measured in years
- agedays. Diagnostic age in days.

fit.cure.model	<i>Parametric cure model</i>
----------------	------------------------------

Description

This function is used to fit parametric cure models on the relative survival.

Usage

```
fit.cure.model(formula, data, bhazard = NULL, formula.surv = NULL,
  type = c("mixture", "nmixture"), dist = c("weibull", "exponential",
    "lognormal", "weiwei", "weexp"), link = c("logit", "loglog", "identity",
    "probit"), covariance = TRUE, link.mix = c("logit", "loglog", "identity",
    "probit"), control = list(maxit = 10000), method = "Nelder-Mead",
  init = NULL)
```

Arguments

formula	Formula for modelling the cure proportion. Reponse has to be of the form <code>Surv(time, status)</code> .
data	Data frame in which to interpret the variable names in <code>formula</code> and <code>formula.surv</code> .
bhazard	Background hazard.
formula.surv	List of formulas for each parameter in the parametric distribution (see details).
type	A character indicating the type of cure model. Possible values are <code>mixture</code> (default) and <code>nmixture</code> .
dist	The parametric distribution of the survival of the uncured.
link	Character. Specifies the link function of the cure proportion.
covariance	Logical. If <code>TRUE</code> (default), the covariance matrix is computed.
link.mix	Character. Specifies the link function for the mixture parameter in a weibull-weibull mixture model and weibull-exponential model. Only used when <code>dist = "weiwei"</code> and <code>dist = "weexp"</code> .
control	List of control parameters passed to <code>optim</code> .
method	Optimization method passed to <code>optim</code> .
init	Initial values for the maximum likelihood optimization. If not provided, the optimization will start in 0.

Details

The function fits the model,

$$S(t) = \pi + (1 - \pi)S_u(t).$$

The `formula.surv` argument is used to model $S_u(t)$. It is a list of formulas with as many entries as there are parameters in the chosen parametric distribution. If not specified, all formulas are assumed to be ~ 1 . The i th formula, i.e., `formula.surv[[i]]` refers θ_i in the below survival functions.

Exponential model:

$$S_u(t) = \exp(-t\theta_1).$$

Weibull model:

$$S_u(t) = \exp(-\theta_1 t^{\theta_2}).$$

Log-normal model:

$$S_u(t) = 1 - \Phi\left(\frac{\log(t) - \theta_1}{\theta_2}\right)$$

Weibull-exponential mixture model:

$$S_u(t) = \theta_1 \exp(-\theta_2 t^{\theta_3}) + (1 - \theta_1) \exp(-\theta_4 t).$$

Weibull-Weibull mixture model:

$$S_u(t) = \theta_1 \exp(-\theta_2 t^{\theta_3}) + (1 - \theta_1) \exp(-\theta_4 t^{\theta_5}).$$

In the the mixture models, the link function for the mixture component is controlled by `link.mix`. The remaining parameters are modelled using an exponential link function except θ_1 in the log-normal model, which is modeled using the identity.

Value

An object of class `cm` containing the parameters of the cure model.

Examples

```
##Use data cleaned version of the colon cancer data from the rstpm2 package
data("colonDC")

##Extract general population hazards
colonDC$bhaz <- general.haz(time = "FU", age = "agedays", sex = "sex", year = "dx",
                           data = colonDC, ratetable = survexp.dk)

###Without covariates
##Fit weibull mixture cure model
fit.wei <- fit.cure.model(Surv(FUyear, status) ~ 1, data = colonDC, bhazard = "bhaz",
                        type = "mixture", dist = "weibull", link = "logit")

##Plot various summaries of the model
plot(fit.wei)
plot(fit.wei, time = seq(0, 40, length.out = 100))
plot(fit.wei, type = "hazard")
plot(fit.wei, type = "survuncured")
plot(fit.wei, type = "probcure")

#Fit a weibull-weibull mixture cure model
fit.weiwei <- fit.cure.model(Surv(FUyear, status) ~ 1, data = colonDC, bhazard = "bhaz",
                          type = "mixture", dist = "weiwei", link = "logit")

#Compare to the weibull model
plot(fit.wei, var.type = "n")
plot(fit.weiwei, add = T, col = 2, var.type = "n")

###With covariates
##Fit weibull mixture cure model
fit <- fit.cure.model(Surv(FUyear, status) ~ age, data = colonDC, bhazard = "bhaz",
                    formula.surv = list(~ age, ~1),
                    type = "mixture", dist = "weibull", link = "logit")
```

```
##Plot model
plot(fit, newdata = data.frame(age = 60),
      time = seq(0, 15, length.out = 100), var.type = "n")
plot(fit, newdata = data.frame(age = 50),
      time = seq(0, 15, length.out = 100), var.type = "n", add = TRUE, col = 2)

plot(fit, newdata = data.frame(age = 60),
      time = seq(0, 15, length.out = 100), var.type = "n", type = "hazard")
plot(fit, newdata = data.frame(age = 50),
      time = seq(0, 15, length.out = 100), var.type = "n", type = "hazard", add = T, col = 2)
```

