Package ‘survextrap’

January 13, 2026

Title Bayesian Flexible Parametric Survival Modelling and Extrapolation

Version 1.0.1

Date 2026-01-11

Description

Survival analysis using a flexible Bayesian model for individual-level right-censored data, optionally combined with aggregate data on counts of survivors in different periods of time. An M-spline is used to describe the hazard function, with a prior on the coefficients that controls overfitting. Proportional hazards or flexible non-proportional hazards models can be used to relate survival to predictors. Additive hazards (relative survival) models, waning treatment effects, and mixture cure models are also supported. Priors can be customised and calibrated to substantive beliefs. Posterior distributions are estimated using 'Stan', and outputs are arranged in a tidy format. See Jackson (2023) <doi:10.1186/s12874-023-02094-1>.

License GPL (>= 3)

Encoding UTF-8

RoxygenNote 7.3.2

Biarch true

Depends R (>= 3.5.0), survival

Imports ggplot2 (>= 3.4.0), gridExtra, loo, methods, posterior, Rcpp (>= 0.12.0), rstan (>= 2.26.0), splines2, tibble

LinkingTo BH (>= 1.66.0), Rcpp (>= 0.12.0), RcppEigen (>= 0.3.3.3.0), RcppParallel (>= 5.0.1), rstan (>= 2.26.0), StanHeaders (>= 2.26.0)

SystemRequirements GNU make

Suggests rstantools (>= 2.3.1), knitr, rmarkdown, testthat, SHELF, flexsurv, flexsurvcure, splines, dplyr, viridis, forcats, fs, purrr, tidyr, stringr, survminer

VignetteBuilder knitr

URL `https://github.com/chjackson/survextrap`, `https://chjackson.github.io/survextrap/`

BugReports `https://github.com/chjackson/survextrap/issues`

LazyData true

NeedsCompilation yes

Author Christopher Jackson [aut, cre, cph] (ORCID:
<https://orcid.org/0000-0002-6656-8913>),
Iain Timmins [ctb] (ORCID: <https://orcid.org/0000-0002-8761-582X>),
Michael Sweeting [ctb] (ORCID: <https://orcid.org/0000-0003-0980-8965>)

Maintainer Christopher Jackson <chris.jackson@mrc-bsu.cam.ac.uk>

Repository CRAN

Date/Publication 2026-01-13 16:00:02 UTC

Contents

`survextrap-package` *The 'survextrap' package.*

Description

For an introduction to and overview of the `survextrap` package, and full documentation, see `https://chjackson.github.io/survextrap/`.

For details of the methods, see the paper by Jackson (2023).

Author(s)

Maintainer: Christopher Jackson `<chris.jackson@mrc-bsu.cam.ac.uk>` (ORCID) [copyright holder]

Other contributors:

- Iain Timmins (ORCID) [contributor]
- Michael Sweeting (ORCID) [contributor]

References

Jackson, C. (2023) `survextrap`: a package for flexible and transparent survival extrapolation. BMC Medical Research Methodology 23:282. doi:10.1186/s12874023020941

Timmins I, Torabi F, Jackson C, Lambert P, Sweeting M J. (2025) Simulation-based assessment of a Bayesian survival model with flexible baseline hazard and time-dependent effects. doi:10.48550/arXiv.2503.21388.

See Also

Useful links:

- `https://github.com/chjackson/survextrap`
- `https://chjackson.github.io/survextrap/`
- Report bugs at `https://github.com/chjackson/survextrap/issues`

cetux *Datasets for evaluation of cetuximab in head and neck cancer*

Description

Datasets for evaluation of cetuximab in head and neck cancer, as previously analysed by Guyot et al. (2017) to demonstrate models for survival extrapolation with Bayesian evidence synthesis.

Usage

```
cetux

cetux_seer

cetux_bh
```

Format

`cetux` contains synthetic individual-level survival data, generated (using the method of Guyot et al 2012) to be consistent with the Kaplan-Meier estimates of survival published by Bonner et al. (2006). Columns `months`, `years`, give survival in months or years since the date of diagnosis of head and neck cancer. `d` is a numeric vector where 0 indicates censoring, and 1 indicates death at this time. `treat` is a factor indicating the treatment group (Cetuximab or Control); both groups also received radiotherapy.

`cetux_seer` Estimates of conditional survival from registry data, matched to the Bonner trial population by age, gender, cancer site, and date of diagnosis. From the "Surveillance Epidemiology and End Results" (SEER) database. Each line gives counts of `r` survivors up to `stop` years, given `n` people alive at `start`. `haz` is the corresponding constant hazard estimate over this period, computed as `-log(r/n)`. There are also 95% interval estimates for the hazard based on the data on one period at a time, derived from Bayesian principles as `-log(qbeta(c(0.975, 0.025), r, n-r))`.

`cetux_bh` Mortality rates for the population of the USA, matched by age and sex to the patients from the Bonner trial. 80% are male, and the median age is 57 (range 34 to 83). Hence the ith row is a weighted average of the male and and female mortality rates for age $57 + i - 1$.

See Guyot et al. (2017) for more details of each of these.

An object of class `data.frame` with 424 rows and 4 columns.

An object of class `data.frame` with 21 rows and 8 columns.

An object of class `data.frame` with 54 rows and 2 columns.

References

Guyot, P., Ades, A.E., Beasley, M., Lueza, B., Pignon, J.P. and Welton, N.J., (2017). Extrapolation of survival curves from cancer trials using external information. Medical Decision Making, 37(4), pp.353-366.

Bonner, J.A., Harari, P.M., Giralt, J., Azarnia, N., Shin, D.M., Cohen, R.B., Jones, C.U., Sur, R., Raben, D., Jassem, J. and Ove, R., (2006) Radiotherapy plus cetuximab for squamous-cell carcinoma of the head and neck. New England Journal of Medicine, 354(6), pp.567-578.

Guyot, P., Ades, A.E., Ouwens, M.J. and Welton, N.J., (2012) Enhanced secondary analysis of survival data: reconstructing the data from published Kaplan-Meier survival curves. BMC Medical Research Methodology, 12, pp.1-13.

colons *Colon cancer survival data*

Description

Survival times of 191 patients from a trial of chemotherapy for colon cancer. This is the data provided in the `survival` package as `colon`, but this is restricted to outcome of recurrence-free survival, artificially censored at 3 years, and taking a 20% random sample.

Usage

```
colons
```

Format

Documented in `colon`.

Source

See `colon`.

References

See `colon`.

cumhaz *Estimates of cumulative hazard from a survextrap model*

Description

Estimates of the cumulative hazard at particular times, from a `survextrap` model

Usage

```
cumhaz(
  x,
  newdata = NULL,
  t = NULL,
  tmax = NULL,
  niter = NULL,
  summ_fns = NULL,
  sample = FALSE,
  newdata0 = NULL,
  wane_period = NULL,
  wane_nt = 10
)
```

Arguments

`x` A fitted model object as returned by `survextrap`

`newdata` Data frame of covariate values to compute the output for. If there are covariates in the model and this is not supplied, the following default is used:

(a) if the only covariate is one factor variable, then the output is computed for each level of this factor.

(b) if there are multiple covariates, or any numeric covariates, then the output is computed at the mean of each numeric covariate in the original data, and at the baseline level of each factor covariate.

Note: caution is required about how treatment groups (for example) are stored in your data. If these are coded as numeric (0/1), then if `newdata` is not specified, only one output will be shown. This relates to the average value of this numeric variable over the data, which doesn't correspond to either of the treatment groups. To avoid this, a treatment group should be stored as a factor.

`t` Vector of times at which to compute the estimates.

`tmax` Maximum time at which to compute the estimates. If `t` is supplied, then this is ignored. If `t` is not supplied, then `t` is set to a set of 100 equally spaced time points from 0 to `tmax`. If both `tmax` and `t` are not supplied, then `tmax` is set to the maximum follow up time in the data.

`niter` Number of MCMC iterations to use to compute credible intervals. Set to a low value to make this function quicker, at the cost of some approximation error (which may not be important for plotting or model development).

`summ_fns` A list of functions to use to summarise the posterior sample. This is passed to `posterior::summarise_draws`. By default this is `list(median=median, ~quantile(.x, probs=c(0.025, 0.975)))`. If the list is named, then the names will be used for the columns of the output.

`sample` If `TRUE` then the MCMC samples are returned instead of being summarised as a median and 95% credible intervals.

`newdata0` Data frame of covariate values defining the "untreated" group for use in treatment waning models. See `Survmspline_wane`.

`wane_period` Vector of two numbers, defining the time period over which the hazard is interpolated between the hazard of the "treated" group (taken from `newdata`) and the hazard of the "untreated" group (taken from `newdata0`). Optional - if this is not supplied, then no waning is assumed.

`wane_nt` Number of intervals defining the piecewise constant approximation to the hazard during the waning period.

Value

A data frame (tibble) giving posterior summary statistics, or (if `sample=TRUE`) an array giving samples from the posterior.

`curedata` *Simulated data for testing mixture cure models*

Description

Survival times of 200 fake people. The cure probability is 0.5 for `x=0`, and the log odds ratio for cure is 0.5, so that the cure probability is about 0.62 for `x=1`. The uncured population have survival times distributed as a Weibull with shape 1.5 and scale 1.2.

Usage

```
curedata
```

Format

`t` Survival times, right-censored at 10 years.

`status`. 1 for an observed death and 0 for censoring.

`x`. A numeric variable with value of 0 for 100 individuals, and 1 for the other 100.

Source

Simulated.

get_draws *Posterior draws from a survextrap model*

Description

Return the matrix of draws from the posterior distribution of parameters in a `survextrap` model, with all chains collapsed.

Usage

```
get_draws(x)
```

Arguments

`x`	A fitted model object as returned by `survextrap`

Value

A matrix of draws in the `draws_matrix` format of the **posterior** package.

hazard *Estimates of hazard from a `survextrap` model*

Description

Estimates of the hazard function at particular times, from a `survextrap` model

Usage

```
hazard(
  x,
  newdata = NULL,
  t = NULL,
  tmax = NULL,
  niter = NULL,
  summ_fns = NULL,
  sample = FALSE,
  newdata0 = NULL,
  wane_period = NULL,
  wane_nt = 10
)
```

Arguments

`x` A fitted model object as returned by `survextrap`

`newdata` Data frame of covariate values to compute the output for. If there are covariates in the model and this is not supplied, the following default is used:

(a) if the only covariate is one factor variable, then the output is computed for each level of this factor.

(b) if there are multiple covariates, or any numeric covariates, then the output is computed at the mean of each numeric covariate in the original data, and at the baseline level of each factor covariate.

Note: caution is required about how treatment groups (for example) are stored in your data. If these are coded as numeric (0/1), then if `newdata` is not specified, only one output will be shown. This relates to the average value of this numeric variable over the data, which doesn't correspond to either of the treatment groups. To avoid this, a treatment group should be stored as a factor.

`t` Vector of times at which to compute the estimates.

`tmax` Maximum time at which to compute the estimates. If `t` is supplied, then this is ignored. If `t` is not supplied, then `t` is set to a set of 100 equally spaced time points from 0 to `tmax`. If both `tmax` and `t` are not supplied, then `tmax` is set to the maximum follow up time in the data.

`niter` Number of MCMC iterations to use to compute credible intervals. Set to a low value to make this function quicker, at the cost of some approximation error (which may not be important for plotting or model development).

`summ_fns` A list of functions to use to summarise the posterior sample. This is passed to `posterior::summarise_draws`. By default this is `list(median=median, ~quantile(.x, probs=c(0.025, 0.975)))`. If the list is named, then the names will be used for the columns of the output.

`sample` If `TRUE` then the MCMC samples are returned instead of being summarised as a median and 95% credible intervals.

`newdata0` Data frame of covariate values defining the "untreated" group for use in treatment waning models. See `Survmspline_wane`.

`wane_period` Vector of two numbers, defining the time period over which the hazard is interpolated between the hazard of the "treated" group (taken from `newdata`) and the hazard of the "untreated" group (taken from `newdata0`). Optional - if this is not supplied, then no waning is assumed.

`wane_nt` Number of intervals defining the piecewise constant approximation to the hazard during the waning period.

Value

A data frame (tibble) giving posterior summary statistics, or (if `sample=TRUE`) an array giving samples from the posterior.

hazard_ratio *Hazard ratio against time in a survextrap model*

Description

Compute the hazard ratio at a series of time points, estimated from a survextrap model. Intended for use with non-proportional hazards models (`survextrap(...,nonprop=TRUE)`). In proportional hazards models (which survextrap fits by default) the hazard ratio is constant with time.

Usage

```
hazard_ratio(
  x,
  newdata = NULL,
  t = NULL,
  tmax = NULL,
  niter = NULL,
  summ_fns = NULL,
  sample = FALSE
)
```

Arguments

`x` A fitted model object as returned by survextrap

`newdata` A data frame with two rows. The hazard ratio will be defined as hazard(second row) divided by hazard(first row). If the only covariate in the model is a factor with two levels, then `newdata` defaults to a data frame containing the levels of this factor, so that the hazard ratio for the second level versus the first level is computed. For any other models, `newdata` must be supplied explicitly.

Standardisation (with standardise_to) is not supported. This might be done by hand by using `hazard(...,sample=TRUE)` to obtain posterior samples for the two standardised hazards separately, then summarising by hand.

`t` Vector of times at which to compute the estimates.

`tmax` Maximum time at which to compute the estimates. If `t` is supplied, then this is ignored. If `t` is not supplied, then `t` is set to a set of 100 equally spaced time points from 0 to `tmax`. If both `tmax` and `t` are not supplied, then `tmax` is set to the maximum follow up time in the data.

`niter` Number of MCMC iterations to use to compute credible intervals. Set to a low value to make this function quicker, at the cost of some approximation error (which may not be important for plotting or model development).

`summ_fns` A list of functions to use to summarise the posterior sample. This is passed to posterior::summarise_draws. By default this is `list(median=median, ~quantile(.x, probs=c(0.025, 0.975)))`. If the list is named, then the names will be used for the columns of the output.

`sample` If `TRUE` then the MCMC samples are returned instead of being summarised as a median and 95% credible intervals.

Value

A data frame (tibble) with each row containing posterior summary statistics for different times.

Or if `sample=TRUE`, an array with dimensions `length(t)`, `niter`, and 1, giving the incremental RMST evaluated at different times and MCMC iterations respectively.

