Package ‘cuRe’

January 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` *Crude event probabilities*

Description

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

Usage

```
calc.Crude(object, newdata = NULL, type = c("cancer", "other", "othertime"),
  time = NULL, last.point = 100, reverse = FALSE, ci = T,
  expected = NULL, ratetable = survexp.dk, rmap, link = "loglog",
  n = 100)
```

Arguments

`object` Fitted model to do predictions from. Possible classes are `fmc`, `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 `othertime` (see details).

`time` Optional time points at which to compute predictions. If empty, a grid of 100 time points between 0 and `last.point` is selected.

`last.point` Upper bound of the cancer related death integral (see details). The argument is only used for `type = othertime`. Default is 100.

`reverse` Logical. If `TRUE`, 1 - probability is provided (default is `FALSE`). Only applicable for `type = othertime`.

`ci` Logical. If `TRUE` (default), confidence intervals are computed.

`expected` Object of class `list` containing objects of class `survexp`, with the expected survival of each row in `newdata`. If not specified, the function computes the expected survival.

`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 `expected = NULL`. Detailed documentation on this argument can be found by `?survexp`.

`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 functions. The crude cumulative incidence of cancer related death (`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 (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 (othertime)

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

The proportion of patients bound to die from the disease (P(D = cancer)) can be computed by using cancer and choosing a sufficiently large time point (default is 100).

Value

An object of class crude 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)

#Fit flexible parametric cure model
fit <- stpm2(Surv(FUyear, status2) ~ 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 = 100),
                  rmap = list(age = agedays, sex = sex, year = dx))
plot(res)

#Compute the probability of eventually dying from other causes than cancer
res <- calc.Crude(fit, time = seq(0, 20, length.out = 100), type = "othertime",
                  rmap = list(age = agedays, sex = sex, year = dx))
plot(res)


#Fit parametric cure model
fit <- fit.cure.model(Surv(FUyear, status2) ~ 1, data = colonDC, bhazard = "bhaz",
                      formula.k1 = ~ 1, formula.k2 = ~ 1,
                      type = "mixture", dist = "weibull", link = "logit")

#Compute the probability of cancer related death
res <- calc.Crude(fit, time = seq(0, 20, length.out = 100),
                  rmap = list(age = agedays, sex = sex, year = dx))
plot(res)

#Compute the probability of eventually dying from other causes than cancer
```

```
res <- calc.Crude(fit, time = seq(0, 20, length.out = 100), type = "othertime",
                  rmap = list(age = agedays, sex = sex, year = dx))
plot(res)
```

calc.Crude.quantile *Compute the time to statistical cure using a probability measure*

Description

The following function estimates the time to statistical cure using the probability of eventually dying from cancer.

Usage

```
calc.Crude.quantile(fit, q = 0.95, newdata = NULL, max.time = 20,
  expected = NULL, ci = TRUE, rmap, ratetable = survexp.dk,
  last.point = 100, reverse = FALSE)
```

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.
expected	Object of class list containing objects of class survexp, with the expected survival of each row in newdata. If not specified, the function computes the expected survival.
ci	Logical. If TRUE (default), confidence intervals are computed.
rmap	List to be passed to survexp from the survival package. 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
last.point	Constant at which the bound to tie probability is calculated. Default is 100.
reverse	Passed to the calc.crude function.
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 points.

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("ll", "mrl"),
  tau = 100, ci = T, expected = NULL, ratetable = survexp.dk, rmap,
  pars = NULL, n = 100)
```

Arguments

`object`	Fitted model to do predictions from. Possible classes are `fmc`, `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 `ll` (default) which gives the loss of lifetime, and `mrl` which gives the mean residual lifetime.
`tau`	The upper limit of the integral. Default is 100.
`ci`	Logical. If `TRUE` (default), confidence intervals are computed.
`expected`	Object of class `list` containing objects of class `survexp`, with the expected survival of each row in newdata. If not specified, the function computes the expected survival.
`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. Detailed documentation on this argument can be found by `?survexp`.

Details

The function computes

$$L(t) = \frac{\int_0^\infty S^*(u)}{S^*(t)} - \frac{\int_0^\infty S(u)}{S(t)}$$

for a given t, where $S^*(t)$ and $S(t)$ is the survival function in the general population and the patient population, respectively. The integral is computed by Gauss-Legendre quadrature and the point wise variance is estimated using the delta method and numerical differentiation. If `type = mrl`, only $\int_0^\infty S^*(u)/S^*(t)$ is computed.