general.haz

Extract general population hazard

Description

Yearly general population hazards matched on age, gender, and calendar year is extracted from a ratetable.

Usage

```
general.haz(time, age, sex, year, data = NULL, ratetable = survexp.dk)
```

Arguments

time	Either a numeric vector of follow-up times or a character indicating the variable containing the follow-up times in the data.
age	Either a numeric vector of ages or a character indicating the variable containing the patient ages in the data.
sex	Either a character vector or factor with the sex of each patient or a character indicating the variable containing the patient sex in the data.
year	Either a vector of class Date with the calendar time points or a character indicating the variable containing the calendar times in the data.
data	The data from which to extract variables from. If time, age, sex, or year are not characters, this will not be used.
ratetable	Object of class ratetable to extract background hazards from. Defaults to survexp.dk.

Value

An object of class numeric containing the yearly expected hazards.

Examples

```
bhaz1 <- general.haz(time = "FU", age = "agedays", sex = "sex", year = "dx",
                     data = colonDC, ratetable = survexp.dk)

bhaz2 <- general.haz(time = colonDC$FU, age = colonDC$agedays, sex = colonDC$sex,
                     year = colonDC$dx, ratetable = survexp.dk)
all(bhaz2 == bhaz1)
```

GenFlexCureModel

*Fit generalized mixture cure model***Description**

The following function fits a generalized mixture or non-mixture cure model using a link function for the cure rate and for the survival of the uncured, i.e.,

$$S(t|z) = \pi(z) + [1 - \pi(z)]S_u(t|z),$$

where

$$g_1[S_u(t|z)] = \eta_1(t, z) \quad \text{and} \quad g_2[\pi(z)] = \eta_2(z).$$

The function implements a range of link functions for both g_1 and g_2 and allows the linear predictors to be specified in any way.

Usage

```
GenFlexCureModel(formula, data, smooth.formula = NULL, smooth.args = NULL,
  df = 3, tvc = NULL, tvc.formula = NULL, bhazard = NULL,
  cr.formula = ~1, type = "mixture", link.type.cr = c("logit", "loglog",
    "identity", "probit"), link.type = c("PH", "P0", "probit"), init = NULL,
  timeVar = "", covariance = T, verbose = T, control = list(maxit =
    10000), method = "Nelder-Mead", constraint = TRUE, ini.types = c("cure",
    "flexpara"))
```

Arguments

formula	Formula for modelling the survival of the uncured. A linear term for time-varying coefficients is required here. Reponse has to be of the form <code>Surv(time, status)</code> .
data	Data frame in which to interpret the variables names in <code>formula</code> , <code>smooth.formula</code> , and <code>cr.formula</code> .
smooth.formula	Formula for describing the time-effect of the survival of the uncured (default is <code>NULL</code>).
smooth.args	List. Optional arguments to the time-effect of the survival of the uncured (default is <code>NULL</code>).
df	Integer. Degrees of freedom (default is 3) for the time-effect of the survival of the uncured.
tvc	Named list of integers. Specifies the degrees of freedom for a time-varying covariate effect. For instance, <code>tvc = list(a = 3)</code> creates a time-varying spline-effect of the covariate "a" with 3 degrees of freedom using the <code>rstpm2::nsx</code> function.
tvc.formula	Formula for the time-varying covariate effects. For time-varying effects, a linear term of the covariate has to be included in <code>formula</code> .
bhazard	Background hazard.
cr.formula	Formula for the cure proportion. The left hand side of the formula is not used and should therefore not be specified.
type	A character indicating the type of cure model. Possible values are <code>mixture</code> (default) and <code>nmixture</code> .

