

Package: exnexSurv (via r-universe)

October 13, 2026

Type Package

Title Bayesian EXNEX Models for Survival Analysis in Basket Trials

Version 1.4.0

Description Implements the Bayesian Exchangeable Non-Exchangeable (EXNEX) framework for right-censored log-normal survival data in basket trials. Based on 'Rcpp' and 'RcppArmadillo', the package provides a fast Gibbs sampler supporting EXNEX, complete pooling, and no pooling models to facilitate methodological comparisons and simulation studies.

License MIT + file LICENSE

URL <https://github.com/victorney/exnexSurv>,
<https://victorney.github.io/exnexSurv/>

BugReports <https://github.com/victorney/exnexSurv/issues>

Imports Rcpp (>= 1.1.1), hardhat (>= 1.4.0), checkmate (>= 2.1.0),
progressr, future, future.apply

Suggests testthat (>= 3.3.0), survival (>= 3.8), knitr, rmarkdown,
ggplot2, bayesplot, viridisLite, posterior

VignetteBuilder knitr

LinkingTo Rcpp, RcppArmadillo

Encoding UTF-8

Language en-US

Config/roxygen2/version 8.1.0

Repository <https://victorney.r-universe.dev>

Date/Publication 2026-09-12 23:02:34 UTC

RemoteUrl <https://github.com/victorney/exnexsurv>

RemoteRef HEAD

RemoteSha 29d927f8c5625e802ba7854a30bb17f9739a9775

Contents

compare_waic	2
compute_waic	3
convergence_diagnostics	4
exnex_surv	5
median_survival	5
plot.pooling_surv	6
plot.survival_exnex	7
pooling_surv	7
posterior_predictive_check	11
print.pooling_surv	12
probability_superiority	12
rank_probabilities	13
rmst	14
shrinkage_plot	15
simulate_data	16
summary.pooling_surv	17
survival_curves	17
Index	19

compare_waic	<i>Compare multiple pooling_surv fits by WAIC</i>
--------------	---

Description

Computes [compute_waic()] for each supplied fit and reports them in a single ‘data.frame’, sorted by ascending WAIC.

Usage

```
compare_waic(..., digit = 2)
```

Arguments

- ... Two or more fitted ‘pooling_surv’ objects.
- digit Number of decimal places for the reported statistics.

Details

Fits are labelled with the variable names used in the call when supplied unnamed (e.g. ‘fit_exnex’, ‘fit_complete’), or by the names given to ‘...’; when even those are unavailable, the fitted ‘pooling’ variant is used, with a final ‘fit1’/‘fit2’ fallback for duplicate or anonymous labels.

Value

A ‘data.frame’ with one row per model and columns ‘model’, ‘waic’, ‘se_elpd_waic’, ‘lpd’, ‘p_waic’, ‘elpd_waic’.

compute_waic	<i>Pointwise log-likelihood and WAIC from an pooling_surv fit</i>
--------------	---

Description

Computes the Watanabe-Akaike Information Criterion (WAIC) for a fitted ‘pooling_surv’ model. WAIC is a fully Bayesian information criterion that uses the posterior draws of the log-likelihood and adds a penalty for effective number of parameters p_{waic} .

Usage

```
compute_waic(fit, ...)
```

Arguments

<code>fit</code>	A fitted ‘pooling_surv’ object.
<code>...</code>	Unused.

Details

For each observation i , with posterior draws indexed by $s = 1, \dots, S$, we compute the pointwise log-likelihood $\log p(y_i | \theta^{(s)})$, where the log-normal AFT model gives for an observed event ($\delta_i = 1$)

$$\log p(y_i | \theta^{(s)}) = \log \phi \left(\frac{\log t_i - \eta_i^{(s)}}{\sigma^{(s)}} \right) - \log(\sigma^{(s)} t_i),$$

and for a censored observation ($\delta_i = 0$)

$$\log p(y_i | \theta^{(s)}) = \log \left[1 - \Phi \left(\frac{\log t_i - \eta_i^{(s)}}{\sigma^{(s)}} \right) \right],$$

where $\eta_i^{(s)}$ is the linear predictor built from the draws.

WAIC is then

$$\begin{aligned} lpd &= \sum_i \log \left(\frac{1}{S} \sum_s \exp \log p(y_i | \theta^{(s)}) \right), \\ p_{waic} &= \sum_i \text{var}_s (\log p(y_i | \theta^{(s)})), \\ \text{WAIC} &= -2(lpd - p_{waic}). \end{aligned}$$

Rows with extreme leverage (‘p_waic’ above a large threshold) may flag issues; the function returns a ‘pointwise’ matrix so users can investigate.