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)

##Fit flexible parametric relative survival model
fit <- stpm2(Surv(FUyear, status2) ~ 1, data = colonDC, df = 6, bhazard = colonDC$bhaz)

#Compute loss of lifetime from 0 to 20 years
res <- calc.LL(fit, time = seq(0, 20, length.out = 100),
               rmap = list(age = agedays, sex = sex, year = dx))

#Plot the loss of lifetime
plot(res)

##Fit flexible parametric cure model
fit <- FlexCureModel(Surv(FUyear, status2) ~ 1, data = colonDC, n.knots = 7, bhazard = "bhaz",
                     ini.types = "cure")

#Compute loss of lifetime from 0 to 20 years
res.cure <- calc.LL(fit, time = seq(0, 20, length.out = 100),
                    rmap = list(age = agedays, sex = sex, year = dx))

#Plot the loss of lifetime
plot(res, ci = F, ylim = c(0, max(res$Ests[[1]]$ll)))
plot(res.cure, add = T, col = 2, ci = F)


#Compute mean residual lifetime
res.cure <- calc.LL(fit, time = seq(0, 20, length.out = 100), type = "mrl",
                    rmap = list(age = agedays, sex = sex, year = dx))
plot(res.cure)
```

calc.LL.quantile *Compute the time to statistical cure using the loss of lifetime function*

Description

Compute the time to statistical cure using the loss of lifetime function

Usage

```
calc.LL.quantile(fit, q = 2, newdata = NULL, max.time = 20, ci = TRUE,
  expected = NULL, rmap = NULL, ratetable = survexp.dk, tau = 100,
  type = "ll")
```

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.

`ci` Logical. If `TRUE` (default), confidence intervals are computed.

`expected` Object of class `list` containing objects of class `survexp`, with the expected survival of each row in newdata. If not specified, the function computes the expected survival.

`rmap` List to be passed to `survexp` from the `survival` package. 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 points.

`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 *Parametric cure model*

Description

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

Usage

```
fit.cure.model(formula, data, bhazard = NULL, formula.k1 = ~1,
  formula.k2 = NULL, formula.k3 = NULL, type = c("mixture", "nmixture"),
  dist = c("weibull", "exponential", "lognormal"), link = c("logit",
  "loglog", "identity", "probit"), covariance = TRUE, optim.args = NULL)
```

Arguments

`formula` Formula for modelling the the cure fraction Reponse has to be of the form `Surv(time, status)`.

`data` Data frame in which to interpret the variables names in `formula`, `formula.k1`, `formula.k2`, and `formula.k3`.

`bhazard` Background hazard.

`formula.k1` Formula for the first linear predictor of the parametric distribution (see details).

`formula.k2` Formula for the second linear predictor of the parametric distribution (see details).

`formula.k3` Formula for the third linear predictor of the parametric distribution (see details).

`type` A character indicating the type of cure model. Possible values are `mixture` (default) and `nmixture`.

`dist` The parametric distribution of the disease-specific survival function. Possible values are `weibull` (default), `exponential`, and `lognormal`.

`link` Specifies the link function of the cure fraction. Possible values are `logit` (default), `identity`, and `loglog`.

`covariance` Logical. If `TRUE` (default), the covariance matrix is computed.

`optim.args` List with additional arguments passed to `optim`.

`init` Initial values for the maximum likelihood optimization. If not provided, the optimization will start in 0.

Details

The arguments for modelling the parameters of the cure model have different meanings dependent on the chosen distribution.

For the exponential distribution, k1 denotes the rate, for the weibull model, k1 denotes the scale parameter and k2 denotes the shape parameter, sfor the log normal distribution k1 denotes the mu and sigma.

Value

An object of class `CureModel`.

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 <- fit.cure.model(Surv(FUyear, status2) ~ 1, data = colonDC, bhazard = "bhaz",
                      formula.k1 = ~ 1, formula.k2 = ~ 1,
                      type = "mixture", dist = "weibull", link = "logit")

##Plot model
plot(fit)
plot(fit, time = seq(0, 40, length.out = 100))
plot(fit, type = "ehaz")
plot(fit, type = "survuncured")
plot(fit, type = "probcure")


###With covariates
##Fit weibull mixture cure model
fit <- fit.cure.model(Surv(FUyear, status2) ~ age, data = colonDC, bhazard = "bhaz",
                      formula.k1 = ~ age, formula.k2 = ~ 1,
                      type = "mixture", dist = "weibull", link = "logit")

##Plot model
plot(fit, newdata = data.frame(age = 50),
     time = seq(0, 15, length.out = 100), ci = F)
plot(fit, newdata = data.frame(age = 60),
     time = seq(0, 15, length.out = 100), col = 2, ci = F, add = T)

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

`FlexCureModel` *Fit spline-based 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.,

$$R(t|z) = \pi(z) + (1 - \pi(z))S_u(t|z)$$

, where

$$g_1[S_u(t|z)] = \eta_1(z) and g_2(\pi(z)) = \eta_2(z)$$

.

Usage