`hrtime` *Hazard ratio between high and low values of the hazard over time*

Description

This is intended as an intuitive single-number measure of how much a hazard function changes over time. The hazard is computed on an equally-spaced fine grid between the boundary knots. The ratio between a "high" and "low" one of these hazard values is computed. For example, if the hazard is constant over time, then this hazard ratio will be 1.

Usage

```
hrtime(
  x,
  newdata = NULL,
  niter = NULL,
  summ_fns = NULL,
  hq = c(0.1, 0.9),
  sample = FALSE
)
```

Arguments

`x` A fitted model object as returned by `survextrap`

`newdata` Data frame of covariate values to compute the output for. If there are covariates in the model and this is not supplied, the following default is used:

(a) if the only covariate is one factor variable, then the output is computed for each level of this factor.

(b) if there are multiple covariates, or any numeric covariates, then the output is computed at the mean of each numeric covariate in the original data, and at the baseline level of each factor covariate.

Note: caution is required about how treatment groups (for example) are stored in your data. If these are coded as numeric (0/1), then if `newdata` is not specified, only one output will be shown. This relates to the average value of this numeric variable over the data, which doesn't correspond to either of the treatment groups. To avoid this, a treatment group should be stored as a factor.

`niter` Number of MCMC iterations to use to compute credible intervals. Set to a low value to make this function quicker, at the cost of some approximation error (which may not be important for plotting or model development).

`summm_fns` A list of functions to use to summarise the posterior sample. This is passed to `posterior::summarise_draws`. By default this is `list(median=median, ~quantile(.x, probs=c(0.025, 0.975)))`. If the list is named, then the names will be used for the columns of the output.

`hq` Quantiles which define the "low" and "high" values of a time-varying quantity (hazard in `prior_haz_sd` and the hazard ratio in `prior_hr_sd`). The ratio between the high and low values will be summarised, as a measure of time-dependence. By default, this is `c(0.1, 0.9)`, so that the 10% and 90% quantiles are used respectively.

`sample` If `TRUE` then the MCMC samples are returned instead of being summarised as a median and 95% credible intervals.

Value

A summary of the posterior distribution of this hazard ratio from the fitted model, as a data frame with one row per covariate value requested in `newdata`, and one column for each posterior summary statistic.

Or if `sample=TRUE`, an array with dimensions `1`, `niter`, and `nrow(newdata)`, giving the incremental RMST evaluated at different MCMC iterations and covariate values respectively.

`irmst` *Incremental restricted mean survival time*

Description

Compute the difference in the restricted mean survival times between two covariate values (e.g. treatment groups).

Usage

```
irmst(
  x,
  t,
  newdata = NULL,
  newdata0 = NULL,
  wane_period = NULL,
  wane_nt = 10,
  niter = NULL,
  summ_fns = NULL,
  sample = FALSE,
  disc_rate = 0,
  method = "gl",
  gl_nodes = 100
)
```

Arguments

`x` A fitted model object as returned by `survextrap`

`t` Vector of times. The restricted mean survival time up to each one of these times will be computed.

`newdata` A data frame with two rows. The result will be the restricted mean for the covariates in the second row, minus the restricted mean for the covariates in the first row. If `newdata` is omitted for models where the only covariate is a factor with two levels, then this is taken from these levels. Otherwise `newdata` must be supplied explicitly.

`newdata0` Data frame of covariate values defining the "untreated" group for use in treatment waning models. See `Survmspline_wane`.

`wane_period` Vector of two numbers, defining the time period over which the hazard is interpolated between the hazard of the "treated" group (taken from `newdata`) and the hazard of the "untreated" group (taken from `newdata0`). Optional - if this is not supplied, then no waning is assumed.

`wane_nt` Number of intervals defining the piecewise constant approximation to the hazard during the waning period.

`niter` Number of MCMC iterations to use to compute credible intervals. Set to a low value to make this function quicker, at the cost of some approximation error (which may not be important for plotting or model development).

`summ_fns` A list of functions to use to summarise the posterior sample. This is passed to `posterior::summarise_draws`. By default this is `list(median=median, ~quantile(.x, probs=c(0.025, 0.975)))`. If the list is named, then the names will be used for the columns of the output.

`sample` If `TRUE` then an MCMC sample is returned from the posterior of the output, rather than summary statistics.

`disc_rate` Discounting rate used to calculate the discounted mean or restricted mean survival time, using an exponential discounting function.

`method` Method of numerical integration to obtain the restricted mean survival time from the survival function.

The default is `method="gl"`, a Gauss-Legendre method with 100 nodes between zero and the maximum of `t`.

`method="adaptive"` uses the base R `integrate` function, which is much slower, but potentially more robust for badly-behaved survival functions.

`gl_nodes` Number of nodes for the Gauss-Legendre method.

Details

The posterior distribution is obtained by calling `rmst` for each group, obtaining each posterior sample from the `"sample"` attribute, and taking the difference to get a posterior sample for the difference.

Value

A data frame (tibble) with each row containing posterior summary statistics for a particular time and covariate value.

Or if `sample=TRUE`, an array with dimensions `length(t)`, `niter`, `nrow(newdata)`, giving the incremental RMST evaluated at different times, MCMC iterations and covariate values respectively.

---

mean.survextrap *Mean survival time*

Description

Compute the mean survival time from a model fitted with `survextrap`. Defined as the integral of the fitted survival curve from zero to infinity. This relies on numerical integration, which is done for every parameter in the MCMC sample, so it may be slow.

Usage

```
## S3 method for class 'survextrap'
mean(
  x,
  newdata = NULL,
  newdata0 = NULL,
  wane_period = NULL,
  wane_nt = 10,
  disc_rate = 0,
  niter = NULL,
  summ_fns = NULL,
  sample = FALSE,
  ...
)
```

Arguments

`x` A fitted model object as returned by `survextrap`

`newdata` Data frame of covariate values to compute the output for. If there are covariates in the model and this is not supplied, the following default is used:

(a) if the only covariate is one factor variable, then the output is computed for each level of this factor.

(b) if there are multiple covariates, or any numeric covariates, then the output is computed at the mean of each numeric covariate in the original data, and at the baseline level of each factor covariate.

Note: caution is required about how treatment groups (for example) are stored in your data. If these are coded as numeric (0/1), then if `newdata` is not specified, only one output will be shown. This relates to the average value of this numeric variable over the data, which doesn't correspond to either of the treatment groups. To avoid this, a treatment group should be stored as a factor.

`newdata0` Data frame of covariate values defining the "untreated" group for use in treatment waning models. See `Survmspline_wane`.

`wane_period` Vector of two numbers, defining the time period over which the hazard is interpolated between the hazard of the "treated" group (taken from `newdata`) and the hazard of the "untreated" group (taken from `newdata0`). Optional - if this is not supplied, then no waning is assumed.

`wane_nt` Number of intervals defining the piecewise constant approximation to the hazard during the waning period.

`disc_rate` Discounting rate used to calculate the discounted mean or restricted mean survival time, using an exponential discounting function.

`niter` Number of MCMC iterations to use to compute credible intervals. Set to a low value to make this function quicker, at the cost of some approximation error (which may not be important for plotting or model development).

`summ_fns` A list of functions to use to summarise the posterior sample. This is passed to `posterior::summarise_draws`. By default this is `list(median=median, ~quantile(.x, probs=c(0.025, 0.975)))`. If the list is named, then the names will be used for the columns of the output.

`sample` If `TRUE` then an MCMC sample is returned from the posterior of the output, rather than summary statistics.

`...` Other options (currently unused).

Details

Additionally for some models, the integration up to infinity may not converge, giving an error message. This typically occurs if there is a substantial probability of high survival times or zero hazards at later times. The restricted mean survival time can usually be computed in these situations with `rmst`, but the model should also be investigated to ensure the posterior distributions are realistic, and simplified or supplemented with external data or informative priors if appropriate.

Value

A data frame with each row containing posterior summary statistics for a particular covariate value.

An attribute `"sample"` is also returned, containing a matrix of samples from the posterior distribution of the RMST.

`msplinemodel_init` *Create an M-spline survival model, both structure and parameters.*

Description

`mspline_init` is first used to create the M-spline model structure, including knot positions. Parameters including basis coefficients and scale are either supplied or set to a default that defines a constant hazard model.

Usage

```
msplinemodel_init(
  df = 10,
  degree = 3,
  bsmooth = TRUE,
  knots = NULL,
  bknot = 10,
  obstimes = NULL,
  coefs = NULL,
  hscale = 1
)
```

Arguments

`df` Desired number of basis terms, or "degrees of freedom" in the spline. If `knots` is not supplied, the number of knots is then chosen to satisfy this.

`degree` Spline polynomial degree. Can only be changed from the default of 3 if `bsmooth` is `FALSE`.

`bsmooth` If `TRUE` then the function is constrained to also have zero derivative and second derivative at the boundary.

`knots` Vector of knot locations. If not supplied, `df` has to be specified. One of two rules is then used to choose the knot locations. If `bknot` is specified, a set of equally spaced knots between zero and `bknot` is used. Otherwise if `obstimes` is supplied, the knots are chosen as equally spaced quantiles of `obstimes`.

The number of knots (excluding zero) is `df - degree + 1` if `bsmooth` is `TRUE`, or `df - degree - 1` otherwise.

`bknot` Location of the final spline knot.

`obstimes` Vector of observation times whose quantiles will be used to choose knot locations

`coefs` Basis coefficients

`hscale` Hazard scale parameter

Details

This function is not for fitting models to data, but for setting up a theoretical M-spline model for illustration.

Value

A list defining the M-spline, with any omitted list components set to defaults. See `mspline_init` for details. The parameters are included as the `coefs` and `hscale` components.

`mspline_basis` *Evaluate an M-spline basis matrix at the specified times.*

Description

Evaluate an M-spline basis matrix at the specified times. Extrapolation beyond the boundary knots is done by assuming that each basis term is constant beyond the boundary.

Usage

```
mspline_basis(times, knots, degree = 3, integrate = FALSE, bsmooth = TRUE)
```

Arguments

`times`	A numeric vector of times at which to evaluate the basis.
`knots`	Spline knots
`degree`	Spline degree
`integrate`	If `TRUE`, then the integrated M-spline (I-spline) basis is returned.
`bsmooth`	If `TRUE` then the function is constrained to also have zero derivative and second derivative at the boundary, which improves smoothness at the boundary.

Details

The lower boundary is fixed to zero, and each basis term is assumed to be zero at times less than zero, since these models are used for hazard functions in survival data.

Value

A two-dimensional array. Rows are the times, and columns are the basis terms.

References

The splines2 package is used.

`mspline_constant_coefs` *Determine M-spline basis coefficients which give a constant function.*

Description

This works by obtaining the coefficients of the corresponding B-spline basis, which are equal if the B-spline is a constant function.

Usage

```
mspline_constant_coefs(mspline, logit = FALSE)
```

Arguments

`mspline` A list with components `knots` (vector of knots), `degree` (polynomial degree) and `bsmooth` (logical for smoothness constraint at boundary), defining an M-spline configuration.

`logit` If `TRUE` then the multinomial logit transform of the coefficients is returned. This is a vector of length one less than the number of coefficients, with the rth element defined by $log(coefs[r+1]/coefs[1])$.

Value

A numeric vector of the coefficients.

References

Ramsay, J. O. (1988). Monotone regression splines in action. Statistical Science, 3(4): 425-441.

`mspline_init` *Create a default M-spline model structure*

Description

Create a default M-spline model structure

Usage

```
mspline_init(
  df = 10,
  degree = 3,
  bsmooth = TRUE,
  knots = NULL,
  bknot = 10,
  obstimes = NULL
)
```

Arguments

`df` Desired number of basis terms, or "degrees of freedom" in the spline. If `knots` is not supplied, the number of knots is then chosen to satisfy this.

`degree` Spline polynomial degree. Can only be changed from the default of 3 if `bsmooth` is `FALSE`.

`bsmooth` If `TRUE` then the function is constrained to also have zero derivative and second derivative at the boundary.

`knots` Vector of knot locations. If not supplied, `df` has to be specified. One of two rules is then used to choose the knot locations. If `bknot` is specified, a set of equally spaced knots between zero and `bknot` is used. Otherwise if `obstimes` is supplied, the knots are chosen as equally spaced quantiles of `obstimes`.

The number of knots (excluding zero) is `df - degree + 1` if `bsmooth` is `TRUE`, or `df - degree - 1` otherwise.

`bknot` Location of the final spline knot.

`obstimes` Vector of observation times whose quantiles will be used to choose knot locations

Value

A list with fundamental components `knots`, `degree`, and `bsmooth`, as documented above.

The component `df` is also included, and derived as a consequence of the fundamental components.

`basis_means` gives the "mean" of each basis term (i.e. the mean of a random variable whose probability density function is given by the basis function)

`basis_spans` and `sqrt_wt` are quantities used in the construction of random walk prior distributions for the basis coefficients (following `https://arxiv.org/abs/2401.12640` and `https://arxiv.org/abs/2201.06808`).

`mspline_list_init` *Validate an M-spline object supplied as a list, choosing defaults if needed.*

Description

Validate an M-spline object supplied as a list, choosing defaults if needed.

Usage

```
mspline_list_init(mspline, obstimes = NULL)
```

Arguments

`mspline` A list with any or none of the following components: `df`, `degree`, `bsmooth`, `knots`, `bknot`, as documented in `mspline_init`.

`obstimes` Vector of observation times whose quantiles will be used to choose knot locations

Value

A list defining the M-spline, with any omitted list components set to defaults. See `mspline_init` for details.

If `mspline$knots` is not supplied, giving knot locations, then either `mspline$bknot` or `obstimes` must be specified, so that default locations can be obtained.

`mspline_plotdata` *Data for plotting an M-spline function, showing how it is built up from its basis*

Description

Data for plotting an M-spline function, showing how it is built up from its basis

Usage

```
mspline_plotdata(
  knots = NULL,
  bknot = 10,
  df = 10,
  degree = 3,
  bsmooth = TRUE,
  coefs = NULL,
  scale = 1,
  tmin = 0,
  tmax = 10
)
```

Arguments

`knots` Vector of knot locations. If not supplied, `df` has to be specified. One of two rules is then used to choose the knot locations. If `bknot` is specified, a set of equally spaced knots between zero and `bknot` is used. Otherwise if `obstimes` is supplied, the knots are chosen as equally spaced quantiles of `obstimes`.