Value

A named list with elements ‘waic’, ‘se_elpd_waic’, ‘lpd’, ‘p_waic’, ‘elpd_waic’, and ‘pointwise’ (an ‘n’ by 3 matrix with columns ‘lpd’, ‘p_waic’, ‘waic’).

`convergence_diagnostics`*Convergence diagnostics (R-hat and effective sample size)*

Description

Computes R-hat, bulk effective sample size (ESS), and tail ESS for every parameter of a fitted ‘pooling_surv’ object, using the ‘posterior’ package. All three follow the definitions in Vehtari, Gelman, Simpson, Carpenter & Buerkner (2021, "Rank-normalization, folding, and localization"): the R-hat is split and rank-normalized, bulk ESS is computed on rank-normalized draws, and tail ESS quantifies sampling adequacy in the 5 diagnostic is computed (so ‘chains = 1’ still gets a meaningful R-hat).

Usage

```
convergence_diagnostics(fit, ...)
```

Arguments

<code>fit</code>	A fitted ‘pooling_surv’ object (result of ‘pooling_surv()’).
<code>...</code>	Unused.

Details

Rules of thumb: R-hat close to 1.0 (say < 1.01) and ESS in the hundreds give reasonably stable summaries. Low ESS with a good R-hat means the chain is slow to move but honest; both R-hat above ~ 1.01 and very low ESS (e.g. below 100) are worth investigating before trusting the posterior.

Value

A data frame with one row per parameter and columns ‘parameter’, ‘rhat’, ‘ess_bulk’, and ‘ess_tail’. All values are ‘NA’ if the ‘posterior’ package is not installed.

Examples

```
set.seed(2719)
trial_data <- simulate_data(n = 90, K = 3, censoring_rate = 0.3)
fit <- pooling_surv(survival::Surv(time, event) ~ group,
  data = trial_data,
  iter = 300, warmup = 150, chains = 2)
convergence_diagnostics(fit)
```

exnex_surv

Deprecated alias for EXNEX fitting

Description

‘exnex_surv()’ is deprecated in favour of [pooling_surv()]. It emits a one-line deprecation message and forces ‘pooling = "exnex"’, so existing calls continue to fit the EXNEX hierarchy unchanged.

Usage

```
exnex_surv(x, ...)
```

Arguments

x	An object containing the predictors (subgroup assignment). Can be a data frame or a formula.
...	Additional arguments passed to methods.

Value

An object of class ‘pooling_surv’, identical to ‘pooling_surv(..., pooling = "exnex)”’.

Examples

```
## Not run:
fit <- exnex_surv(survival::Surv(time, event) ~ group, data = d)

## End(Not run)
```

median_survival

Posterior median survival time

Description

For the log-normal AFT model the median survival time for a linear predictor η is $t_{med} = \exp(\eta)$, since $S(t) = 0.5$ when $\log t = \eta$. Posterior draws of θ and β therefore induce a posterior distribution of t_{med} whose quantiles are reported.

Usage

```
median_survival(fit, newdata = NULL, level = 0.95, ...)
```

Arguments

fit	A fitted 'pooling_surv' object.
newdata	Optional data frame (one row gives one median).
level	Credible-interval level (default '0.95').
...	Unused.

Value

A 'data.frame' with columns 'group', 'median', 'lower', 'upper'.

plot.pooling_surv	<i>Plot parameter traces from an pooling_surv fit</i>
-------------------	---

Description

Generates one bayesplot traceplot per parameter.

Usage

```
## S3 method for class 'pooling_surv'
plot(x, parameters = NULL, ask = interactive(), ...)
```

Arguments

x	A fitted 'pooling_surv' object.
parameters	Optional character vector of parameter names to plot.
ask	Should R pause between plots? Defaults to 'interactive()'.
...	Unused.

Value

Invisibly returns 'x'.

plot.survival_exnex	<i>Plot survival curves from an pooling_surv fit</i>
---------------------	--

Description

Draws the posterior median and credible band of the survival function for each group. Requires ‘ggplot2’.

Usage

```
## S3 method for class 'survival_exnex'
plot(x, ...)
```

Arguments

x	An object of class ‘survival_exnex’ returned by [survival_curves()].
...	Unused.