```
FlexCureModel(formula, data, bhazard = NULL, smooth.formula = ~1,
  knots = NULL, n.knots = 3, knots.time = NULL, n.knots.time = NULL,
  covariance = T, type = c("mixture", "nmixture"), linkpi = c("logit",
  "identity", "loglog", "probit"), linksu = c("loglog", "logit", "probit"),
  verbose = T, constr.optim = F, optim.args = NULL, ortho = TRUE,
  ini.types = c("cure", "flexpara"))
```

Arguments

`formula` Formula for modelling the the cure rate. Reponse has to be of the form `Surv(time, status)`.

`data` Data frame in which to interpret the variables names in `formula`, `smooth.formula`.

`bhazard` Background hazard.

`smooth.formula` Formula for modelling the disease-specific survival of the uncured.

`knots` Knots used for the baseline hazard in the disease-specific survival function.

`n.knots` Number of knots for the disease-specific survival function. The knots are calculated as the equidistant quantiles of the uncensored event-times. If `knots` is supplied, this argument will be ignored.

`knots.time` A named list containing the knots of each of time-varying covariate effect.

`n.knots.time` A named list, containing the number of knots for the time-varying covariate effects. The knots are calculated as the equidistant quantiles of the uncensored event-times. If `knots.time` is supplied, this argument will be ignored.

`covariance` Logical. If `TRUE` (default), the covariance matrix is computed.

`type` A character indicating the type of cure model. Possible values are `mixture` (default) and `nmixture`.

`linkpi` Character giving the link function selected for the cure rate. Possible values are `logit` (default), `identity`, `loglog`, and `probit`.

`linksu` Character giving the link function selected for the survival of the uncured. Possible values are `loglog` (default), `logit`, and `probit`.

`verbose` Logical. If `TRUE` status messages of the function is outputted.

`constr.optim` Logical. If `TRUE` the model is fitted using constraints optimization yielding a non-negative hazard of the uncured (default is `FALSE`). This option is only implemented for `linksu = loglog`.

`optim.args` List with additional arguments passed to `optim`.

`ortho` Logical. If `TRUE` (default), all splines are orthogonalized using a QR-decomposition.

`ini.types` Character vector denoting the executed schemes for computing initial values.

Value

An object of class `fcm`.

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 <- FlexCureModel(Surv(FUyear, status) ~ 1, data = colonDC, n.knots = 5, bhazard = "bhaz")

##Plot model
plot(fit)
plot(fit, time = seq(0, 40, length.out = 100))
plot(fit, type = "ehaz")
plot(fit, type = "survuncured")
plot(fit, type = "probcure")

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


##Fit non-mixture cure model
fit <- FlexCureModel(Surv(FUyear, status) ~ 1, data = colonDC, n.knots = 5, bhazard = "bhaz", type = "nmixture")

##Plot relative survival
plot(fit)

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

###With covariates
##Fit mixture cure model
fit <- FlexCureModel(Surv(FUyear, status) ~ sex, data = colonDC, n.knots = 5, bhazard = "bhaz",
                     smooth.formula = ~ sex)

##Plot model
plot(fit, newdata = data.frame(sex = factor("female", levels = c("male", "female"))),
     time = seq(0, 15, length.out = 100), ci = F)
plot(fit, newdata = data.frame(sex = factor("male", levels = c("male", "female"))),
     time = seq(0, 15, length.out = 100), col = 2, ci = F, add = T)


plot(fit, newdata = data.frame(sex = factor("female", levels = c("male", "female"))),
     time = seq(0, 15, length.out = 100), ci = F, type = "survuncured")
plot(fit, newdata = data.frame(sex = factor("male", levels = c("male", "female"))),
     time = seq(0, 15, length.out = 100), col = 2, ci = F, 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
fit <- FlexCureModel(Surv(FUyear, status) ~ age, data = colonDC, n.knots = 5, bhazard = "bhaz",
                     n.knots.time = list(age = 3))

##Plot model
plot(fit, newdata = data.frame(age = 70))
plot(fit, newdata = data.frame(age = 60), add = T, col = 2)

plot(fit, type = "ehaz", newdata = data.frame(age = 70), ci = F)
plot(fit, type = "ehaz", newdata = data.frame(age = 60), add = T, col = 2, ci = F)
```