The number of knots (excluding zero) is `df - degree + 1` if `bsmooth` is `TRUE`, or `df - degree - 1` otherwise.

`bknot` Location of the final spline knot.

`df` Desired number of basis terms, or "degrees of freedom" in the spline. If `knots` is not supplied, the number of knots is then chosen to satisfy this.

`degree` Spline polynomial degree. Can only be changed from the default of 3 if `bsmooth` is `FALSE`.

`bsmooth` If `TRUE` then the function is constrained to also have zero derivative and second derivative at the boundary.

`coefs` Coefficients of the spline basis terms. These are normalised internally to sum to 1, if they do not already sum to 1.

`scale` Scale parameter. After computing the standard M-spline function as a weighted sum of the basis terms, the function is multiplied by `scale`. The log of the scale is the parameter called `alpha` in the results of a `survextrap` model, the intercept of the linear model on the log hazard.

`tmin` Minimum plotting time. Defaults to zero.

`tmax` Maximum plotting time. Defaults to the highest knot.

Value

A data frame with values of the hazard at different times for different basis functions.

`mspline_plotsetup` *Get basis for an illustration of an M-spline with given knots.*

Description

Get basis for an illustration of an M-spline with given knots.

Usage

```
mspline_plotsetup(
  knots,
  bknot = 10,
  tmin = NULL,
  tmax = NULL,
  degree = 3,
  df = 10,
  bsmooth = TRUE
)
```

Arguments

knots	Vector of knot locations. If not supplied, df has to be specified. One of two rules is then used to choose the knot locations. If bknot is specified, a set of equally spaced knots between zero and bknot is used. Otherwise if obstimes is supplied, the knots are chosen as equally spaced quantiles of obstimes. The number of knots (excluding zero) is df - degree + 1 if bsmooth is TRUE, or df - degree - 1 otherwise.
bknot	Location of the final spline knot.
tmin	Minimum plotting time. Defaults to zero.
tmax	Maximum plotting time. Defaults to the highest knot.
degree	Spline polynomial degree. Can only be changed from the default of 3 if bsmooth is FALSE.
df	Desired number of basis terms, or "degrees of freedom" in the spline. If knots is not supplied, the number of knots is then chosen to satisfy this.
bsmooth	If TRUE then the function is constrained to also have zero derivative and second derivative at the boundary.

Value

Data frame containing the basis, as returned by mspline_basis.

mspline_spec *Make default M-spline knot specification given a survival dataset.*

Description

Choose default M-spline knot locations given a dataset and desired number of spline parameters. Assumes a cubic spline, and knots based on quantiles of event times observed in the individual data.

Usage

```
mspline_spec(
  formula,
  data,
  cure = FALSE,
  nonprop = NULL,
  backhaz = NULL,
  backhaz_strata = NULL,
  external = NULL,
  df = 10,
  add_knots = NULL,
  degree = 3,
  bsmooth = TRUE
)
```

Arguments

`formula` A survival formula in standard R formula syntax, with a call to `Surv()` on the left hand side.

Covariates included on the right hand side of the formula with be modelled with proportional hazards, or if `nonprop` is `TRUE` then a non-proportional hazards is used.

If `data` is omitted, so that the model is being fitted to external aggregate data alone, without individual data, then the formula should not include a `Surv()` call. The left-hand side of the formula will then be empty, and the right hand side specifies the covariates as usual. For example, `formula = ~1` if there are no covariates.

`data` Data frame containing variables in `formula`. Variables should be in a data frame, and not in the working environment.

This may be omitted, in which case `external` must be supplied. This allows a model to be fitted to external aggregate data alone, without any individual-level data.

`cure` If `TRUE`, a mixture cure model is used, where the "uncured" survival is defined by the M-spline model, and the cure probability is estimated.

`nonprop` Non-proportional hazards model specification. This is achieved by modelling the spline basis coefficients in terms of the covariates. See the methods vignette for more details.

If `TRUE`, then all covariates are modelled with non-proportional hazards, using the same model formula as `formula`.

If this is a formula, then this is assumed to define a model for the dependence of the basis coefficients on the covariates.

IF this is `NULL` or `FALSE` (the default) then any covariates are modelled with proportional hazards.

`backhaz` Background hazard, that is, for causes of death other than the cause of interest. This defines a "relative survival" or "additive hazards" model. The overall hazard that describes the all-cause survival data (given in the `data` and/or `external` argument) is then modelled as the sum of a cause-specific hazard and a background hazard.

The background hazard is assumed to be known, and the cause-specific hazard is modelled with the flexible parametric model.

The background hazard can be supplied in two forms. The meaning of predictions from the model depends on which of these is used.

(a) A data frame with columns `"hazard"` and `"time"`, specifying the background hazard at all times as a piecewise-constant (step) function. Each row gives the background hazard between the specified time and the next time. The first element of `"time"` should be 0, and the final row specifies the hazard at all times greater than the last element of `"time"`. Predictions from the model fitted by `survextrap` will then include this background hazard, because it is known at all times.

(b) The (quoted) name of a variable in the data giving the background hazard. For censored cases, the exact value does not matter. The predictions from `survextrap` will then describe the excess hazard or survival on top of this background. The overall hazard cannot be predicted in general, because the background hazard is only specified over a limited range of time.

If there is external data, and `backhaz` is supplied in form (b), then the user should also supply the background survival at the start and stop points in columns of the external data named `"backsurv_start"` and `"backsurv_stop"`. That is, the probability of survival up to each of these times for someone alive at time 0. This should describe the same reference population as `backhaz`, though the package does not check for consistency between these.

If there are stratifying variables specified in `backhaz_strata`, then there should be multiple rows giving the background hazard value for each time period and stratifying variable.

If `backhaz` is `NULL` (the default) then no background hazard component is included in the model.

`backhaz_strata` A character vector of names of variables that appear in `backhaz` that indicate strata, e.g. `backhaz_strata = c("agegroup","sex")`. This allows different background hazard values to be used for different subgroups. These variables must also appear in the datasets being modelled, that is, in `data`, `external` or both. Each row of those datasets should then have a corresponding row in `backhaz` which has the same values of the stratifying variables.

This is `NULL` by default, indicating no stratification of the background hazard.

If stratification is done, then `backhaz` must be supplied in form (a), as a data frame rather than a variable in the data.

`external` External data as a data frame of aggregate survival counts with columns named:

`start`: Start time

`stop`: Follow-up time

`n`: Number of people alive at `start`

`r`: Number of those people who are still alive at `stop`

If there are covariates in `formula`, then the values they take in the external data must be supplied as additional columns in `external`. Therefore if there are external data, the covariates in `formula` and `data` should not be named `start`,`stop`,`n` or `r`.

`df` Desired number of basis terms, or "degrees of freedom" in the spline. If `knots` is not supplied, the number of knots is then chosen to satisfy this.

`add_knots` Additional knots, other than those determined from the quantiles of the individual data. Typically used to add a maximum knot at the time that we want to extrapolate to.

`degree` Spline polynomial degree. Can only be changed from the default of 3 if `bsmooth` is `FALSE`.

`bsmooth` If `TRUE` then the function is constrained to also have zero derivative and second derivative at the boundary.

Details

If there are also external data, then these are based on quantiles of a vector defined by concatenating the event times in the individual data with the unique start and stop times in the external data.

This is designed to have the same arguments as `survextrap`. It is intended for use when we want to fit a set of `survextrap` models with the same spline specification.

See also `mspline_list_init` and `mspline_init`, which have lower-level interfaces, and are designed for use without data, e.g. when illustrating a theoretical M-spline model.

Value

A list with components

`knots` Knot locations. The number of knots will be equal to `df + degree + 2`. `degree` Spline polynomial degree (i.e. 3) `nvars` Number of basis variables (an alias for `df`)

`plot.survextrap` *Plot method for survextrap model objects*

Description

Intended as a quick check of the results of a model fit. See `plot_survival` and `plot_hazard` for the functions behind each panel, and `survival` and `hazard` to extract the data being plotted here to enable custom plots like these.

Usage

```
## S3 method for class 'survextrap'
plot(x, type = "hazsurv", newdata = NULL, ...)
```

Arguments

`x` A fitted model object as returned by `survextrap`

`type` `"survival"` for a plot of the survival function, `"hazard"` for the hazard function, against time.

`newdata` Data frame of covariate values to compute the output for. If there are covariates in the model and this is not supplied, the following default is used:

(a) if the only covariate is one factor variable, then the output is computed for each level of this factor.

(b) if there are multiple covariates, or any numeric covariates, then the output is computed at the mean of each numeric covariate in the original data, and at the baseline level of each factor covariate.

Note: caution is required about how treatment groups (for example) are stored in your data. If these are coded as numeric (0/1), then if `newdata` is not specified, only one output will be shown. This relates to the average value of this numeric variable over the data, which doesn't correspond to either of the treatment groups. To avoid this, a treatment group should be stored as a factor.

`...` Additional arguments, passed on to `plot_hazard` and `plot_survival`.

Value

A `ggplot2` plot object.

`plot_hazard` *Plot hazard curves from a survextrap model*

Description

Plot hazard curves from a `survextrap` model. This function is intended as a quick check of a fitted model, so it deliberately has limited options for customisation. The data behind these plots can be extracted with `hazard` into a tidy data frame to enable custom plots to be constructed with **ggplot2**. See the case study vignette for some examples.

Usage

```
plot_hazard(
  x,
  newdata = NULL,
  t = NULL,
  tmax = NULL,
  niter = NULL,
  newdata0 = NULL,
  wane_period = NULL,
  wane_nt = 10,
  ci = NULL,
  xlab = "Time",
  ylab = "Hazard",
  line_size = 1.5,
  ci_alpha = 0.2,
  show_knots = FALSE
)
```

Arguments

`x` A fitted model object as returned by `survextrap`

`newdata` Data frame of covariate values to compute the output for. If there are covariates in the model and this is not supplied, the following default is used:

(a) if the only covariate is one factor variable, then the output is computed for each level of this factor.

(b) if there are multiple covariates, or any numeric covariates, then the output is computed at the mean of each numeric covariate in the original data, and at the baseline level of each factor covariate.

Note: caution is required about how treatment groups (for example) are stored in your data. If these are coded as numeric (0/1), then if `newdata` is not specified, only one output will be shown. This relates to the average value of this numeric variable over the data, which doesn't correspond to either of the treatment groups. To avoid this, a treatment group should be stored as a factor.

`t` Vector of times at which to compute the estimates.

`tmax` Maximum time at which to compute the estimates. If `t` is supplied, then this is ignored. If `t` is not supplied, then `t` is set to a set of 100 equally spaced time points from 0 to `tmax`. If both `tmax` and `t` are not supplied, then `tmax` is set to the maximum follow up time in the data.

`niter` Number of MCMC iterations to use to compute credible intervals. Set to a low value to make this function quicker, at the cost of some approximation error (which may not be important for plotting or model development).

`newdata0` Data frame of covariate values defining the "untreated" group for use in treatment waning models. See `Survmspline_wane`.

`wane_period` Vector of two numbers, defining the time period over which the hazard is interpolated between the hazard of the "treated" group (taken from `newdata`) and the hazard of the "untreated" group (taken from `newdata0`). Optional - if this is not supplied, then no waning is assumed.

`wane_nt` Number of intervals defining the piecewise constant approximation to the hazard during the waning period.

`ci` If `TRUE` then credible intervals are drawn. Defaults to drawing the intervals if the plot shows the curve for only one covariate value.

`xlab` X-axis label

ylab	Y-axis label
line_size	Passed to geom_line
ci_alpha	Transparency for the credible interval ribbons
show_knots	Show the locations of the spline knots as vertical lines

Details

If the model has a single factor covariate (excluding background hazard strata), then curves are produced for each level of this factor if newdata requests this (or is left to its default). Otherwise, only a single curve is produced, illustrating the corresponding output from hazard.

Value

A ggplot2 plot object.

plot_hazard_ratio *Plot hazard ratio against time from a survextrap model*

Description

For use with non-proportional hazards models (survextrap(...,nonprop=TRUE)). Intended as a quick check of a model fit, so there are limited customisation options. The underlying data can be extracted with hazard_ratio.

Usage

```
plot_hazard_ratio(
  x,
  newdata = NULL,
  t = NULL,
  tmax = NULL,
  niter = NULL,
  ci = TRUE,
  xlab = "Time",
  ylab = "Hazard ratio",
  line_size = 1.5,
  ci_alpha = 0.2
)
```

Arguments

x	A fitted model object as returned by survextrap
newdata	A data frame with two rows. The hazard ratio will be defined as hazard(second row) divided by hazard(first row). If the only covariate in the model is a factor with two levels, then newdata defaults to a data frame containing the levels of this factor, so that the hazard ratio for the second level versus the first level is computed. For any other models, newdata must be supplied explicitly.

Standardisation (with standardise_to) is not supported. This might be done by hand by using `hazard(...,sample=TRUE)` to obtain posterior samples for the two standardised hazards separately, then summarising by hand.

`t` Vector of times at which to compute the estimates.

`tmax` Maximum time at which to compute the estimates. If `t` is supplied, then this is ignored. If `t` is not supplied, then `t` is set to a set of 100 equally spaced time points from 0 to `tmax`. If both `tmax` and `t` are not supplied, then `tmax` is set to the maximum follow up time in the data.

`niter` Number of MCMC iterations to use to compute credible intervals. Set to a low value to make this function quicker, at the cost of some approximation error (which may not be important for plotting or model development).

`ci` If `TRUE` then credible intervals are drawn. Defaults to drawing the intervals if the plot shows the curve for only one covariate value.

`xlab` X-axis label

`ylab` Y-axis label

`line_size` Passed to geom_line

`ci_alpha` Transparency for the credible interval ribbons

Value

A `ggplot2` plot object.

`plot_mspline` *Plot a M-spline function, showing how it is built up from its basis*

Description

See mspline_plotdata for the data behind the plot

Usage

```
plot_mspline(
  knots = NULL,
  bknot = 10,
  df = 10,
  degree = 3,
  bsmooth = TRUE,
  coefs = NULL,
  scale = 1,
  tmin = 0,
  tmax = 10,
  show_knots = FALSE,
  show_means = FALSE
)
```

Arguments

`knots` Vector of knot locations. If not supplied, `df` has to be specified. One of two rules is then used to choose the knot locations. If `bknot` is specified, a set of equally spaced knots between zero and `bknot` is used. Otherwise if `obstimes` is supplied, the knots are chosen as equally spaced quantiles of `obstimes`.