Value

Invisibly returns ‘x’.

pooling_surv	<i>Fit Bayesian Pooling Survival Models</i>
--------------	---

Description

Fits Bayesian models for right-censored log-normal survival data in basket trials using a data-augmented Gibbs sampler. The ‘pooling’ argument selects the structural assumption on the basket effects:

Usage

```
pooling_surv(x, ...)

## Default S3 method:
pooling_surv(x, ...)

## S3 method for class 'formula'
pooling_surv(
  formula,
  data,
  pooling = "exnex",
  priors = list(),
  iter = 2000,
```

```

warmup = 1000,
thin = 1,
chains = 1,
parallel_chains = FALSE,
group_col = NULL,
seed = NULL,
verbose = TRUE,
...
)

## S3 method for class 'data.frame'
pooling_surv(
  x,
  y,
  pooling = "exnex",
  priors = list(),
  iter = 2000,
  warmup = 1000,
  thin = 1,
  chains = 1,
  parallel_chains = FALSE,
  group_col = NULL,
  seed = NULL,
  verbose = TRUE,
  ...
)

```

Arguments

<code>x</code>	An object containing the predictors (subgroup assignment). Can be a data frame or a formula.
<code>...</code>	Additional arguments passed to methods.
<code>formula</code>	A model formula with structure: 'Surv(time, event) ~ group + covariates'. The first RHS variable is the group/basket assignment. Additional variables are covariates.
<code>data</code>	A data frame containing the variables in the formula.
<code>pooling</code>	Pooling mode for the basket effects: one of "exnex" (default), "complete", or "none" (see Description).
<code>priors</code>	Optional named list of prior hyperparameters. Supported fields: <code>a_sigma</code> , <code>b_sigma</code> (inverse-Gamma shape and scale for the residual variance σ^2); <code>a_tau</code> , <code>b_tau</code> (inverse-Gamma for the between-basket variance τ^2 . Only used with <code>pooling = "exnex"</code>); <code>p_mix</code> (EXNEX mixture weight, strictly between 0 and 1. Only used with <code>pooling = "exnex"</code>); <code>m_mu</code> , <code>v_mu</code> (prior mean and variance of the exchangeable center μ ; with <code>pooling = "complete"</code> they act as the prior of the shared basket effect); <code>m_nex</code> , <code>v_nex</code> (prior mean and variance of the nonexchangeable component; with <code>pooling = "none"</code> they are the priors of the independent basket effects); <code>v_beta</code> (variance of the regression-coefficient prior).

	p_mix, m_nex, and v_nex each accept either a scalar, replicated across baskets, or a numeric vector of length K with one value per basket, matching the basket-specific notation of the model. All other fields are scalars. Absent fields keep the defaults (inverse-Gamma(2,2) for the variances, p_mix = 0.5, m_mu = 0, v_mu = 1e4, m_nex = 0, v_nex = 1e4, v_beta = 1e4); unknown fields are ignored. Priors that are irrelevant for the selected pooling mode are ignored with a warning.
iter	Total number of MCMC iterations. Default is 2000.
warmup	Number of warmup iterations to discard. Default is 1000. Posterior samples will have 'ceiling((iter - warmup) / thin)' rows per chain.
thin	Thinning interval applied after warmup: keep every 'thin'-th post-warmup draw. Default is 1 (keep all draws). Use 'thin > 1' to reduce memory and autocorrelation.
chains	Number of independent MCMC chains. Default is 1.
parallel_chains	Logical. If 'TRUE', all 'chains' run concurrently on background R sessions (via the 'future' framework); if 'FALSE' (default), chains run sequentially, one after the other.
group_col	Name of the column in 'x' that represents the basket/group assignment. If NULL (default), assumes the first column in x is the group.
seed	Random seed for reproducibility (optional).
verbose	If TRUE (default), show a live progress bar ('progressr') that advances as the chains run, in both sequential and parallel execution. Silence it with 'verbose = FALSE'.
y	A Surv object or matrix containing outcome (time and event status).

Details

For patient i , the model is

$$\log T_i = \theta_{g[i]} + X_i^T \beta + \varepsilon_i, \quad \varepsilon_i \sim \mathcal{N}(0, \sigma^2),$$

and the argument pooling selects the prior structure of the basket effects θ_j :

"exnex" (default) latent EXNEX hierarchy,

$$\theta_j \mid Z_j \sim Z_j \mathcal{N}(\mu, \tau^2) + (1 - Z_j) \mathcal{N}(m_{0j}, v_{0j}), \quad Z_j \sim \text{Bern}(p_{\text{exch},j}),$$

so that each basket either borrows strength from the exchangeable component $\mathcal{N}(\mu, \tau^2)$ or follows its own non-exchangeable prior $\mathcal{N}(m_{0j}, v_{0j})$.

