Package 'blendR'

September 3, 2025

Title Blended Survival Curves

Version 1.0.0

Description Create a blended curve from two survival curves, which is particularly useful for survival extrapolation in health technology assessment. The main idea is to mix a flexible model that fits the observed data well with a parametric model that encodes assumptions about long-term survival. The two curves are blended into a single survival curve that is identical to the first model over the range of observed times and gradually approaches the parametric model over the extrapolation period based on a given weight function. This approach allows for the inclusion of external information, such as data from registries or expert opinion, to guide long-term extrapolations, especially when dealing with immature trial data. See Che et al. (2022) <doi:10.1177/0272989X221134545>.

License GPL (>= 3)

Additional_repositories https://giabaio.r-universe.dev, https://inla.r-inla-download.org/R/stable

Encoding UTF-8

RoxygenNote 7.3.2.9000

Imports dplyr, flexsurv, ggplot2, manipulate, sn, survHE, tibble

Depends R (>= 4.4.0)

Suggests INLA, knitr, remotes, rlang, rmarkdown, survHEhmc, survival, testthat (>= 3.0.0)

URL https://github.com/StatisticsHealthEconomics/blendR/, https://StatisticsHealthEconomics.github.io/blendR/

BugReports https://github.com/StatisticsHealthEconomics/blendR/issues/

Config/testthat/edition 3

NeedsCompilation no

Author Nathan Green [aut] (ORCID: <https://orcid.org/0000-0003-2745-1736>, ROR: <https://ror.org/02jx3x895>),
Zhaojing Che [aut, cph, cre] (ORCID: <https://orcid.org/0000-0003-2245-1606>, ROR: <https://ror.org/052gg0110>)

Maintainer Zhaojing Che <blendr-pkg@proton.me>

Repository CRAN

Date/Publication 2025-09-03 20:50:02 UTC

Contents

blendsurv *Blended survival object*

Description

This is the main function in the **blendR** package. Two survival curves are supplied and blended according to the blending distribution characterised by the blending interval and the beta distribution parameters.

Usage

```
blendsurv(
  obs_Surv,
  ext_Surv,
  blend_interv,
  beta_params = list(alpha = 3, beta = 3),
  times = NULL,
  nsim = 100
)
```

Arguments

`obs_Surv, ext_Surv`
: Observed and external data survival curves. These can come from **survHE**, **INLA** or **flexsurv** fits.

`blend_interv`
: Maximum and minimum values for the blending interval.

`beta_params`
: coefficients of a beta distribution

`times`
: A vector of times for which the survival curves are to be computed; optional

`nsim`
: The number of simulations from the distribution of the survival curves; default 100

Value

List of S for observed, external and blended curves.

Examples

```
library(survHE)

## trial data
data("TA174_FCR", package = "blendR")

## externally estimated data
data_sim <- ext_surv_sim(t_info = 144,
                         S_info = 0.05,
                         T_max = 180)

obs_Surv <- fit.models(formula = Surv(death_t, death) ~ 1,
                       data = dat_FCR,
                       distr = "exponential",
                       method = "hmc")

ext_Surv <- fit.models(formula = Surv(time, event) ~ 1,
                       data = data_sim,
                       distr = "exponential",
                       method = "hmc")

blend_interv <- list(min = 48, max = 150)
beta_params <- list(alpha = 3, beta = 3)

ble_Surv <- blendsurv(obs_Surv, ext_Surv, blend_interv, beta_params)

plot(ble_Surv)
```

`dat_FCR` *Survival data*

Description

Survival data

ext_surv_sim *Create an external survival data based on expert opinion*

Description

Generally, the sampling is done is two steps

$$p(T) = p(T|intervali)p(intervali)$$

Usage

```
ext_surv_sim(t_info, S_info, T_max, n = 100)
```

Arguments

t_info	A vector of times for which expert opinion is elicited
S_info	A vector of mean survival probabilities estimated by experts corresponding to time points in t_info
T_max	The maximum survival time to be used
n	The number of patients to construct the artificial external data set; default 100

Details

In particular $T\ U(x_i, x_{i+1})\ i\ multinomial(\hat{\pi})$

Value

Dataframe of times and censoring status

Examples

```
dat <- ext_surv_sim(t_info = c(10,20,50),
                    S_info = c(0.9, 0.8, 0.2),
                    T_max = 100, n = 100)
if (require(survival)) {
    km_fit <- survfit(Surv(time, event) ~ 1, data = dat)
    plot(km_fit)
}
```

fit_inla_pw *Generate survival estimates with a piecewise exponential Cox model (using INLA)*