<code>link.type.cr</code>	Character providing the link function for the cure proportion. Possible values are <code>logit</code> (default), <code>loglog</code> , <code>identity</code> , and <code>probit</code> .
<code>link.type</code>	Character providing the link function for the survival of the uncured. Possible values are <code>PH</code> (default), <code>P0</code> , and <code>probit</code> .
<code>init</code>	Providing initial values for the optimization procedure. If not specified, the function will create initial values internally.
<code>timeVar</code>	Optional character giving the name of the variable specifying the time component of the <code>Surv</code> object. Should currently not be used.
<code>covariance</code>	Logical. If <code>TRUE</code> (default), the covariance matrix is computed.
<code>verbose</code>	Logical. If <code>TRUE</code> status messages of the function is outputted.
<code>control</code>	Named list with control arguments passed to <code>optim</code> .
<code>method</code>	Character passed to <code>optim</code> indicating the method for optimization. See <code>?optim</code> for details.
<code>constraint</code>	Logical. Indicates whether non-negativity constraints should be forced upon the hazard of the uncured patients
<code>ini.types</code>	Character vector denoting the executed schemes for computing initial values (see details).

Details

The default smoother is natural cubic splines established by the `rstpm2::nsx` function. Functions such as `ns`, `bs` are readily available for usage. Also the basis function of `flexsurv` works. Initial values are calculated by two procedures and the model is fitted under each set of initial values. The model producing the highest likelihood is selected.

Value

An object of class `gfc`.

Examples

```
##Use data cleaned version of the colon cancer data from the rstpm2 package
data("colonDC")

##Extract general population hazards
colonDC$bhaz <- general.haz(time = "FU", age = "agedays", sex = "sex", year = "dx",
                           data = colonDC, ratetable = survexp.dk)

###Without covariates
##Fit mixture cure model
fit <- GenFlexCureModel(Surv(FUyear, status) ~ 1, data = colonDC, df = 4, bhazard = "bhaz")

##Plot model
plot(fit)
plot(fit, time = seq(0.001, 40, length.out = 100))
plot(fit, type = "hazard")
plot(fit, type = "survuncured")
plot(fit, type = "probcure")

##Predict cure proportion
predict(fit, type = "curerate")
```

```

##Fit non-mixture cure model
fit <- GenFlexCureModel(Surv(FUyear, status) ~ 1, data = colonDC, df = 4,
                        bhazard = "bhaz", type = "nmixture")

##Plot relative survival
plot(fit)

##Predict cure proportion
predict(fit, type = "curerate")

###With covariates
##Fit mixture cure model
fit <- GenFlexCureModel(Surv(FUyear, status) ~ sex, data = colonDC, df = 4, bhazard = "bhaz", cr.formula = ~ sex)

##Plot model
plot(fit, newdata = data.frame(sex = factor("female", levels = c("male", "female"))),
     time = seq(0.001, 15, length.out = 100), ci = "n")
plot(fit, newdata = data.frame(sex = factor("male", levels = c("male", "female"))),
     time = seq(0.001, 15, length.out = 100), col = 2, ci = "n", add = T)

plot(fit, newdata = data.frame(sex = factor("female", levels = c("male", "female"))),
     time = seq(0.001, 15, length.out = 100), ci = "n", type = "survuncured")
plot(fit, newdata = data.frame(sex = factor("male", levels = c("male", "female"))),
     time = seq(0.001, 15, length.out = 100), col = 2, ci = "n", add = T, type = "survuncured")

predict(fit, type = "curerate", data.frame(sex = factor("female", levels = c("male", "female"))))

##Fit mixture cure model with time-varying covariates
colonDC$gender <- as.numeric(colonDC$sex) - 1
fit <- GenFlexCureModel(Surv(FUyear, status) ~ gender, data = colonDC, df = 4, bhazard = "bhaz",
                        cr.formula = ~ gender, tvc = list(gender = 2))

##Plot model
plot(fit, newdata = data.frame(gender = 0))
plot(fit, newdata = data.frame(gender = 1), add = T, col = 2)

plot(fit, type = "hazard", newdata = data.frame(gender = 0), ci = "n")
plot(fit, type = "hazard", newdata = data.frame(gender = 1), add = T, col = 2, ci = "n")

#Predict cure proportions for a male and female patients
predict(fit, type = "curerate", newdata = data.frame(gender = 0))
predict(fit, type = "curerate", newdata = data.frame(gender = 1))