"complete" complete pooling: a single shared basket effect $\theta_j = \theta$ for all baskets with prior $\theta \sim \mathcal{N}(m_\mu, v_\mu)$.

"none" no pooling: independent basket effects, $\theta_j \sim \mathcal{N}(m_{0j}, v_{0j})$ for $j = 1, \dots, K$.

The first predictor variable in the model formula is interpreted as the basket/group assignment; additional variables are covariates in the linear predictor on the log-survival scale.

Censored event times are handled by data augmentation: a censored log-time is imputed from its truncated-Normal conditional distribution (inverse-CDF in log space) before the remaining parameters are updated with their conjugate full conditionals. The sampler keeps only the posterior draws of θ_j , β , and σ^2 .

Value

An object of class 'pooling_surv' containing posterior samples and metadata. Components include pooling (the fitted variant), draws (a data frame with columns theta_1, ..., theta_K, beta_1, ..., beta_P, and sigma2; under pooling = "complete" the theta_* columns hold identical draws from the shared effect), data (the processed data: time, event, group, X, n, n_groups, n_covariates, cov_names, chain_seeds), priors (the supplied priors), resolved_priors (defaults merged with overrides), iter, warmup, chains, and blueprint. Use summary(), print(), and plot() to inspect it.

Examples

```
# Small simulated dataset: three baskets, one covariate, ~25% censoring
set.seed(1)
n <- 90
group <- factor(rep(1:3, each = 30))
x1 <- rnorm(n)
eta <- rep(c(1.1, 1.6, 2.0), each = 30) + 0.5 * x1
true_time <- exp(eta + rnorm(n, 0, 0.6))
cens_time <- runif(n, 2, 9)
d <- data.frame(
  time = pmin(true_time, cens_time),
  event = as.integer(true_time <= cens_time),
  group = group,
  x1 = x1
)

# Fit the EXNEX hierarchy with group effects and one covariate
fit <- pooling_surv(
  survival::Surv(time, event) ~ group + x1,
  data = d,
  priors = list(p_mix = 0.7),
  iter = 300, warmup = 150, chains = 1
)
print(fit, show_trace = FALSE)
summary(fit)

# Complete and no pooling use the same data augmentation machinery
fit_complete <- pooling_surv(
  survival::Surv(time, event) ~ group + x1,
  data = d, pooling = "complete",
  iter = 300, warmup = 150, chains = 1
)
fit_none <- pooling_surv(
  survival::Surv(time, event) ~ group + x1,
  data = d, pooling = "none",
  iter = 300, warmup = 150, chains = 1
)
```

posterior_predictive_check

Posterior predictive survival curves

Description

Posterior predictive check of the fitted log-normal AFT model: the posterior predictive mean survival curve per group (averaged over all posterior draws) is overlaid with the observed Kaplan-Meier curve and a band of replicated Kaplan-Meier curves. Systematic separation of the observed KM from the posterior mean band indicates model misfit for that group.

Usage

```
posterior_predictive_check(
  fit,
  times = NULL,
  n_replicates = 200,
  newdata = NULL,
  plot = TRUE,
  ...
)
```

Arguments

<code>fit</code>	A fitted ‘pooling_surv’ object.
<code>times</code>	Optional vector of evaluation times; defaults to a grid of 100 points between 0 and the largest observed time.
<code>n_replicates</code>	Number of posterior draws used for the replicated Kaplan-Meier band. Default 200.
<code>newdata</code>	Optional data frame with one row per group giving the covariates for the predictive curves. If ‘NULL’, covariates are zero.
<code>plot</code>	Should a ‘ggplot2’ figure be drawn? Default ‘TRUE’.
<code>...</code>	Unused.

Value

Invisibly returns a list with ‘posterior’ (data frame: ‘group’, ‘time’, ‘mean’, ‘lwr’, ‘upr’) and ‘observed’ (data frame: ‘group’, ‘time’, ‘surv’). Requires ‘ggplot2’ when ‘plot = TRUE’.