Description

Generate survival estimates with a piecewise exponential Cox model (using INLA)

Usage

```
fit_inla_pw(
  inla.formula = inla.surv(death_t, death) ~ -1,
  data,
  cutpoints,
  nsim = 100,
  ...
)
```

Arguments

`inla.formula`	The formula for PEM which must be an `inla.surv` object
`data`	A dataframe for survival data with time (`death_t`) and event (`death`)
`cutpoints`	A sequence of cut points for intervals in the baseline hazard
`nsim`	The number of simulations from posteriors; default 100
`...`	Additional arguments

Value

INLA object

Examples

```
if (requireNamespace("INLA", quietly = TRUE)) {
 data("TA174_FCR", package = "blendR")
 head(dat_FCR)
 obs_Surv <- fit_inla_pw(data = dat_FCR, cutpoints = seq(0, 180, by = 5))
}
```

make_surv_methods *Create survival probabilities*

Description

These function are version of the survHE::make.surv() function from **survHE**. These are needed prior to blending.

Usage

```
make_surv(Surv, ...)

## S3 method for class 'survHE'
make_surv(Surv, t, nsim = 100, ...)

## S3 method for class 'flexsurvreg'
make_surv(Surv, t = NULL, nsim = 100, ...)

## S3 method for class 'inla'
make_surv(Surv, t = NULL, nsim = 100, ...)

## Default S3 method:
make_surv(Surv, t = NULL, nsim = 100, ...)
```

Arguments

Surv	survival analysis object
...	Additional arguments
t	Time points; vector
nsim	Number of simulations; integer

Value

Matrix of survival probabilities

Examples

```
library(survHE)

## trial data
data("TA174_FCR", package = "blendR")

## externally estimated data
data_sim <- ext_surv_sim(t_info = 144,
                         S_info = 0.05,
                         T_max = 180)
```

```
ext_Surv <- fit.models(formula = Surv(time, event) ~ 1,
                       data = data_sim,
                       distr = ”exponential”,
                       method = ”hmc”)

S_ext <- make_surv(ext_Surv, t = 1:100, nsim = 100)
```

`manip_plot` *Blended survival plot with manipulate*

Description

RStudio bug need to run base R first `manipulate(plot(1:x), x = slider(5, 10))`

Usage

```
manip_plot(obs_Surv, ext_Surv, blend_interv)
```

Arguments

`obs_Surv`	Observed survival
`ext_Surv`	External survival
`blend_interv`	Blending interval

Value

Blended survival plot

`plot.blended` *Blended survival curve based on short-term data and external information*

Description

Blended survival curve based on short-term data and external information

Usage

```
## S3 method for class 'blended'
plot(x, alpha = c(0.1, 0.05), ...)
```

Arguments

`x`	A blended survival curve object obtain from `blendsurv()`
`alpha`	A vector specifying the opacity of ribbon for the blended curve and other curves
`...`	Additional arguments

Value

A **ggplot2** object

See Also

`blendsurv()`

Examples

```
library(survHE)

## trial data
data("TA174_FCR", package = "blendR")

## externally estimated data
data_sim <- ext_surv_sim(t_info = 144,
                         S_info = 0.05,
                         T_max = 180)

obs_Surv <- fit.models(formula = Surv(death_t, death) ~ 1,
                       data = dat_FCR,
                       distr = "exponential",
                       method = "hmc")

ext_Surv <- fit.models(formula = Surv(time, event) ~ 1,
                       data = data_sim,
                       distr = "exponential",
                       method = "hmc")

blend_interv <- list(min = 48, max = 150)
beta_params <- list(alpha = 3, beta = 3)

ble_Surv <- blendsurv(obs_Surv, ext_Surv, blend_interv, beta_params)

plot(ble_Surv)
```

`weightplot` *Plots the weights for the blending procedure*

Description

Plots the weights for the blending procedure

Usage

```
weightplot(x, ...)
```

Arguments

`x` A blended survival curve object obtained from `blendsurv()`

`...` Additional arguments

Value

ggplot2 object

See Also

`blendsurv()`

Index