The number of knots (excluding zero) is `df - degree + 1` if `bsmooth` is `TRUE`, or `df - degree - 1` otherwise.

`bknot` Location of the final spline knot.

`df` Desired number of basis terms, or "degrees of freedom" in the spline. If `knots` is not supplied, the number of knots is then chosen to satisfy this.

`degree` Spline polynomial degree. Can only be changed from the default of 3 if `bsmooth` is `FALSE`.

`bsmooth` If `TRUE` then the function is constrained to also have zero derivative and second derivative at the boundary.

`coefs` Coefficients of the spline basis terms. These are normalised internally to sum to 1, if they do not already sum to 1.

`scale` Scale parameter. After computing the standard M-spline function as a weighted sum of the basis terms, the function is multiplied by `scale`. The log of the scale is the parameter called `alpha` in the results of a `survextrap` model, the intercept of the linear model on the log hazard.

`tmin` Minimum plotting time. Defaults to zero.

`tmax` Maximum plotting time. Defaults to the highest knot.

`show_knots` Show the positions of the knots, including the upper boundary

`show_means` Show the "mean" around which each basis term is centred (defined as the mean of a random variable whose PDF is defined by the basis term).

Value

A `ggplot2` plot object.

`plot_survival` *Plot survival curves from a survextrap model*

Description

Plot survival curves from a `survextrap` model. This function is intended as a quick check of a fitted model, so it deliberately has limited options for customisation. The data behind these plots can be extracted with `survival` into a tidy data frame to enable custom plots to be constructed with **ggplot2**. See the case study vignette for some examples.

Usage

```
plot_survival(
  x,
  newdata = NULL,
  t = NULL,
  tmax = NULL,
  km = NULL,
  niter = NULL,
  newdata0 = NULL,
  wane_period = NULL,
  wane_nt = 10,
  ci = NULL,
  xlab = "Time",
  ylab = "Survival",
  line_size = 1.5,
  ci_alpha = 0.2,
  show_knots = FALSE
)
```

Arguments

`x` A fitted model object as returned by `survextrap`

`newdata` Data frame of covariate values to compute the output for. If there are covariates in the model and this is not supplied, the following default is used:

(a) if the only covariate is one factor variable, then the output is computed for each level of this factor.

(b) if there are multiple covariates, or any numeric covariates, then the output is computed at the mean of each numeric covariate in the original data, and at the baseline level of each factor covariate.

Note: caution is required about how treatment groups (for example) are stored in your data. If these are coded as numeric (0/1), then if `newdata` is not specified, only one output will be shown. This relates to the average value of this numeric variable over the data, which doesn't correspond to either of the treatment groups. To avoid this, a treatment group should be stored as a factor.

`t` Vector of times at which to compute the estimates.

`tmax` Maximum time at which to compute the estimates. If `t` is supplied, then this is ignored. If `t` is not supplied, then `t` is set to a set of 100 equally spaced time points from 0 to `tmax`. If both `tmax` and `t` are not supplied, then `tmax` is set to the maximum follow up time in the data.

`km` If `TRUE` then a Kaplan-Meier curve of the observed data is plotted, using the results of `survival::survfit()` on the formula originally used for the `survextrap` fit. By default, this is only done when there are no covariates or one factor covariate.

The Kaplan-Meier estimates are returned in the `km` component of the fitted model object returned by `survextrap`, for use in hand-crafted plots like these.

niter Number of MCMC iterations to use to compute credible intervals. Set to a low value to make this function quicker, at the cost of some approximation error (which may not be important for plotting or model development).

newdata0 Data frame of covariate values defining the "untreated" group for use in treatment waning models. See Survmspline_wane.

wane_period Vector of two numbers, defining the time period over which the hazard is interpolated between the hazard of the "treated" group (taken from newdata) and the hazard of the "untreated" group (taken from newdata0). Optional - if this is not supplied, then no waning is assumed.

wane_nt Number of intervals defining the piecewise constant approximation to the hazard during the waning period.

ci If TRUE then credible intervals are drawn. Defaults to drawing the intervals if the plot shows the curve for only one covariate value.

xlab X-axis label

ylab Y-axis label

line_size Passed to geom_line

ci_alpha Transparency for the credible interval ribbons

show_knots Show the locations of the spline knots as vertical lines

Details

If the model has a single factor covariate (excluding background hazard strata), then curves are produced for each level of this factor if newdata requests this (or is left to its default). Otherwise, only a single curve is produced, illustrating the corresponding output from hazard.

Value

A ggplot2 plot object.

print.survextrap *Print a fitted survextrap model*

Description

Print a fitted survextrap model

Usage

```
## S3 method for class 'survextrap'
print(x, ...)
```

Arguments

x A fitted model object as returned by survextrap

... Other arguments (currently unused).

Value

No value is returned.

This prints a summary of the data, a statement of the fitted model, the prior distributions, and a table of summary statistics of the posterior distributions of the model parameters. For descriptions of the parameters in this summary table, see `summary.survextrap`.

`print_priors` *Print the priors used in a fitted survextrap model*

Description

Print the priors used in a fitted survextrap model

Usage

```
print_priors(x)
```

Arguments

`x`	A fitted model object as returned by `survextrap`

Value

No explicit value is returned.

`priors` *Prior distributions and options*

Description

The functions described on this page are used to specify the prior distributions for the parameters in a survextrap model.

Usage

```
p_normal(location = 0, scale = 2.5)

p_t(location = 0, scale = 2.5, df = 1)

p_beta(shape1 = 1, shape2 = 1)

p_gamma(shape = 2, rate = 1)
```

Arguments

location	Prior location. For the normal distribution, this is the mean. Defaults to 0
scale	Prior scale. For the normal distribution, this is the standard deviation. Defaults to 2.5.
df	Prior degrees of freedom (only for Student t distribution).
shape1	First shape parameter (for Beta distribution, defaults to 1).
shape2	Second shape parameter (for Beta distribution, defaults to 1).
shape	Shape parameter (for Gamma distribution, defaults to 2).
rate	Rate parameter (for Gamma distribution, defaults to 1).

Value

A named list with the prior distribution name and parameters, to be used internally by the survextrap model fitting functions.

See Also

survextrap.

prior_haz *Determine priors for time-varying hazards and hazard ratios*

Description

Computes consequences of priors chosen for the parameters hsd and hrsd in a flexible hazard model survextrap on an interpretable scale. This can be used to calibrate Gamma priors for these parameters to match interpretable beliefs.

Usage

```
prior_haz_sd(
  mspline,
  coefs_mean = NULL,
  prior_hsd = p_gamma(2, 1),
  prior_hscale = p_normal(0, 20),
  smooth_model = "exchangeable",
  prior_loghr = NULL,
  formula = NULL,
  cure = NULL,
  nonprop = NULL,
  newdata = NULL,
  prior_hrsd = NULL,
  tmin = 0,
  tmax = NULL,
  nsim = 1000,
  hq = c(0.1, 0.9),
  quantiles = c(0.025, 0.5, 0.975)
)

prior_hr_sd(
  mspline,
  coefs_mean = NULL,
  prior_hsd = p_gamma(2, 1),
  prior_hscale = p_normal(0, 20),
  smooth_model = "exchangeable",
  prior_loghr = NULL,
  formula = NULL,
  cure = NULL,
  nonprop = NULL,
  newdata = NULL,
  newdata0 = NULL,
  prior_hrsd = NULL,
  tmin = 0,
  tmax = 10,
  nsim = 100,
  hq = c(0.1, 0.9),
  quantiles = c(0.025, 0.5, 0.975)
)
```

Arguments

mspline A list of control parameters defining the spline model.

knots: Spline knots. If this is not supplied, then the number of knots is taken from df, and their location is taken from equally-spaced quantiles of the observed event times in the individual-level data.

add_knots: This is intended to be used when there are external data included in the model. External data are typically outside the time period covered by the individual data. add_knots would then be chosen to span the time period covered by the external data, so that the hazard trajectory can vary over that time.

If there are external data, and both knots and add_knots are omitted, then a default set of knots is chosen to span both the individual and external data, by taking the quantiles of a vector defined by concatenating the individual-level event times with the start and stop times in the external data.

df: Degrees of freedom, i.e. the number of parameters (or basis terms) intended to result from choosing knots based on quantiles of the data. The total number of parameters will then be df plus the number of additional knots specified in add_knots. df defaults to 10. This does not necessarily overfit, because the function is smoothed through the prior.

degree: Polynomial degree used for the basis function. The default is 3, giving a cubic. This can only be changed from 3 if bsmooth is FALSE.

bsmooth: If TRUE (on by default) the spline is smoother at the highest knot, by defining the derivative and second derivative at this point to be zero.

`coefs_mean` Spline basis coefficients that define the prior mean for the hazard function. By default, these are set to values that define a constant hazard function (see `mspline_constant_coefs`). They are normalised to sum to 1 internally (if they do not already).

`prior_hsd` Gamma prior for the standard deviation that controls the variability over time (or smoothness) of the hazard function. This should be a call to `p_gamma()`. The default is `p_gamma(2,1)`. See `prior_haz_sd` for a way to calibrate this to represent a meaningful belief.

`prior_hscale` Prior for the baseline log hazard scale parameter (`alpha` or `log(eta)`). This should be a call to a prior constructor function, such as `p_normal(0,1)` or `p_t(0,2,2)`. Supported prior distribution families are normal (parameters mean and SD) and t distributions (parameters location, scale and degrees of freedom). The default is a normal distribution with mean 0 and standard deviation 20.

Note that `eta` is not in itself a hazard, but it is proportional to the hazard (see the vignette for the full model specification).

"Baseline" is defined by the continuous covariates taking a value of zero and factor covariates taking their reference level. To use a different baseline, the data should be transformed appropriately beforehand, so that a value of zero has a different meaning. For continuous covariates, it helps for both computation and interpretation to define the value of zero to denote a typical value in the data, e.g. the mean.

`smooth_model` The default `"random_walk"`, specifies a random walk prior for the multinomial-logit spline coefficients, based on logistic distributions. See the methods vignette for full details.

The alternative `"exchangeable"` uses independent logistic priors on the multinomial-logit spline coefficients, conditionally on a common smoothing variance parameter. Note this is the method explained in the original survextrap paper (Jackson, BMC Med Res 2023). The random walk model is shown to perform better in Timmins et al (2025).

In non-proportional hazards models, setting `smooth_model` also determines whether an exchangeable or random walk model is used for the non-proportionality parameters (δ).

`prior_loghr` Priors for log hazard ratios. This should be a call to `p_normal()` or `p_t()`. A list of calls can also be provided, to give different priors to different coefficients, where the name of each list component matches the name of the coefficient, e.g. `list("age45-59" = p_normal(0,1), "age60+" = p_t(0,2,3))`

The default is `p_normal(0,2.5)` for all coefficients.

`formula` A survival formula in standard R formula syntax, with a call to `Surv()` on the left hand side.

Covariates included on the right hand side of the formula with be modelled with proportional hazards, or if `nonprop` is `TRUE` then a non-proportional hazards is used.

If `data` is omitted, so that the model is being fitted to external aggregate data alone, without individual data, then the formula should not include a `Surv()` call. The left-hand side of the formula will then be empty, and the right hand side specifies the covariates as usual. For example, `formula = ~1` if there are no covariates.

`cure` If `TRUE`, a mixture cure model is used, where the "uncured" survival is defined by the M-spline model, and the cure probability is estimated.

`nonprop` Non-proportional hazards model specification. This is achieved by modelling the spline basis coefficients in terms of the covariates. See the methods vignette for more details.

If `TRUE`, then all covariates are modelled with non-proportional hazards, using the same model formula as `formula`.

If this is a formula, then this is assumed to define a model for the dependence of the basis coefficients on the covariates.

IF this is `NULL` or `FALSE` (the default) then any covariates are modelled with proportional hazards.

`newdata` A data frame with one row, containing variables in the model formulae. Samples will then be drawn, for any covariate-dependent parameters, with covariates set to the values given here.

`prior_hrsd` Prior for the standard deviation parameters that smooth the non-proportionality effects over time in non-proportional hazards models. This should be a call to `p_gamma()` or a list of calls to `p_gamma()` with one component per covariate, as in `prior_loghr`. See `prior_hr_sd` for a way to calibrate this to represent a meaningful belief.

`tmin` Minimum plotting time. Defaults to zero.

`tmax` Maximum plotting time. Defaults to the highest knot.

`nsim` Number of simulations to draw

`hq` Quantiles which define the "low" and "high" values of a time-varying quantity (hazard in `prior_haz_sd` and the hazard ratio in `prior_hr_sd`). The ratio between the high and low values will be summarised, as a measure of time-dependence. By default, this is `c(0.1, 0.9)`, so that the 10% and 90% quantiles are used respectively.

`quantiles` Quantiles used to summarise the implied prior distributions of the simulated quantities.

`newdata0` A data frame with one row, containing "reference" values of variables in the model formulae. The hazard ratio between the hazards at `newdata` and `newdata0` will be returned.

Details

The spline model in `survextrap` allows the hazard to change over time in an arbitrarily flexible manner. The prior distributions on the parameters of this model have implications for how much we expect the hazard to plausibly vary over time. These priors are hard to interpret directly, but this function can be used to compute their implications on a more easily-understandable scale.

This is done by:

(1) simulating a set of parameters from their prior distributions

(2) computing the hazard at a fine grid of equally-spaced points spanning the boundary knots

(3) calculating the empirical standard deviation of the set of hazards at these points

(4) repeatedly performing steps 1-3, and summarising the distribution of the resulting standard deviations. This is the implied prior for the hazard variability.

`prior_haz_sd` computes the SD of the hazard, and the SD of the inverse hazard is also computed. The inverse hazard at time `t` is the expected time to the event given survival to `t`. The hazard ratio between a high and low value (defined by quantiles of values at different times) is also computed.

`prior_hr_sd` computes the SD of the hazard ratio between two covariate values supplied by the user.

All of these SDs refer to the variability over time, e.g. a SD of 0 indicates that the hazard (or inverse hazard, or hazard ratio) is constant with time.

Value

A data frame with columns `sd_haz` (SD of the hazard), `sd_mean` (SD of the inverse hazard) and `hr` (ratio between high/low hazards) (for `prior_haz_sd`), and rows giving prior quantiles of these.

In `prior_hr_sd`, `sd_hr` is the SD of hazard ratios over time, and `hrr` is the ratio between high/low hazard ratios.