Examples

```
set.seed(2719)
trial_data <- simulate_data(n = 90, K = 3, censoring_rate = 0.3)
fit <- pooling_surv(survival::Surv(time, event) ~ group,
  data = trial_data,
  iter = 300, warmup = 150, chains = 2)
## Not run: posterior_predictive_check(fit)
```

```
print.pooling_surv      Print an pooling_surv fit
```

Description

Prints a compact summary and optionally traceplots of posterior draws.

Usage

```
## S3 method for class 'pooling_surv'
print(x, show_trace = TRUE, parameters = NULL, max_parameters = Inf, ...)
```

Arguments

<code>x</code>	A fitted 'pooling_surv' object.
<code>show_trace</code>	Should traceplots be shown? Default is 'TRUE'.
<code>parameters</code>	Optional character vector of parameter names for traceplots.
<code>max_parameters</code>	Maximum number of parameters to plot.
<code>...</code>	Unused.

Value

Invisibly returns 'x'.

```
probability_superiority
      Posterior probability that one group beats another
```

Description

Estimates the posterior probability that a certain scalar summary of the survival distribution for group 'a' is greater than the same summary for group 'b'. Supported summaries are the median survival time, the survival probability at a fixed time 'S(t_0)', and the restricted mean survival time (RMST). The function works draw-by-draw on the posterior, so the probability is computed on the joint posterior of the two groups.

Usage

```
probability_superiority(
  fit,
  a,
  b,
  function_of = "median",
  newdata = NULL,
```

```

    times = NULL,
    tmax = NULL,
    ...
)

```

Arguments

fit	A fitted ‘pooling_surv’ object.
a	An integer index (or group label) of the first group.
b	An integer index (or group label) of the second group.
function_of	Character; one of “median”, “survival” (alias “S_t”), or “rmst”. Selects which summary is compared.
newdata	Optional data frame with one row per group, used to fix covariates. If ‘NULL’, covariates are set to zero.
times	A scalar time at which to evaluate the survival function when ‘function_of = “survival”’.
tmax	A positive horizon when ‘function_of = “rmst”’.
...	Unused.

Value

A named list with elements ‘prob’ (posterior probability that the summary of group ‘a’ exceeds that of group ‘b’), ‘summary’ (which summary was used), and ‘groups’ (the two groups compared).

rank_probabilities	<i>Posterior rank probabilities for each group</i>
--------------------	--

Description

For every posterior draw, the groups are ranked by a chosen survival summary (median survival, survival at a fixed time, or RMST) and the frequency of each rank is returned. Useful as a decision aid: the row of the output for rank 1 gives the posterior probability that each group is the best.

Usage

```

rank_probabilities(
  fit,
  newdata = NULL,
  function_of = "median",
  times = NULL,
  tmax = NULL,
  ...
)

```

Arguments

<code>fit</code>	A fitted ‘pooling_surv’ object.
<code>newdata</code>	Optional data frame with one row per group used to fix covariates. If ‘NULL’, covariates are set to zero.
<code>function_of</code>	Character; one of “median”, “survival” (alias “S_t”), or “rmst”.
<code>times</code>	Scalar time when ‘function_of = “survival”’.
<code>tmax</code>	Positive horizon when ‘function_of = “rmst”’.
<code>...</code>	Unused.

Value

A data frame with columns ‘group’, ‘rank’, and ‘prob’. Rank 1 is the largest summary value (best group).

Examples

```
set.seed(2719)
trial_data <- simulate_data(n = 90, K = 3, censoring_rate = 0.3)
fit <- pooling_surv(survival::Surv(time, event) ~ group,
  data = trial_data,
  iter = 300, warmup = 150, chains = 2)
rank_probabilities(fit)
```

rmst

*Restricted mean survival time (RMST) from an pooling_surv fit***Description**

Computes the restricted mean survival time up to a horizon ‘tmax’:

$$RMST(tmax) = \int_0^{tmax} S(t) dt.$$

The integral is evaluated numerically (trapezoidal rule over a fine grid) for every posterior draw, then summarised with posterior quantiles.

Usage

```
rmst(fit, tmax = NULL, newdata = NULL, level = 0.95, grid_points = 400, ...)
```

Arguments

<code>fit</code>	A fitted ‘pooling_surv’ object.
<code>tmax</code>	A positive scalar horizon up to which the RMST is computed. Defaults to ‘max(fit\$data\$time)’.
<code>newdata</code>	Optional data frame (one row gives one RMST).
<code>level</code>	Credible-interval level (default ‘0.95’).
<code>grid_points</code>	Resolution of the numerical quadrature (default ‘400’).
<code>...</code>	Unused.