```

Description

Function for computing survival estimates using a relative survival model and the expected background survival.

Usage

```
lts(fit, type = c("surv", "hazard", "cumhaz", "loghaz"), newdata = NULL,
    time = NULL, var.type = c("ci", "se", "n"), exp.fun = NULL,
    ratetable = survexp.dk, rmap)
```

Arguments

<code>fit</code>	Fitted model to do predictions from. Possible classes are <code>fcm</code> , <code>gfc</code> , <code>stpm2</code> , <code>pstpm2</code> , and <code>cm</code> .
<code>newdata</code>	Data frame from which to compute predictions. If empty, predictions are made on the the data which the model was fitted on.
<code>time</code>	Optional time points at which to compute predictions. If empty, a grid of 100 time points between 0 and the maximum follow-up time is selected.
<code>var.type</code>	Character. Possible values are "ci" (default) for confidence intervals, "se" for standard errors, and "n" for neither.
<code>exp.fun</code>	Object of class <code>list</code> containing functions for the expected survival of each row in <code>newdata</code> . If not specified, the function computes the expected survival using the <code>survival::survexp</code> function and smoothing by <code>smooth.spline</code> .
<code>ratetable</code>	Object of class <code>ratetable</code> used to compute the general population survival. Default is <code>survexp.dk</code> .
<code>rmap</code>	List to be passed to <code>survexp</code> from the <code>survival</code> package if <code>exp.fun = NULL</code> . Detailed documentation on this argument can be found by <code>?survexp</code> .

Value

A object of class `lts` containing the loss of lifetime estiamtes of each individual in `newdata`.

Examples

```
##Use data cleaned version of the colon cancer data from the rstpm2 package
data("colonDC")

##Extract general population hazards
colonDC$bhaz <- general.haz(time = "FU", age = "agedays", sex = "sex", year = "dx",
                           data = colonDC, ratetable = survexp.dk)

##Fit flexible parametric relative survival model
fit <- stpm2(Surv(FUyear, status) ~ 1, data = colonDC, df = 6, bhazard = colonDC$bhaz)

##Compute survival probabilities from 0 to 20 years
pred <- lts(fit, rmap = list(age = agedays, sex = sex, year = dx))

##Plot the survival function
plot(pred)
```

plot.crude	<i>Plot crude event probabilities</i>
------------	---------------------------------------

Description

Plot function for the computed crude event probabilities.

Usage

```
plot.crude(obj, ylim = c(0, 1), xlim = NULL, ci = T, col = 1,
  ylab = NULL, xlab = "Time", add = F, ...)
```

Arguments

obj	Object of class crude in which crude probabilities are calculated.
ylim	Limits of y-axis.
xlim	Limits of x-axis.
ci	Logical denoting whether confidence intervals should be plotted (default is TRUE).
col	Colour of each curve.
ylab	Label of the y-axis. If NULL, the function uses its default labels depending on obj\$type.
xlab	Label of the x-axis (default is "Time").
add	Logical indicating whether the curves should be added to the current plot window (default is FALSE).
...	Further arguments passed to plot and lines.

plot.cuRe	<i>Plot function for Flexible mixture cure model</i>
-----------	--

Description

Plot function associated with the classes gfcM and cm

Usage

```
plot.cuRe(object, newdata = NULL, type = c("surv", "probcure",
  "survuncured", "hazarduncured", "cumhazuncured", "densityuncured",
  "failuncured", "oddsuncured", "loghazarduncured", "hazard", "density", "fail",
  "loghazard", "odds", "cumhaz"), time = NULL, xlim = NULL, ylim = c(0,
  1), xlab = "Time", ylab = NULL, col = 1, ci = T, add = F)
```