`prior_haz_const` *Summarises the prior for the constant hazard implied by a particular prior on the hazard scale parameter and spline specification.*

Description

Summarises the prior for the constant hazard implied by a particular prior on the hazard scale parameter and M-spline specification, when the spline coefficients are fixed to define a constant hazard using `mspline_constant_coefs`.

Usage

```
prior_haz_const(
  mspline,
  prior_hscale = p_normal(0, 20),
  quantiles = c(0.025, 0.5, 0.975)
)
```

Arguments

`mspline` A list of control parameters defining the spline model.

`knots`: Spline knots. If this is not supplied, then the number of knots is taken from `df`, and their location is taken from equally-spaced quantiles of the observed event times in the individual-level data.

`add_knots`: This is intended to be used when there are `external` data included in the model. External data are typically outside the time period covered by the individual data. `add_knots` would then be chosen to span the time period covered by the external data, so that the hazard trajectory can vary over that time.

If there are external data, and both `knots` and `add_knots` are omitted, then a default set of knots is chosen to span both the individual and external data, by taking the quantiles of a vector defined by concatenating the individual-level event times with the `start` and `stop` times in the external data.

`df`: Degrees of freedom, i.e. the number of parameters (or basis terms) intended to result from choosing knots based on quantiles of the data. The total number of parameters will then be `df` plus the number of additional knots specified in `add_knots`. `df` defaults to 10. This does not necessarily overfit, because the function is smoothed through the prior.

`degree`: Polynomial degree used for the basis function. The default is 3, giving a cubic. This can only be changed from 3 if `bsmooth` is `FALSE`.

`bsmooth`: If `TRUE` (on by default) the spline is smoother at the highest knot, by defining the derivative and second derivative at this point to be zero.

`prior_hscale` Prior for the baseline log hazard scale parameter (`alpha` or `log(eta)`). This should be a call to a prior constructor function, such as `p_normal(0,1)` or `p_t(0,2,2)`. Supported prior distribution families are normal (parameters mean and SD) and t distributions (parameters location, scale and degrees of freedom). The default is a normal distribution with mean 0 and standard deviation 20.

Note that `eta` is not in itself a hazard, but it is proportional to the hazard (see the vignette for the full model specification).

"Baseline" is defined by the continuous covariates taking a value of zero and factor covariates taking their reference level. To use a different baseline, the data should be transformed appropriately beforehand, so that a value of zero has a different meaning. For continuous covariates, it helps for both computation and interpretation to define the value of zero to denote a typical value in the data, e.g. the mean.

`quantiles` Quantiles used to summarise the implied prior distributions of the simulated quantities.

Value

A data frame with columns `haz` for the hazard and `mean` for the inverse hazard (on the mean survival scale) and rows for three summary quantiles.

See Also

`p_meansurv`, `mspline_constant_coefs`

`prior_hr` *Summarises the prior for the hazard ratio implied by a particular prior on the log hazard ratio*

Description

Summarises the prior for the hazard ratio implied by a particular prior on the log hazard ratio. Simply applies an exponential transform to quantiles of the given prior.

Usage

```
prior_hr(prior_loghr = p_normal(0, 2.5), quantiles = c(0.025, 0.5, 0.975))
```

Arguments

`prior_loghr` Priors for log hazard ratios. This should be a call to `p_normal()` or `p_t()`. A list of calls can also be provided, to give different priors to different coefficients, where the name of each list component matches the name of the coefficient, e.g. `list("age45-59" = p_normal(0,1), "age60+" = p_t(0,2,3))`

The default is `p_normal(0,2.5)` for all coefficients.

`quantiles` Quantiles used to summarise the implied prior distributions of the simulated quantities.

Value

A vector of quantiles.

---

prior_pred *Simulate a dataset from the prior predictive distribution of survival times in an M-spline survival model.*

Description

Simulate a dataset from the prior predictive distribution of survival times in an M-spline survival model. Additive hazards models not currently supported.

Usage

```
prior_pred(
  n,
  fix_prior = FALSE,
  mspline,
  censtime = Inf,
  coefs_mean = NULL,
  prior_hscale = p_normal(0, 20),
  prior_hsd = p_gamma(2, 1),
  newdata = NULL,
  formula = NULL,
  prior_loghr = NULL,
  prior_hrsd = NULL,
  prior_cure = NULL
)
```

Arguments

`n` Sample size of the simulated dataset. Each observation in the dataset is generated from a model with the same parameters. These parameters are generated from a single simulation from the prior distribution.

`fix_prior` If `TRUE`, then one value of the parameter vector is drawn from the prior, followed by `n` individual-level times given this common prior value. If `FALSE`, then to produce each sampled individual time, a different sample from the prior is used.

`mspline` A list of control parameters defining the spline model.

`knots`: Spline knots. If this is not supplied, then the number of knots is taken from `df`, and their location is taken from equally-spaced quantiles of the observed event times in the individual-level data.

`add_knots`: This is intended to be used when there are `external` data included in the model. External data are typically outside the time period covered by the individual data. `add_knots` would then be chosen to span the time period covered by the external data, so that the hazard trajectory can vary over that time.

If there are external data, and both `knots` and `add_knots` are omitted, then a default set of knots is chosen to span both the individual and external data, by taking the quantiles of a vector defined by concatenating the individual-level event times with the `start` and `stop` times in the external data.

`df`: Degrees of freedom, i.e. the number of parameters (or basis terms) intended to result from choosing knots based on quantiles of the data. The total number of parameters will then be `df` plus the number of additional knots specified in `add_knots`. `df` defaults to 10. This does not necessarily overfit, because the function is smoothed through the prior.

`degree`: Polynomial degree used for the basis function. The default is 3, giving a cubic. This can only be changed from 3 if `bsmooth` is `FALSE`.

`bsmooth`: If `TRUE` (on by default) the spline is smoother at the highest knot, by defining the derivative and second derivative at this point to be zero.

`censtime` Right-censoring time to impose on the simulated event times.

`coefs_mean` Spline basis coefficients that define the prior mean for the hazard function. By default, these are set to values that define a constant hazard function (see `mspline_constant_coefs`). They are normalised to sum to 1 internally (if they do not already).

`prior_hscale` Prior for the baseline log hazard scale parameter (`alpha` or `log(eta)`). This should be a call to a prior constructor function, such as `p_normal(0,1)` or `p_t(0,2,2)`. Supported prior distribution families are normal (parameters mean and SD) and t distributions (parameters location, scale and degrees of freedom). The default is a normal distribution with mean 0 and standard deviation 20.

Note that `eta` is not in itself a hazard, but it is proportional to the hazard (see the vignette for the full model specification).

"Baseline" is defined by the continuous covariates taking a value of zero and factor covariates taking their reference level. To use a different baseline, the data should be transformed appropriately beforehand, so that a value of zero has a different meaning. For continuous covariates, it helps for both computation and interpretation to define the value of zero to denote a typical value in the data, e.g. the mean.

`prior_hsd` Gamma prior for the standard deviation that controls the variability over time (or smoothness) of the hazard function. This should be a call to `p_gamma()`. The default is `p_gamma(2,1)`. See `prior_haz_sd` for a way to calibrate this to represent a meaningful belief.

`newdata` A data frame with one row, containing variables in the model formulae. Samples will then be drawn, for any covariate-dependent parameters, with covariates set to the values given here.

`formula` A model formula with no response, defining the covariates on the hazard scale.

`prior_loghr` Priors for log hazard ratios. This should be a call to `p_normal()` or `p_t()`. A list of calls can also be provided, to give different priors to different coefficients, where the name of each list component matches the name of the coefficient, e.g. `list("age45-59" = p_normal(0,1), "age60+" = p_t(0,2,3))`

The default is `p_normal(0,2.5)` for all coefficients.

`prior_hrsd` Prior for the standard deviation parameters that smooth the non-proportionality effects over time in non-proportional hazards models. This should be a call to `p_gamma()` or a list of calls to `p_gamma()` with one component per covariate, as in `prior_loghr`. See `prior_hr_sd` for a way to calibrate this to represent a meaningful belief.

`prior_cure` Prior for the baseline cure probability. This should be a call to `p_beta()`. The default is a uniform prior, `p_beta(1,1)`. Baseline is defined by the mean of continuous covariates and the reference level of factor covariates.

Value

A data frame with columns `time` (simulated time) and `event` (indicator for whether the time is an event time, as opposed to a right-censoring time). The prior parameters are returned in the `prior` attribute as a list with components `alpha` (baseline log hazard) and `coefs` (spline coefficients).

See Also

`prior_sample`

`prior_sample` *Sample from the joint prior of parameters in a survextrap model*

Description

Draws a sample from the joint prior distribution of the parameters governing a survextrap model for given covariates. This is used, for example, in `prior_sample_hazard`, to visualise the prior distribution around hazard curves implied by a particular M-spline model and parameter priors.

Usage

```
prior_sample(
  mspline,
  coefs_mean = NULL,
  prior_hsd = p_gamma(2, 1),
  prior_hscale,
  smooth_model = ”random_walk”,
  prior_loghr = NULL,
  formula = NULL,
  cure = NULL,
  nonprop = NULL,
  newdata = NULL,
  newdata0 = NULL,
  prior_hrsd = NULL,
  prior_cure = NULL,
  prior_logor_cure = NULL,
  nsim = 100
)
```

Arguments

`mspline` A list of control parameters defining the spline model.

`knots`: Spline knots. If this is not supplied, then the number of knots is taken from `df`, and their location is taken from equally-spaced quantiles of the observed event times in the individual-level data.

`add_knots`: This is intended to be used when there are `external` data included in the model. External data are typically outside the time period covered by the individual data. `add_knots` would then be chosen to span the time period covered by the external data, so that the hazard trajectory can vary over that time.

If there are external data, and both `knots` and `add_knots` are omitted, then a default set of knots is chosen to span both the individual and external data, by taking the quantiles of a vector defined by concatenating the individual-level event times with the `start` and `stop` times in the external data.

`df`: Degrees of freedom, i.e. the number of parameters (or basis terms) intended to result from choosing knots based on quantiles of the data. The total number of parameters will then be `df` plus the number of additional knots specified in `add_knots`. `df` defaults to 10. This does not necessarily overfit, because the function is smoothed through the prior.

`degree`: Polynomial degree used for the basis function. The default is 3, giving a cubic. This can only be changed from 3 if `bsmooth` is `FALSE`.

`bsmooth`: If `TRUE` (on by default) the spline is smoother at the highest knot, by defining the derivative and second derivative at this point to be zero.

`coefs_mean` Spline basis coefficients that define the prior mean for the hazard function. By default, these are set to values that define a constant hazard function (see `mspline_constant_coefs`). They are normalised to sum to 1 internally (if they do not already).

`prior_hsd` Gamma prior for the standard deviation that controls the variability over time (or smoothness) of the hazard function. This should be a call to `p_gamma()`. The default is `p_gamma(2,1)`. See `prior_haz_sd` for a way to calibrate this to represent a meaningful belief.

`prior_hscale` Prior for the baseline log hazard scale parameter (`alpha` or `log(eta)`). This should be a call to a prior constructor function, such as `p_normal(0,1)` or `p_t(0,2,2)`. Supported prior distribution families are normal (parameters mean and SD) and t distributions (parameters location, scale and degrees of freedom). The default is a normal distribution with mean 0 and standard deviation 20.

Note that `eta` is not in itself a hazard, but it is proportional to the hazard (see the vignette for the full model specification).

"Baseline" is defined by the continuous covariates taking a value of zero and factor covariates taking their reference level. To use a different baseline, the data should be transformed appropriately beforehand, so that a value of zero has a different meaning. For continuous covariates, it helps for both computation and interpretation to define the value of zero to denote a typical value in the data, e.g. the mean.

`smooth_model` The default `"random_walk"`, specifies a random walk prior for the multinomial-logit spline coefficients, based on logistic distributions. See the methods vignette for full details.

The alternative `"exchangeable"` uses independent logistic priors on the multinomial-logit spline coefficients, conditionally on a common smoothing variance parameter. Note this is the method explained in the original survextrap paper (Jackson, BMC Med Res 2023). The random walk model is shown to perform better in Timmins et al (2025).

In non-proportional hazards models, setting `smooth_model` also determines whether an exchangeable or random walk model is used for the non-proportionality parameters (δ).

`prior_loghr` Priors for log hazard ratios. This should be a call to `p_normal()` or `p_t()`. A list of calls can also be provided, to give different priors to different coefficients, where the name of each list component matches the name of the coefficient, e.g. `list("age45-59" = p_normal(0,1), "age60+" = p_t(0,2,3))`

The default is `p_normal(0,2.5)` for all coefficients.

`formula` A model formula with no response, defining the covariates on the hazard scale.

`cure` A model formula with no response, giving any covariates on the cure proportion.

`nonprop` A model formula with no response, defining any covariates affecting the spline basis coefficients, which gives a nonproportional hazards model.

`newdata` A data frame with one row, containing variables in the model formulae. Samples will then be drawn, for any covariate-dependent parameters, with covariates set to the values given here.

`newdata0` A data frame with one row, containing "reference" values of variables in the model formulae. The hazard ratio between the hazards at `newdata` and `newdata0` will be returned.

`prior_hrsd` Prior for the standard deviation parameters that smooth the non-proportionality effects over time in non-proportional hazards models. This should be a call to p_gamma() or a list of calls to p_gamma() with one component per covariate, as in prior_loghr. See prior_hr_sd for a way to calibrate this to represent a meaningful belief.

prior_cure — Prior for the baseline cure probability. This should be a call to p_beta(). The default is a uniform prior, p_beta(1,1). Baseline is defined by the mean of continuous covariates and the reference level of factor covariates.

prior_logor_cure — Priors for log odds ratios on cure probabilities. This should be a call to p_normal() or p_t(). The default is p_normal(0,2.5).

nsim — Number of simulations to draw

Value

prior_sample_hazard returns a data frame of the samples, and plot_prior_hazard generates a plot. No customisation options are provided for the plot function, which is just intended as a quick check.

A list with components:

alpha: Baseline log hazard scale parameter (log(eta) in the notation of the manual). For models with covariates, this is at the covariate values supplied in X, or at zero if X is not supplied.

hscale: Baseline hazard scale parameter (eta).

coefs: Spline coefficients. For non-proportional hazards model with covariates, these are returned at the suppled value of X, or at values of zero if X is not supplied.

beta: Multinomial logit-transformed spline coefficients.

hsd: Smoothing standard deviation for spline coefficients.