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 which variable is the follow-up time in the data
age	Either a numeric vector of age or a character indicating which variable is the age in the data
sex	A numeric vector including the time points for which to make predictions.
data	The data from which to extract variables from. If `time`, `age`, `sex`, and `date` are not characters, leave this empty.
ratetable	Object of class `ratetable` to extract background hazards from. Defaults to survexp.dk
date	A character denoting the type of prediction. Possible values are `relsurv` (default), `curerate`, and `probcure` (see details)

Value

An object of class `numeric` of yearly 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.,

$$R(t|z) = \pi(z) + (1 - \pi(z))S_u(t|z)$$

, where

$$g_1[S_u(t|z)] = \eta_1(z) and g_2(\pi(z)) = \eta_2(z)$$

. This function deviates from `FlexCureModel` in its use of formulas, which allows the use of more smoothers than the restricted cubic splines from the `flexsurv` package.

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", "PO", "probit", "AH", "AO"),
  init = NULL, timeVar = "", covariance = T, verbose = T,
  control = list(maxit = 10000, method = c("Nelder-Mead")),
  ini.types = c("cure", "flexpara"), cure = FALSE)
```

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 `Surv(time, status)`.

`data` Data frame in which to interpret the variables names in `formula`, `smooth.formula`.

`smooth.formula` Formula for describing the time-effect of the survival of the uncured (default is `NULL`).

`smooth.args` List. Optional arguments to the time-effect of the survival of the uncured (default is `NULL`).

`df` Integer. Degrees of freedom (default is 3) for the time-effect of the survival of the uncured.

`tvc` Name list of integers. Specifies the degrees of freedom for a time-varying covariate effect. For instance, `tvc = list(a = 3)` creates a time-varying spline-effect of the covariate a with 3 degrees of freedom.

`tvc.formula` Formula for the time-varying covariate effects. For time-varying effects, a linear term of the covariate has to be included in `formula`.

`bhazard` Background hazard.

`cr.formula` Formula for the cure rate. 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 `mixture` (default) and `nmixture`.

`link.type.cr` Character giving the link function selected for the cure rate. Possible values are `logit` (default), `loglog`, `identity`, and `probit`.

`link.type` Character giving the link function selected for the survival of the uncured. Possible values are `PH` (default), `PO`, `probit`, `AH`, and `AO`.

`init` Providing initial values for the optimization procedure. If not specified, the function will create initial values internally.

`timeVar` Character giving the name of the variable specifying the time component of the `Surv` object.

`covariance` Logical. If `TRUE` (default), the covariance matrix is computed.

`verbose` Logical. If `TRUE` status messages of the function is outputted.

`control` Named list with additional arguments passed to `optim`.

`ini.types` Character vector denoting the executed schemes for computing initial values.

`cure` Logical. Indicates whether a cure model specification is needed for the survival of the uncured. This is usually `FALSE` (default).

Details

This functions generalizes the `FlexCureModel` function using formulas. 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. The function also allows for the use of any smoother from the `mgvc` package.

Value

An object of class `gfcm`.

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 rate
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 rate
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
fit <- GenFlexCureModel(Surv(FUyear, status) ~ age, data = colonDC, df = 4, bhazard = "bhaz",
                        cr.formula = ~ age, tvc = list(age = 2))

##Plot model
plot(fit, newdata = data.frame(age = 70))
plot(fit, newdata = data.frame(age = 60), add = T, col = 2)

plot(fit, type = "hazard", newdata = data.frame(age = 70), ci = "n")
plot(fit, type = "hazard", newdata = data.frame(age = 60), add = T, col = 2, ci = "n")
```

lts *Long term survival predictions*

Description

Function for computing loss of lifetime estimates from an estimated relative survival model

Usage

```
lts(fit, newdata = NULL, time = NULL, ci = T, expected = NULL,
  ratetable = survexp.dk, rmap = rmap)
```

Arguments

`fit`	Fitted model to do predictions from. Possible classes are `fcm`, `gfcm`, `stpm2`, `pstpm2`, and `cm`.
`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 the maximum follow-up time is selected.

`ci` Logical indicating whether confidence intervals should be computed.

`expected` Object of class `list` containing objects of class `survexp` denoting the expected survival of each row in newdata. If not specified, the function computes the expected survival.

`ratetable` Object of class `ratetable` to compute the general population survival from.

`rmap` List to be passed to `survexp` from the `survival` package. Detailed documentation on this argument can be found by `?survexp`.