Arguments

newdata	Data frame from which to compute predictions. If empty, predictions are made on the the data which the model was fitted on.
type	Character. Defines the desired scale to plot. See ?predict.gfcm to see explanation of possible values.
time	Optional time points at which to compute predictions. This argument is not used if type is curerate.
xlim	Limits of the x-axis
ylim	Limits of the y-axis.
xlab	Label of the x-axis. Default is "Time".
ylab	Label of the y-axis. If NULL, the function uses its default y-labels depending on fit\$type and fit\$excess.
col	Colour of each line.
ci	Logical. If TRUE, confidence intervals are added to the plot.
add	Logical. If TRUE the curve is added to the existing plot. #' @return A plot containing the predictions of each observation in newdata.
fit	Object of class cuRe.

plot.le

*Plot the loss of lifetime function***Description**

Plot function for the class le

Usage

```
plot.le(obj, ylim = NULL, xlim = NULL, ci = T, col = 1, ylab = NULL,
        xlab = "Time", add = F, ...)
```

Arguments

obj	Object of class le.
ylim	Limit of the y-axis.
xlim	Limit of x-axis.
ci	Logical. If TRUE, confidence intervals are added to the plot.
col	Numeric or character indicating the colours of the curves.
ylab	Label to be written on the y-axis.
xlab	Label to be written on the x-axis.
add	Logical. If TRUE, the curve is added to the current plot window.
...	Further argument passed to plot and lines.

plot.lts	<i>Plot the long term survival</i>
----------	------------------------------------

Description

Plot function for the class lts.

Usage

```
plot.lts(obj, ylim = NULL, xlim = NULL, ci = F, col = 1, ylab = NULL,
         xlab = "Time", add = F, ...)
```

Arguments

obj	Object of class lts.
ylim	Limit of the y-axis. If empty, these are chosen according to the extremes in obj.
xlim	Limit of x-axis. If empty, these are chosen according to the time extremes of obj.
ci	Logical indicating wether confidence intervals should be included in the plot (default is TRUE).
col	Numeric or character indicating the colours of the curves.
ylab	Label to be written on the y-axis.
xlab	Label to be written on the x-axis.
add	Logical indicating whether to add to current plot window (default is FALSE).
...	Further argument passed to plot and lines.

predict.cm	<i>Predict function for cure models</i>
------------	---

Description

This function is used to make predictions of the cure models.

Usage

```
predict.cm(fit, newdata = NULL, type = c("surv", "curerate", "probcure",
    "survuncured", "hazarduncured", "cumhazuncured", "densityuncured",
    "failuncured", "oddsuncured", "loghazarduncured", "hazard", "density", "fail",
    "loghazard", "odds", "cumhaz"), time = NULL, var.type = c("ci", "se",
    "n"), pars = NULL, link = NULL, keep.attributes = F)
```

Arguments

fit	Object of class cm to do predictions from.
newdata	Data frame from which to compute predictions. If empty, predictions are made on the data which the model was fitted on.
type	Prediction type (see details). The default is surv.
time	Optional time points at which to compute predictions. This argument is not used if type is curerate.
var.type	Character. Possible values are "ci" (default) for confidence intervals, "se" for standard errors, and "n" for neither.
pars	Numerical vector containing the parameters values of the model. In general, this argument can be ignored by the user.
link	Character, indicating the link function for the variance calculations. Possible values are "log", "cloglog", "log2", and "I". If NULL (default), the function will determine link from type.
keep.attributes	Logical. If TRUE, newdata will be added to the attributes of the output.