If X0 is supplied, then alpha0, hscale0, beta0, coefs0 are also returned, representing reference covariate values.

pcure is returned in cure models (the cure probability).

prior_sample_hazard *Generate and/or plot a sample from the prior distribution of M-spline hazard curves*

Description

Generates and/or plots the hazard curves (as functions of time) implied by a prior mean for the spline coefficients (a constant hazard by default) and particular priors for the baseline log hazard and smoothness standard deviation.

Usage

```
prior_sample_hazard(
  knots = NULL,
  df = 10,
  degree = 3,
  bsmooth = TRUE,
  coefs_mean = NULL,
  prior_hsd = p_gamma(2, 1),
  prior_hscale = NULL,
  smooth_model = "exchangeable",
  prior_loghr = NULL,
  formula = NULL,
  cure = NULL,
  nonprop = NULL,
  newdata = NULL,
  newdata0 = NULL,
  prior_hrsd = NULL,
  tmin = 0,
  tmax = 10,
  nsim = 10
)

plot_prior_hazard(
  knots = NULL,
  df = 10,
  degree = 3,
  bsmooth = TRUE,
  coefs_mean = NULL,
  prior_hsd = p_gamma(2, 1),
  prior_hscale = p_normal(0, 20),
  smooth_model = "random_walk",
  prior_loghr = NULL,
  formula = NULL,
  cure = NULL,
  nonprop = NULL,
  newdata = NULL,
  prior_hrsd = p_gamma(2, 1),
  tmin = 0,
  tmax = NULL,
  nsim = 10
)
```

Arguments

`knots` Vector of knot locations. If not supplied, `df` has to be specified. One of two rules is then used to choose the knot locations. If `bknot` is specified, a set of equally spaced knots between zero and `bknot` is used. Otherwise if `obstimes` is supplied, the knots are chosen as equally spaced quantiles of `obstimes`.

The number of knots (excluding zero) is `df - degree + 1` if `bsmooth` is `TRUE`, or `df - degree - 1` otherwise.

`df` Desired number of basis terms, or "degrees of freedom" in the spline. If `knots` is not supplied, the number of knots is then chosen to satisfy this.

`degree` Spline polynomial degree. Can only be changed from the default of 3 if `bsmooth` is `FALSE`.

`bsmooth` If `TRUE` then the function is constrained to also have zero derivative and second derivative at the boundary.

`coefs_mean` Spline basis coefficients that define the prior mean for the hazard function. By default, these are set to values that define a constant hazard function (see `mspline_constant_coefs`). They are normalised to sum to 1 internally (if they do not already).

`prior_hsd` Gamma prior for the standard deviation that controls the variability over time (or smoothness) of the hazard function. This should be a call to `p_gamma()`. The default is `p_gamma(2,1)`. See `prior_haz_sd` for a way to calibrate this to represent a meaningful belief.

`prior_hscale` Prior for the baseline log hazard scale parameter (`alpha` or `log(eta)`). This should be a call to a prior constructor function, such as `p_normal(0,1)` or `p_t(0,2,2)`. Supported prior distribution families are normal (parameters mean and SD) and t distributions (parameters location, scale and degrees of freedom). The default is a normal distribution with mean 0 and standard deviation 20.

Note that `eta` is not in itself a hazard, but it is proportional to the hazard (see the vignette for the full model specification).

"Baseline" is defined by the continuous covariates taking a value of zero and factor covariates taking their reference level. To use a different baseline, the data should be transformed appropriately beforehand, so that a value of zero has a different meaning. For continuous covariates, it helps for both computation and interpretation to define the value of zero to denote a typical value in the data, e.g. the mean.

`smooth_model` The default `"random_walk"`, specifies a random walk prior for the multinomial-logit spline coefficients, based on logistic distributions. See the methods vignette for full details.

The alternative `"exchangeable"` uses independent logistic priors on the multinomial-logit spline coefficients, conditionally on a common smoothing variance parameter. Note this is the method explained in the original survextrap paper (Jackson, BMC Med Res 2023). The random walk model is shown to perform better in Timmins et al (2025).

In non-proportional hazards models, setting `smooth_model` also determines whether an exchangeable or random walk model is used for the non-proportionality parameters (δ).

`prior_loghr` Priors for log hazard ratios. This should be a call to `p_normal()` or `p_t()`. A list of calls can also be provided, to give different priors to different coefficients, where the name of each list component matches the name of the coefficient, e.g. `list("age45-59" = p_normal(0,1), "age60+" = p_t(0,2,3))`

The default is `p_normal(0,2.5)` for all coefficients.

`formula` A survival formula in standard R formula syntax, with a call to `Surv()` on the left hand side.

Covariates included on the right hand side of the formula with be modelled with proportional hazards, or if `nonprop` is `TRUE` then a non-proportional hazards is used.

If `data` is omitted, so that the model is being fitted to external aggregate data alone, without individual data, then the formula should not include a `Surv()` call. The left-hand side of the formula will then be empty, and the right hand side specifies the covariates as usual. For example, `formula = ~1` if there are no covariates.

`cure` If `TRUE`, a mixture cure model is used, where the "uncured" survival is defined by the M-spline model, and the cure probability is estimated.

`nonprop` Non-proportional hazards model specification. This is achieved by modelling the spline basis coefficients in terms of the covariates. See the methods vignette for more details.

If `TRUE`, then all covariates are modelled with non-proportional hazards, using the same model formula as `formula`.

If this is a formula, then this is assumed to define a model for the dependence of the basis coefficients on the covariates.

IF this is `NULL` or `FALSE` (the default) then any covariates are modelled with proportional hazards.

`newdata` A data frame with one row, containing variables in the model formulae. Samples will then be drawn, for any covariate-dependent parameters, with covariates set to the values given here.

`newdata0` A data frame with one row, containing "reference" values of variables in the model formulae. The hazard ratio between the hazards at `newdata` and `newdata0` will be returned.

`prior_hrsd` Prior for the standard deviation parameters that smooth the non-proportionality effects over time in non-proportional hazards models. This should be a call to `p_gamma()` or a list of calls to `p_gamma()` with one component per covariate, as in `prior_loghr`. See `prior_hr_sd` for a way to calibrate this to represent a meaningful belief.

`tmin` Minimum plotting time. Defaults to zero.

`tmax` Maximum plotting time. Defaults to the highest knot.

`nsim` Number of simulations to draw

Value

A tidy data frame containing values of the hazard at a sequence of times, for different draws from the prior distribution.

p_hr *Derive a normal prior for the log hazard ratio parameter based on a guess at the hazard ratio*

Description

Derive a normal prior for the log hazard ratio parameter based on a guess at the hazard ratio. This assumes that the log upper limit is 2 standard deviations away from the log median.

Usage

```
p_hr(median, upper)
```

Arguments

`median`	Best guess (prior median) for a typical hazard ratio
`upper`	Upper limit of 95% credible interval for hazard ratio

Value

A normal prior in the format returned by `p_normal`, which can be passed directly to the `prior_loghr` argument in `survextrap`.

p_meansurv *Derive a normal prior for the log hazard scale parameter based on a guess at survival times*

Description

Derive a normal prior for the log hazard scale parameter based on a guess at survival times. The scale parameter is hard to interpret, and depends on the spline knots. However for any scale parameter, we can determine the spline coefficients that give a constant hazard (`mspline_constant_coefs`). Therefore if we can guess a typical survival time, we can guess a typical hazard (as 1 divided by the survival time) and deduce the scale parameter. The prior is then constructed by assuming normality on the log scale, and assuming the log upper credible limit is two SDs away from the log median.

Usage

```
p_meansurv(median, upper, mspline = NULL)
```

Arguments

`median`	Best guess (prior median) for a typical survival time
`upper`	Upper limit of 95% credible interval for a survival time
`mspline`	A list with components `knots` (vector of knots), `degree` (polynomial degree) and `bsmooth` (logical for smoothness constraint at boundary), defining an M-spline configuration.

Value

A normal prior in the format returned by `p_normal`, which can be passed directly to the `prior_hscale` argument in `survextrap`.

See Also

`prior_haz_const`, `mspline_constant_coefs`

`rmst` *Restricted mean survival time*

Description

Compute the restricted mean survival time from a model fitted with `survextrap`. Defined as the integral of the fitted survival curve up to a specified time.

Usage

```
rmst(
  x,
  t,
  newdata = NULL,
  newdata0 = NULL,
  wane_period = NULL,
  wane_nt = 10,
  disc_rate = 0,
  method = "gl",
  gl_nodes = 100,
  niter = NULL,
  summ_fns = NULL,
  sample = FALSE
)
```

Arguments

`x` A fitted model object as returned by `survextrap`

`t` Vector of times. The restricted mean survival time up to each one of these times will be computed.

`newdata` Data frame of covariate values to compute the output for. If there are covariates in the model and this is not supplied, the following default is used:

(a) if the only covariate is one factor variable, then the output is computed for each level of this factor.

(b) if there are multiple covariates, or any numeric covariates, then the output is computed at the mean of each numeric covariate in the original data, and at the baseline level of each factor covariate.

Note: caution is required about how treatment groups (for example) are stored in your data. If these are coded as numeric (0/1), then if `newdata` is not specified, only one output will be shown. This relates to the average value of this numeric variable over the data, which doesn't correspond to either of the treatment groups. To avoid this, a treatment group should be stored as a factor.

`newdata0` Data frame of covariate values defining the "untreated" group for use in treatment waning models. See `Survmspline_wane`.

`wane_period` Vector of two numbers, defining the time period over which the hazard is interpolated between the hazard of the "treated" group (taken from `newdata`) and the hazard of the "untreated" group (taken from `newdata0`). Optional - if this is not supplied, then no waning is assumed.

`wane_nt` Number of intervals defining the piecewise constant approximation to the hazard during the waning period.

`disc_rate` Discounting rate used to calculate the discounted mean or restricted mean survival time, using an exponential discounting function.

`method` Method of numerical integration to obtain the restricted mean survival time from the survival function.

The default is `method="gl"`, a Gauss-Legendre method with 100 nodes between zero and the maximum of `t`.

`method="adaptive"` uses the base R `integrate` function, which is much slower, but potentially more robust for badly-behaved survival functions.

`gl_nodes` Number of nodes for the Gauss-Legendre method.

`niter` Number of MCMC iterations to use to compute credible intervals. Set to a low value to make this function quicker, at the cost of some approximation error (which may not be important for plotting or model development).

`summ_fns` A list of functions to use to summarise the posterior sample. This is passed to `posterior::summarise_draws`. By default this is `list(median=median, ~quantile(.x, probs=c(0.025, 0.975)))`. If the list is named, then the names will be used for the columns of the output.

`sample` If `TRUE` then an MCMC sample is returned from the posterior of the output, rather than summary statistics.

Value

A data frame (tibble) with each row containing posterior summary statistics for a particular time and covariate value.

Or if `sample=TRUE`, an array with dimensions `length(t)`, `niter`, `nrow(newdata)`, giving the RMST evaluated at different times, MCMC iterations and covariate values respectively.

Examples

```
mod <- survextrap(Surv(years, status) ~ rx, data=colons, fit_method="opt")
rmst(mod, t=3, niter=100)
rmst(mod, t=3, summ_fns=list(mean=mean), niter=100)
```

standardise_to — *Constructor for a standardising population used for survextrap outputs*

Description

Standardised outputs are outputs from models with covariates, that are defined by marginalising (averaging) over covariate values in a given population, rather than being conditional on a given covariate value.

Usage

```
standardise_to(newdata, nstd = 1, random = FALSE)

standardize_to(newdata, nstd = 1, random = FALSE)
```

Arguments

`newdata`	Data frame describing a population.
`nstd`	Number of draws from the population distribution used per MCMC sample from the parameters when `random=TRUE`. With the default of 1, the value of the covariate vector X is essentially treated as if it were an additional parameter in the Bayesian model, drawn by Monte Carlo independently of the remaining parameters.
`random`	By default this is `FALSE`, indicating that standardised samples should be obtained by concatenating the posterior samples for each covariate value in the standard population. The sample from the standardised posterior of parameters then has size `niter` times the number of rows in `newdata`, where `niter` is the number of MCMC iterations used in the original `survextrap` fit. Computing the resulting output function (e.g. RMST which uses numerical integration) can then be computationally intensive if this sample size is large. A quicker alternative is to sample a random row of the standard population for each MCMC iteration. The standardised sample from the posterior then has size `niter`. This is specified by using `random=TRUE`. If this is used, then the result depends on the random number seed, and it should be checked that the results are stable to within the required number of significant figures. If not, run `survextrap` with more MCMC iterations or increase `nstd` here.

Details

These are produced by generating a Monte Carlo sample from the joint distribution of parameters θ and covariate values X, $p(X,\theta) = p(\theta|X)p(X)$, where $p(X)$ is defined by the empirical distribution of covariates in the standard population.

Hence applying a vectorised output function $g()$ (such as the RMST or survival probability) to this sample produces a sample from the posterior of $\int g(\theta|X)dX$: the average RMST (say) for a heterogeneous population.

See the Examples vignette for some examples and notes on computation.

Value

A copy of newdata, but with attributes added to indicate that this should be used as a standard population. When this newdata is passed to survextrap's output functions, the outputs will then be presented as an average over the empirical distribution of covariate values described by newdata, rather than as one output per row of newdata (distinct covariate values).

Examples

```
rxph_mod <- survextrap(Surv(years, status) ~ rx, data=colons, fit_method="opt")
ref_pop <- data.frame(rx = c("Obs","Lev+5FU"))

# covariate-specific outputs
survival(rxph_mod, t = c(5,10), newdata = ref_pop)

# standardised outputs
survival(rxph_mod, t = c(5,10), newdata = standardise_to(ref_pop))
```

---

summary.survextrap *Posterior summary statistics for parameters of survextrap models*

Description

Posterior summary statistics for parameters of survextrap models. The summary statistics presented by default include the posterior median and 95% credible intervals, alongside the Rhat convergence diagnostic and the bulk effective sample size (as computed by the posterior package). For models fitted by optimisation rather than MCMC, the posterior mode is always returned.

Any other posterior summary can be computed if the appropriate function to compute it is supplied in summ_fns.