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, status2) ~ 1, data = colonDC, df = 6, bhazard = colonDC$bhaz)

##Compute survival probabilities from 0 to 20 years
pred <- predict.lts(fit, time = seq(0, 20, length.out = 100),
                    rmap = list(age = agedays, sex = sex, year = dx))

##Plot the survival function
plot(pred)

##Fit flexible parametric cure model
fit <- FlexMixtureCureModel(Surv(FUyear, status2) ~ 1, data = colonDC, n.knots = 7, bhazard = "bhaz",
                            ini.types = "cure")

##Compute survival probabilities from 0 to 20 years
pred.cure <- predict.lts(fit, time = seq(0, 20, length.out = 100),
                         rmap = list(age = agedays, sex = sex, year = dx))

##Plot the loss of lifetime
plot(pred, ci = F)
plot(pred.cure, add = T, col = 2, ci = F)
```

`plot.crude` *Plot crude event probabilities*

Description

Plot function for the computed crude event probabilties.

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 type is `cancer`, the label is "Cumulative incidence of cancer related death", if `other`, the label is "Cumulative incidence of death from other causes than cancer", and if type is `othertime`, the label is "probability of eventually dying from other causes than cancer".

`xlab` Label of the x-axis (default is "Time").

`add` Logical indicating wether the cruves should be added to the current plot window (default is FALSE).

`...` Further arguments passed to `plot` and `lines`.

`plot.cuRe` *Plot function for Flexible mixture cure model*

Description

Plot function associated with the class `fmcm`

Usage

```
plot.cuRe(fit, newdata = NULL, type = c("relsurv", "ehaz", "probcure",
  "survuncured"), time = NULL, xlim = NULL, ylim = c(0, 1),
  xlab = "Time", ylab = NULL, col = 1, ci = NULL, non.parametric = F,
  col.non.para = 2, include.knots = F, add = F, ...)
```

Arguments

`fit` Object of class `cuRe`.

`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 function to plot. Possible values are `relsurv` (default) for the relative survival, `ehaz` for the excess hazard, `probcure` for the conditional probability of being cured, and `survuncured` for the survival of the uncured patients.

`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.

`col` Colour of each line.

`ci` Logical. If `TRUE` (default), confidence intervals are added.

`non.parametric` Logical. If `TRUE`, a non-parametic estimate (Ederer II method) is added to the plot (default is FALSE).

`col.non.para` Colour of the non-parametric estimate.

`include.knots` Logical. If `TRUE` vertical lines are added at each knot of the baseline spline function (default is `FALSE`).

`xlabel` Label of the x-axis. Default is "Time".

`ylabel` Label of the y-axis. Default is "Relative survival" if `type = relsurv`, "Excess hazard" if `type = ehaz`, "Conditional probability of cure" if `type = probcure`, "Disease-specific survival of the uncured" if `type = survuncured`.

Value

A plot containing the predictions of each observation in `newdata`.

`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. 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`.

plot.lts *Plot the long term survival*

Description

Plot function for the class lts

Usage

```
plot.lts(obj, ylim = NULL, xlim = NULL, ci = T, 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 occording 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.

predict.cm *Predict function for cure models*

Description

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

Usage

```
predict.cm(fit, newdata = NULL, type = c("relsurv", "ehaz", "probcure",
  "survuncured"), time = NULL, ci = T, pars = NULL)
```

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	Type of prediction to do. Possible values are relsurv (default) for the relative survival, curerate for the cure rate, ehaz for the excess hazard, probcure for the conditional probability of being cured, and survuncured for the disease-specific survival of the uncured.

`time` Optional time points at which to compute predictions.

`ci` Logical. If `TRUE`, confidence intervals are computed.

`pars` Numerical vector containing the parameters values of the model.

Value

An object of class `matrix` including the predictions.

predict.fcm *Predict function for Flexible mixture cure model*

Description

Function for doing predictions for class `fmcm`

Usage

```
predict.fcm(fit, newdata = NULL, type = c("relsurv", "ehaz", "probcure",
  "survuncured"), time = NULL, ci = T, pars = NULL)
```

Arguments

`fit` Object of class `fcm` 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` Type of prediction to do. Possible values are `relsurv` (default) for the relative survival, `curerate` for the cure rate, `ehaz` for the excess hazard, `probcure` for the conditional probability of being cured, and `survuncured` for the disease-specific survival of the uncured.

`time` Optional time points at which to compute predictions. This argument is not used if type is `curerate`.

`ci` Logical. If `TRUE`, confidence intervals are computed.

`pars` Numerical vector containing the parameters values of the model. In general, this argument can be ignored by the user.

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 seperately for male and female Danish citizens.
The data can be accessed through:
Female: http://www.mortality.org/hmd/DNK/STATS/fltper_1x1.txt
Male: http://www.mortality.org/hmd/DNK/STATS/mltper_1x1.txt

Index