Details

Possible values for argument type are:

surv: Survival function

curerate: The cure rate

probcurate: The conditional probability of being cured

survuncured: The survival of the uncured

hazarduncured: The hazard function of the uncured

cumhazuncured: The cumulative hazard of the uncured

densityuncured: The density function of the uncured

failuncured: The distribution function of the uncured, i.e., 1 - survuncured

oddsuncured: Odds of the uncured, i.e., (1 - survuncured) / survuncured

loghazarduncured: The log-hazard of the uncured

hazard: The hazard function

density: The density function

fail: The distribution function

loghazard: The log-hazard function

odds: The odds, i.e., (1 - surv) / surv

cumhaz: The cumulative hazard function

Value

A list containing the predictions of each individual in newdata.

predict.gfcm

Predict function for flexible mixture cure model

Description

Function for doing predictions for class gfcm.

Usage

```
predict.gfcm(object, newdata = NULL, type = c("surv", "curerate",
  "probcure", "survuncured", "hazarduncured", "cumhazuncured", "densityuncured",
  "failuncured", "oddsuncured", "loghazarduncured", "hazard", "density", "fail",
  "loghazard", "odds", "cumhaz"), indi = TRUE, time = NULL,
  var.type = c("ci", "se", "n"), pars = NULL, link = NULL,
  keep.attributes = FALSE, ...)
```

Arguments

<code>newdata</code>	Data frame from which to compute predictions. If empty, predictions are made on the data which the model was fitted on.
<code>type</code>	Prediction type (see details). The default is <code>surv</code> .
<code>time</code>	Optional time points at which to compute predictions. This argument is not used if <code>type</code> is <code>curerate</code> .
<code>var.type</code>	Character. Possible values are "ci" (default) for confidence intervals, "se" for standard errors, and "n" for neither.
<code>pars</code>	Numerical vector containing the parameters values of the model. In general, this argument can be ignored by the user.
<code>link</code>	Character, indicating the link function for the variance calculations. Possible values are "log", "cloglog", "log2", and "I". If NULL (default), the function will determine link from <code>type</code> .
<code>keep.attributes</code>	Logical. If TRUE, <code>newdata</code> will be added to the attributes of the output.
<code>fit</code>	Object of class <code>gfcm</code> to do predictions from.

Details

Possible values for argument `type` are:

`surv`: Survival function

`curerate`: The cure rate

`probcure`: The conditional probability of being cured

`survuncured`: The survival of the uncured

`hazarduncured`: The hazard function of the uncured

`cumhazuncured`: The cumulative hazard of the uncured

`densityuncured`: The density function of the uncured

`failuncured`: The distribution function of the uncured, i.e., $1 - \text{survuncured}$

`oddsuncured`: Odds of the uncured, i.e., $(1 - \text{survuncured}) / \text{survuncured}$

`loghazarduncured`: The log-hazard of the uncured

`hazard`: The hazard function

`density`: The density function

`fail`: The distribution function

`loghazard`: The log-hazard function

`odds`: The odds, i.e., $(1 - \text{surv}) / \text{surv}$

`cumhaz`: The cumulative hazard function

Value

A list containing the predictions of each individual in `newdata`.

`survexp.dk`*Ratetable of the Danish general population*

Description

Object of class `ratetable` containing the daily hazards in the Danish general population as reported by the Human Mortality Database (www.mortality.org).

Usage`survexp.dk`**Format**

An object of class `ratetable` of dimension 111 x 180 x 2.

Details

The `ratetable` was generated by using the `relsurv::transrate.hmd` function. The data were downloaded on 15-09-2017 separately for male and female Danish citizens.

The data can be accessed through:

Female: http://www.mortality.org/hmd/DNK/STATS/ftper_1x1.txt

Male: http://www.mortality.org/hmd/DNK/STATS/mltper_1x1.txt

Index

*Topic **datasets**

colonDC, [9](#)

survexp.dk, [22](#)

calc.Crude, [2](#)

calc.Crude.quantile, [4](#)

calc.cure.quantile, [5](#)

calc.LL, [6](#)

calc.LL.quantile, [8](#)

colonDC, [9](#)

fit.cure.model, [10](#)

general.haz, [12](#)

GenFlexCureModel, [13](#)

lts, [15](#)

plot.crude, [17](#)

plot.cuRe, [17](#)

plot.le, [18](#)

plot.lts, [19](#)

predict.cm, [19](#)

predict.gfcm, [20](#)

survexp.dk, [22](#)