Usage

```
## S3 method for class 'survextrap'
summary(object, summ_fns = NULL, ...)
```

Arguments

object	A fitted model object as returned by survextrap
summ_fns	A list of functions to calculate different posterior summaries from the MCMC sample. This is passed to posterior::summarise_draws. If the list is named, then the names will be used for the columns of the output. See the examples below for different ways this can be used. Defaults to list(median = median, ~quantile(.x, probs=c(0.025, 0.975)), sd = sd, rhat = posterior::rhat, ess_bulk = posterior::ess_bulk) Many useful such functions are provided with the posterior package.
...	Summary functions can also be supplied in separate arguments here. They will then be added to those supplied in summ_fns.

Value

A data frame (actually a `tibble`) of summary statistics for the model parameters.

The parameters, as indicated in the `variable` column, are:

`alpha`: Baseline log hazard scale. If there are covariates, this describes the log hazard scale with continuous covariates set to zero, and factor covariates set to their baseline levels. Note that this is not the log hazard, which also depends on the spline coefficients and basis. See `hazard` to extract the actual hazard.

`coefs`: Coefficients of the M-spline basis terms. If a non-proportional hazards model was fitted, these are with covariates set to zero or baseline levels.

`loghr`: Log hazard ratios for each covariate in the model. For cure models, this refers to covariates on survival for uncured people. For non-proportional hazards models, these are the multiplicative effects of covariates on the hazard scale parameter. See the methods vignette for a full description of this model.

`hr`: Hazard ratios (the exponentials of `loghr`).

`pcure`: Probability of cure (for cure models only). If there are covariates on cure, this parameter describes the probability of cure with continuous covariates set to zero, and factor covariates set to their baseline levels.

`logor_cure`: Log odds ratio of cure for each covariate on cure.

`or_cure`: Odds ratios of cure (the exponentials of `logor_cure`).

`nperr`: Standardised departures from proportional hazards in the non-proportional hazards model, defined as $b_{ks}^{(np)}/\sigma_s^{(np)}$ (see the methods vignette for definitions of these).

`hrsd`: Smoothness standard deviations τ_s for the non-proportionality effects.

Examples

```
mod <- survextrap(Surv(years, status) ~ rx, data=colons, fit_method="opt")
summary(mod)
summary(mod, mean=mean)
summary(mod, list(mean=mean))
summary(mod, list(mean=mean, ess_tail=posterior::ess_tail))
summary(mod, mean=mean, ess_tail=posterior::ess_tail)
```

---

survextrap *Flexible Bayesian parametric survival models*

Description

Flexible Bayesian parametric survival models. Individual data are represented using M-splines and a proportional hazards or flexible non-proportional hazards model. External aggregate data can be included, for example, to enable extrapolation outside the individual data. A fixed background hazard can also be included in an additive hazards (relative survival) model. Mixture cure versions of these models can also be used.

Usage

```
survextrap(
  formula,
  data = NULL,
  external = NULL,
  cure = FALSE,
  nonprop = FALSE,
  prior_hscale = p_normal(0, 20),
  prior_loghr = NULL,
  prior_hsd = p_gamma(2, 1),
  prior_cure = p_beta(1, 1),
  prior_logor_cure = NULL,
  prior_hrsd = p_gamma(2, 1),
  backhaz = NULL,
  backhaz_strata = NULL,
  mspline = NULL,
  add_knots = NULL,
  smooth_model = "random_walk",
  hsd = "bayes",
  coefs_mean = NULL,
  fit_method = "mcmc",
  loo = (fit_method == "mcmc"),
  ...
)
```

Arguments

`formula` A survival formula in standard R formula syntax, with a call to `Surv()` on the left hand side.

Covariates included on the right hand side of the formula with be modelled with proportional hazards, or if `nonprop` is `TRUE` then a non-proportional hazards is used.

If `data` is omitted, so that the model is being fitted to external aggregate data alone, without individual data, then the formula should not include a `Surv()` call. The left-hand side of the formula will then be empty, and the right hand side specifies the covariates as usual. For example, `formula = ~1` if there are no covariates.

`data` Data frame containing variables in `formula`. Variables should be in a data frame, and not in the working environment.

This may be omitted, in which case `external` must be supplied. This allows a model to be fitted to external aggregate data alone, without any individual-level data.

`external` External data as a data frame of aggregate survival counts with columns named:

`start`: Start time

`stop`: Follow-up time

`n`: Number of people alive at `start`

`r`: Number of those people who are still alive at `stop`

If there are covariates in formula, then the values they take in the external data must be supplied as additional columns in external. Therefore if there are external data, the covariates in formula and data should not be named start,stop,n or r.

cure
If TRUE, a mixture cure model is used, where the "uncured" survival is defined by the M-spline model, and the cure probability is estimated.

nonprop
Non-proportional hazards model specification. This is achieved by modelling the spline basis coefficients in terms of the covariates. See the methods vignette for more details.

If TRUE, then all covariates are modelled with non-proportional hazards, using the same model formula as formula.

If this is a formula, then this is assumed to define a model for the dependence of the basis coefficients on the covariates.

IF this is NULL or FALSE (the default) then any covariates are modelled with proportional hazards.

prior_hscale
Prior for the baseline log hazard scale parameter (alpha or log(eta)). This should be a call to a prior constructor function, such as p_normal(0,1) or p_t(0,2,2). Supported prior distribution families are normal (parameters mean and SD) and t distributions (parameters location, scale and degrees of freedom). The default is a normal distribution with mean 0 and standard deviation 20.

Note that eta is not in itself a hazard, but it is proportional to the hazard (see the vignette for the full model specification).

"Baseline" is defined by the continuous covariates taking a value of zero and factor covariates taking their reference level. To use a different baseline, the data should be transformed appropriately beforehand, so that a value of zero has a different meaning. For continuous covariates, it helps for both computation and interpretation to define the value of zero to denote a typical value in the data, e.g. the mean.

prior_loghr
Priors for log hazard ratios. This should be a call to p_normal() or p_t(). A list of calls can also be provided, to give different priors to different coefficients, where the name of each list component matches the name of the coefficient, e.g. list("age45-59" = p_normal(0,1), "age60+" = p_t(0,2,3))

The default is p_normal(0,2.5) for all coefficients.

prior_hsd
Gamma prior for the standard deviation that controls the variability over time (or smoothness) of the hazard function. This should be a call to p_gamma(). The default is p_gamma(2,1). See prior_haz_sd for a way to calibrate this to represent a meaningful belief.

prior_cure
Prior for the baseline cure probability. This should be a call to p_beta(). The default is a uniform prior, p_beta(1,1). Baseline is defined by the mean of continuous covariates and the reference level of factor covariates.

prior_logor_cure
Priors for log odds ratios on cure probabilities. This should be a call to p_normal() or p_t(). The default is p_normal(0,2.5).

prior_hrsd
Prior for the standard deviation parameters that smooth the non-proportionality effects over time in non-proportional hazards models. This should be a call to

p_gamma() or a list of calls to p_gamma() with one component per covariate, as in prior_loghr. See prior_hr_sd for a way to calibrate this to represent a meaningful belief.

backhaz

Background hazard, that is, for causes of death other than the cause of interest. This defines a "relative survival" or "additive hazards" model. The overall hazard that describes the all-cause survival data (given in the data and/or external argument) is then modelled as the sum of a cause-specific hazard and a background hazard.

The background hazard is assumed to be known, and the cause-specific hazard is modelled with the flexible parametric model.

The background hazard can be supplied in two forms. The meaning of predictions from the model depends on which of these is used.

(a) A data frame with columns "hazard" and "time", specifying the background hazard at all times as a piecewise-constant (step) function. Each row gives the background hazard between the specified time and the next time. The first element of "time" should be 0, and the final row specifies the hazard at all times greater than the last element of "time". Predictions from the model fitted by survextrap will then include this background hazard, because it is known at all times.

(b) The (quoted) name of a variable in the data giving the background hazard. For censored cases, the exact value does not matter. The predictions from survextrap will then describe the excess hazard or survival on top of this background. The overall hazard cannot be predicted in general, because the background hazard is only specified over a limited range of time.

If there is external data, and backhaz is supplied in form (b), then the user should also supply the background survival at the start and stop points in columns of the external data named "backsurv_start" and "backsurv_stop". That is, the probability of survival up to each of these times for someone alive at time 0. This should describe the same reference population as backhaz, though the package does not check for consistency between these.

If there are stratifying variables specified in backhaz_strata, then there should be multiple rows giving the background hazard value for each time period and stratifying variable.

If backhaz is NULL (the default) then no background hazard component is included in the model.

backhaz_strata

A character vector of names of variables that appear in backhaz that indicate strata, e.g. backhaz_strata = c("agegroup","sex"). This allows different background hazard values to be used for different subgroups. These variables must also appear in the datasets being modelled, that is, in data, external or both. Each row of those datasets should then have a corresponding row in backhaz which has the same values of the stratifying variables.

This is NULL by default, indicating no stratification of the background hazard.

If stratification is done, then backhaz must be supplied in form (a), as a data frame rather than a variable in the data.

mspline

A list of control parameters defining the spline model.

`knots`: Spline knots. If this is not supplied, then the number of knots is taken from `df`, and their location is taken from equally-spaced quantiles of the observed event times in the individual-level data.

`add_knots`: This is intended to be used when there are `external` data included in the model. External data are typically outside the time period covered by the individual data. `add_knots` would then be chosen to span the time period covered by the external data, so that the hazard trajectory can vary over that time.

If there are external data, and both `knots` and `add_knots` are omitted, then a default set of knots is chosen to span both the individual and external data, by taking the quantiles of a vector defined by concatenating the individual-level event times with the `start` and `stop` times in the external data.

`df`: Degrees of freedom, i.e. the number of parameters (or basis terms) intended to result from choosing knots based on quantiles of the data. The total number of parameters will then be `df` plus the number of additional knots specified in `add_knots`. `df` defaults to 10. This does not necessarily overfit, because the function is smoothed through the prior.

`degree`: Polynomial degree used for the basis function. The default is 3, giving a cubic. This can only be changed from 3 if `bsmooth` is `FALSE`.

`bsmooth`: If `TRUE` (on by default) the spline is smoother at the highest knot, by defining the derivative and second derivative at this point to be zero.

`add_knots` Any extra knots beyond those chosen from the individual-level data (or supplied in `knots`). All other spline specifications are set to their defaults, as described in `mspline`. For example, `add_knots = 10` is a shorthand for `mspline = list(add_knots = 10)`.

`smooth_model` The default `"random_walk"`, specifies a random walk prior for the multinomial-logit spline coefficients, based on logistic distributions. See the methods vignette for full details.

The alternative `"exchangeable"` uses independent logistic priors on the multinomial-logit spline coefficients, conditionally on a common smoothing variance parameter. Note this is the method explained in the original survextrap paper (Jackson, BMC Med Res 2023). The random walk model is shown to perform better in Timmins et al (2025).

In non-proportional hazards models, setting `smooth_model` also determines whether an exchangeable or random walk model is used for the non-proportionality parameters (δ).

`hsd` Smoothing variance parameter estimation.

`"bayes"`: the smoothing parameter is estimated by full Bayes (the default).

`"eb"`: empirical Bayes is used.

Alternatively, if a number is supplied here, then the smoothing parameter is fixed to this number.

`coefs_mean` Spline basis coefficients that define the prior mean for the hazard function. By default, these are set to values that define a constant hazard function (see `mspline_constant_coefs`). They are normalised to sum to 1 internally (if they do not already).

`fit_method` Method from **rstan** used to fit the model.

If `fit_method="mcmc"` then a sample from the posterior is drawn using Markov Chain Monte Carlo sampling, via **rstan**'s `rstan::sampling()` function. This is the default. It is the most accurate but the slowest method.

If `fit_method="opt"`, then instead of an MCMC sample from the posterior, `survextrap` returns the posterior mode calculated using optimisation, via **rstan**'s `rstan::optimizing()` function. A sample from a normal approximation to the (real-line-transformed) posterior distribution is drawn in order to obtain credible intervals. This is useful for model development, while using MCMC for the "final answer".

If `fit_method="vb"`, then variational Bayes methods are used, via **rstan**'s `rstan::vb()` function. This is labelled as "experimental" by **rstan**. It might give a better approximation to the posterior than `fit_method="opt"`, and is faster than MCMC, in particular for large datasets, but has not been investigated in depth for these models.

`loo` Compute leave-one-out cross-validation statistics. This is done by default. Set to `FALSE` to not compute them. If these statistics are computed, then they are returned in the `loo` and `loo_external` components of the object returned by `survextrap`. `loo` describes the fit of the model to the individual-level data, and `loo_external` describes fit to the external data.

See the `"examples"` vignette for more explanation of these.

`...` Additional arguments to supply to control the Stan fit, passed to the appropriate **rstan** function, depending on which is chosen through the `fit_method` argument.

Value

A list of objects defining the fitted model. These are not intended to be extracted directly by users. Instead see `summary.survextrap` for summarising the parameter estimates, and the functions `hazard`, `survival`, `rmst` and others for computing interesting summaries of the fitted survival distribution.

The object returned by `rstan` containing samples from the fitted model is returned in the `stanfit` component. See the rstan documentation. The function `get_draws` is provided to convert this to a simple matrix of posterior samples with all chains collapsed.

References

Jackson, C. (2023) `survextrap`: a package for flexible and transparent survival extrapolation. BMC Medical Research Methodology 23:282. doi:10.1186/s12874023020941

Timmins I, Torabi F, Jackson C, Lambert P, Sweeting M J. (2025) Simulation-based assessment of a Bayesian survival model with flexible baseline hazard and time-dependent effects. doi:10.48550/arXiv.2503.21388.

`survival` *Estimates of survival from a* `survextrap` *model*

Description

Estimates of survival probabilities at particular times, from a `survextrap` model

Usage

```
survival(
  x,
  newdata = NULL,
  t = NULL,
  tmax = NULL,
  niter = NULL,
  summ_fns = NULL,
  sample = FALSE,
  newdata0 = NULL,
  wane_period = NULL,
  wane_nt = 10
)
```

Arguments

`x` A fitted model object as returned by `survextrap`

`newdata` Data frame of covariate values to compute the output for. If there are covariates in the model and this is not supplied, the following default is used:

(a) if the only covariate is one factor variable, then the output is computed for each level of this factor.

(b) if there are multiple covariates, or any numeric covariates, then the output is computed at the mean of each numeric covariate in the original data, and at the baseline level of each factor covariate.

Note: caution is required about how treatment groups (for example) are stored in your data. If these are coded as numeric (0/1), then if `newdata` is not specified, only one output will be shown. This relates to the average value of this numeric variable over the data, which doesn't correspond to either of the treatment groups. To avoid this, a treatment group should be stored as a factor.

`t` Vector of times at which to compute the estimates.

`tmax` Maximum time at which to compute the estimates. If `t` is supplied, then this is ignored. If `t` is not supplied, then `t` is set to a set of 100 equally spaced time points from 0 to `tmax`. If both `tmax` and `t` are not supplied, then `tmax` is set to the maximum follow up time in the data.

`niter` Number of MCMC iterations to use to compute credible intervals. Set to a low value to make this function quicker, at the cost of some approximation error (which may not be important for plotting or model development).

summ_fns
: A list of functions to use to summarise the posterior sample. This is passed to `posterior::summarise_draws`. By default this is `list(median=median, ~quantile(.x, probs=c(0.025, 0.975)))`. If the list is named, then the names will be used for the columns of the output.

sample
: If `TRUE` then the MCMC samples are returned instead of being summarised as a median and 95% credible intervals.

newdata0
: Data frame of covariate values defining the "untreated" group for use in treatment waning models. See `Survmspline_wane`.

wane_period
: Vector of two numbers, defining the time period over which the hazard is interpolated between the hazard of the "treated" group (taken from `newdata`) and the hazard of the "untreated" group (taken from `newdata0`). Optional - if this is not supplied, then no waning is assumed.

wane_nt
: Number of intervals defining the piecewise constant approximation to the hazard during the waning period.

Value

A data frame (tibble) giving posterior summary statistics, or (if `sample=TRUE`) an array giving samples from the posterior.

Survmspline *M-spline survival distribution*

Description

Probability density, distribution, quantile, random generation, hazard, cumulative hazard, mean and restricted mean functions for the M-spline time-to-event model. This can optionally be combined with a cure probability, and / or with a known background hazard trajectory that is a piecewise-constant function of time.

Usage

```
psurvmspline(
  q,
  alpha,
  coefs,
  knots,
  degree = 3,
  lower.tail = TRUE,
  log.p = FALSE,
  pcure = 0,
  offsetH = 0,
  backhaz = NULL,
  bsmooth = TRUE
)
```

```
Hsurvmspline(
  x,
  alpha,
  coefs,
  knots,
  degree = 3,
  log = FALSE,
  pcure = 0,
  offsetH = 0,
  backhaz = NULL,
  bsmooth = TRUE
)

hsurvmspline(
  x,
  alpha,
  coefs,
  knots,
  degree = 3,
  log = FALSE,
  pcure = 0,
  offseth = 0,
  backhaz = NULL,
  bsmooth = TRUE
)

dsurvmspline(
  x,
  alpha,
  coefs,
  knots,
  degree = 3,
  log = FALSE,
  pcure = 0,
  offseth = 0,
  offsetH = 0,
  backhaz = NULL,
  bsmooth = TRUE
)

qsurvmspline(
  p,
  alpha,
  coefs,
  knots,
  degree = 3,
  lower.tail = TRUE,
  log.p = FALSE,
  pcure = 0,
  offsetH = 0,
  backhaz = NULL,
  bsmooth = TRUE
)

rsurvmspline(
  n,
  alpha,
  coefs,
  knots,
  degree = 3,
  pcure = 0,
  offsetH = 0,
  backhaz = NULL,
  bsmooth = TRUE
)

rmst_survmspline(
  t,
  alpha,
  coefs,
  knots,
  degree = 3,
  pcure = 0,
  offsetH = 0,
  backhaz = NULL,
  bsmooth = TRUE,
  disc_rate = 0,
  method = "gl",
  gl_nodes = 100
)

mean_survmspline(
  alpha,
  coefs,
  knots,
  degree = 3,
  pcure = 0,
  offsetH = 0,
  backhaz = NULL,
  bsmooth = TRUE,
  disc_rate = 0
)
```

Arguments

`alpha`	Log hazard scale parameter.

`coefs` Spline basis coefficients. These should sum to 1, otherwise they are normalised internally to sum to 1. Supplied either as a vector with one element per basis term, or a matrix with one column per basis term, and rows for alternative values of the coefficients (in vectorised usage of this function). If an array is supplied, it is collapsed into a matrix with number of columns equal to the final dimension of the array.

`knots` Locations of knots on the axis of time, supplied in increasing order. These include the two boundary knots.

In vectorised usage of these functions, the knots and degree must be the same for all alternative times and parameter values.

`degree` Spline polynomial degree. Can only be changed from the default of 3 if `bsmooth` is `FALSE`.

`lower.tail` logical; if `TRUE` (default), probabilities are $P(X \le x)$, otherwise, $P(X > x)$.

`pcure` Probability of "cure", which defaults to zero. If this is non-zero, this defines a "mixture cure" version of the M-spline model.

`offsetH` Constant to be added to the cumulative hazard.

`backhaz` A data frame that defines the background hazard as a piecewise-constant function of time. This should have two columns, named `"time"` and `"hazard"`. Each row gives the "background" hazard between the specified time and the next time. The first element of `"time"` should be 0, and the final row specifies the hazard at all times greater than the last element of `"time"`.

`bsmooth` If `TRUE` then the function is constrained to also have zero derivative and second derivative at the boundary.

`x, q, t` Vector of times.

`log, log.p` Return log density or probability.

`offseth` Constant to be added to the hazard, e.g. representing a "background" risk of death from causes other than the cause of interest.

`p` Vector of probabilities.

`n` Number of random numbers to simulate.

`disc_rate` Discounting rate used to calculate the discounted mean or restricted mean survival time, using an exponential discounting function.

`method` Method of numerical integration to obtain the restricted mean survival time from the survival function.

The default is `method="gl"`, a Gauss-Legendre method with 100 nodes between zero and the maximum of `t`.

`method="adaptive"` uses the base R `integrate` function, which is much slower, but potentially more robust for badly-behaved survival functions.

`gl_nodes` Number of nodes for the Gauss-Legendre method.

Details

These are the same as the M-splines used to model survival data in `rstanarm`, except that an additional assumption is made that the hazard is constant beyond the boundary knots at its value at the boundary. This gives a continuous but non-smooth function.

The "cure" model can be interpreted in two different ways. These result in identical probability distribution functions for the event time, hence they are indistinguishable from data:

(a) a model where everyone follows the same hazard trajectory that is decreasing through time, with a higher rate of decrease for higher `pcure`.

(b) a model where a person either has a constant zero hazard at all times, or a hazard that follows a parametric model (M-spline in this case). The probability that a person has a zero hazard is `pcure`. This is the "mixture model" interpretation.

In the "background hazard" model, the overall hazard is defined as a sum of the background hazard and a cause-specific hazard. The cause-specific hazard is modelled with the M-spline model, and the background hazard is assumed to be a known piecewise-constant function defined by `backhaz`.

If both `backhaz` and `pcure` are supplied, then the cure probablity applies only to the cause-specific hazard. That is, the cause-specific hazard decreases to zero, and the overall hazard converges towards the background hazard, as time increases.

Value

`dsurvmspline` gives the density, `psurvmspline` gives the distribution function, `hsurvmspline` gives the hazard and `Hsurvmspline` gives the cumulative hazard.

`qsurvmspline` gives the quantile function, which is computed by numerical inversion.

`rsurvmspline` generates random survival times by using `qsurvmspline` on a sample of uniform random numbers.

Author(s)

Christopher Jackson chris.jackson@mrc-bsu.cam.ac.uk

References

Ramsay, J. O. (1988). Monotone regression splines in action. Statistical Science, 3(4): 425-441.

Brilleman, S. L., Elci, E. M., Novik, J. B., & Wolfe, R. (2020). Bayesian survival analysis using the rstanarm R package. arXiv preprint arXiv:2002.09633.

Wang, W., Yan, J. (2021). Shape-restricted regression splines with R package splines2. Journal of Data Science, *19*(3), 498-517.

`Survmspline_wane` *M-spline survival distribution under treatment effect waning*

Description

This defines the CDF, cumulative hazard and hazard of a survival distribution defined by combining the hazards of two different groups (e.g. "treated" and "untreated") each defined by a standard M-spline model. The log hazards of one group and the other are interpolated over a defined period of time. This may be used for models where the treatment effect wanes, over a period of time, between an estimated hazard ratio and zero.

Usage

```
Hsurvmspline_wane(
  x,
  alpha1,
  alpha0,
  coefs1,
  coefs0,
  knots,
  degree = 3,
  wane_period,
  wane_nt = 10,
  pcure1 = 0,
  pcure0 = 0,
  offsetH = 0,
  backhaz = NULL,
  bsmooth = TRUE,
  log = FALSE
)

psurvmspline_wane(
  q,
  alpha1,
  alpha0,
  coefs1,
  coefs0,
  knots,
  degree = 3,
  bsmooth = TRUE,
  wane_period,
  wane_nt = 10,
  lower.tail = TRUE,
  pcure1 = 0,
  pcure0 = 0,
  offsetH = 0,
  backhaz = NULL,
  log.p = FALSE
)

hsurvmspline_wane(
  x,
  alpha1,
  alpha0,
  coefs1,
  coefs0,
  knots,
  degree = 3,
  bsmooth = TRUE,
  wane_period,
  wane_nt = 10,
  pcure1 = 0,
  pcure0 = 0,
  offseth = 0,
  backhaz = NULL,
  log = FALSE
)

dsurvmspline_wane(
  x,
  alpha1,
  alpha0,
  coefs1,
  coefs0,
  knots,
  degree = 3,
  bsmooth = TRUE,
  wane_period,
  wane_nt = 10,
  pcure1 = 0,
  pcure0 = 0,
  offseth = 0,
  offsetH = 0,
  backhaz = NULL,
  log = FALSE
)

qsurvmspline_wane(
  p,
  alpha1,
  alpha0,
  coefs1,
  coefs0,
  knots,
  degree = 3,
  bsmooth = TRUE,
  lower.tail = TRUE,
  log.p = FALSE,
  pcure1 = 0,
  pcure0 = 0,
  offsetH = 0,
  backhaz = NULL,
  wane_period,
  wane_nt = 10
)

rsurvmspline_wane(
  n,
  alpha1,
  alpha0,
  coefs1,
  coefs0,
  knots,
  degree = 3,
  bsmooth = TRUE,
  pcure1 = 0,
  pcure0 = 0,
  offsetH = 0,
  backhaz = NULL,
  wane_period,
  wane_nt = 10
)

rmst_survmspline_wane(
  t,
  alpha1,
  alpha0,
  coefs1,
  coefs0,
  knots,
  degree = 3,
  pcure1 = 0,
  pcure0 = 0,
  offsetH = 0,
  backhaz = NULL,
  bsmooth = TRUE,
  wane_period,
  wane_nt = 10,
  disc_rate = 0,
  method = "gl",
  gl_nodes = 100
)

mean_survmspline_wane(
  alpha1,
  alpha0,
  coefs1,
  coefs0,
  knots,
  degree = 3,
  pcure1 = 0,
  pcure0 = 0,
  backhaz = NULL,
  bsmooth = TRUE,
  wane_period,
  wane_nt = 10,
  disc_rate = 0
)
```

Arguments

x, q, t Vector of times.

alpha1, coefs1, pcure1 log hazard intercept, spline coefficients and cure probability before the start of the waning period ("treated")

alpha0, coefs0, pcure0 log hazard intercept, spline coefficients and cure probability after the end of the waning period ("untreated")

knots Locations of knots on the axis of time, supplied in increasing order. These include the two boundary knots.

In vectorised usage of these functions, the knots and degree must be the same for all alternative times and parameter values.

degree Spline polynomial degree. Can only be changed from the default of 3 if bsmooth is FALSE.

wane_period vector of two components giving start and stop of waning period

wane_nt time resolution for piecewise constant hazard approximation in the waning period. If this is not specified, defaults to dividing the waning period into 10 pieces.

offsetH Constant to be added to the cumulative hazard.

backhaz A data frame that defines the background hazard as a piecewise-constant function of time. This should have two columns, named "time" and "hazard". Each row gives the "background" hazard between the specified time and the next time. The first element of "time" should be 0, and the final row specifies the hazard at all times greater than the last element of "time".

bsmooth If TRUE then the function is constrained to also have zero derivative and second derivative at the boundary.

log, log.p Return log density or probability.

lower.tail logical; if TRUE (default), probabilities are $P(X \leq x)$, otherwise, $P(X > x)$.

offseth Constant to be added to the hazard, e.g. representing a "background" risk of death from causes other than the cause of interest.

p Vector of probabilities.

n Number of random numbers to simulate.

disc_rate Discounting rate used to calculate the discounted mean or restricted mean survival time, using an exponential discounting function.

method Method of numerical integration to obtain the restricted mean survival time from the survival function.

The default is method="gl", a Gauss-Legendre method with 100 nodes between zero and the maximum of t.

method="adaptive" uses the base R integrate function, which is much slower, but potentially more robust for badly-behaved survival functions.

gl_nodes Number of nodes for the Gauss-Legendre method.

Details

This distribution is defined as follows.

- Between time 0 and `wane_period[1]`, the "treated" hazard is used, as defined by an M-spline with intercept `alpha1`.
- Between `wane_period[1]` and `wane_period[2]`, the log hazard is defined by linear interpolation. The waning period is divided into a number of discrete pieces in which the hazard is assumed to be constant, defined by the hazard at the start of the piece. These hazard values are obtained from the spline model, using an intercept parameter `alpha` (log scale parameter) defined by linearly interpolating between `alpha1` and `alpha0` over the waning period. The cumulative hazard at any time can then be deduced by adding up contributions on each piece.
- After `wane_period[2]`, the "untreated" hazard is used, as defined by an M-spline with intercept `alpha0`.

See the methods vignette for more details and examples.

This can be used to predict the hazard of a person treated with a treatment whose short-term effect is estimated from shorter-term data, but we wish to extrapolate this model over a longer period where the effect is assumed to diminish.

This may not work if the hazard is zero or infinite at any point in the waning period (thus the log hazard is indeterminate). This might typically happen at time 0.

Value

`psurvmspline_wane` gives the CDF, `Hsurvmspline_wane` gives the cumulative hazard, `hsurvmspline_wane` gives the hazard, `dsurvmspline_wane` gives the PDF, `qsurvmspline_wane` gives the quantiles, and `rsurvmspline_wane` generates random numbers from the distribution.

Index