Value

A 'data.frame' with columns 'group', 'rmst', 'lower', 'upper'.

shrinkage_plot

Visualize shrinkage of group effects

Description

Compares the hierarchical (EXNEX posterior) estimates of the group-specific log-median survival 'theta_j' (at zero covariates) against the unpooled, basket-only Kaplan-Meier median estimate computed from the original data. A short segment between the two points per group shows the amount of shrinkage (partial pooling) applied by the model.

Usage

```
shrinkage_plot(fit, ...)
```

Arguments

fit	A fitted 'pooling_surv' object.
...	Unused.

Value

Invisibly returns a data frame with columns 'group', 'posterior_mean', 'posterior_lwr', 'posterior_upr', and 'unpooled' – the log-scale values used in the plot. Requires 'ggplot2'.

Examples

```
set.seed(2719)
trial_data <- simulate_data(n = 90, K = 3, censoring_rate = 0.3)
fit <- pooling_surv(survival::Surv(time, event) ~ group,
                   data = trial_data,
                   iter = 300, warmup = 150, chains = 2)
## Not run: shrinkage_plot(fit)
```

simulate_data

*Simulate basket-trial log-normal survival data***Description**

Generates a synthetic dataset from the log-normal AFT model used by ‘exnexSurv’:

$$\log T_i = \theta_{g[i]} + X_i^\top \beta + \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, \sigma^2).$$

Healthy baskets draw their location from a common baseline plus random noise; "outlier" baskets (specified via ‘outlier_baskets’) have their location shifted by ‘resist_delta’. The function is useful for simulation studies and teaching examples.

Usage

```
simulate_data(
  n = 30,
  K = 9,
  beta = c(0.5, -0.2),
  sigma = 1.2,
  outlier_baskets = NULL,
  resist_delta = -0.8,
  censoring_rate = NULL,
  censor_upper = NULL,
  theta = 0,
  seed = NULL
)
```

Arguments

n	Number of patients per basket. A scalar is replicated across all ‘K’ baskets; a numeric vector of length ‘K’ assigns a size to each basket individually.
K	Number of baskets (default ‘9’).
beta	Numeric vector of regression coefficients for the covariates. Length determines the number of covariates.
sigma	Residual standard deviation (default ‘1.2’).
outlier_baskets	Optional integer vector of basket indices (1 to K) whose true location is shifted away from the healthy population. These are the baskets the EXNEX model is designed to detect.
resist_delta	Additive shift applied to the ‘theta’ of outlier baskets (default ‘-0.8’).
censoring_rate	Approximate proportion of censoring after ‘censor_upper’; if ‘NULL’, no censoring is applied.
censor_upper	Upper bound of the censoring-time uniform distribution.
theta	Baseline location for the healthy (non-outlier) baskets; a scalar used as the centre around which the healthy basket locations vary.
seed	Optional seed for reproducibility.

Value

A ‘data.frame’ with columns ‘time’, ‘event’, ‘group’ (a factor), and one covariate column (‘x1’, ...) per entry in ‘beta’. True parameter values are stored as attributes ‘true_theta’, ‘true_beta’, and ‘true_sigma’.

summary.pooling_surv	<i>Summarize posterior draws from an pooling_surv fit</i>
----------------------	---

Description

Summarize posterior draws from an pooling_surv fit

Usage

```
## S3 method for class 'pooling_surv'
summary(object, probs = c(0.05, 0.5, 0.95), ...)
```

Arguments

object	A fitted ‘pooling_surv’ object.
probs	Quantiles to report.
...	Unused.

Value

A data frame with posterior summaries.

survival_curves	<i>Survival curves from an pooling_surv fit</i>
-----------------	---

Description

Computes posterior survival curves $S(t) = \Pr(T > t)$ from a fitted ‘pooling_surv’ model for a log-normal AFT specification:

$$\log T_i = \theta_{g[i]} + X_i^\top \beta + \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, \sigma^2).$$

For a fixed time t and linear predictor η , the survival probability of the log-normal distribution is

$$S(t) = 1 - \Phi\left(\frac{\log t - \eta}{\sigma}\right),$$

where σ^2 is the residual variance. Posterior draws of $(\theta, \beta, \sigma^2)$ are propagated through this expression to obtain a full posterior distribution of $S(t)$ at each time point.

Usage

```
survival_curves(fit, newdata = NULL, times = NULL, level = 0.95, ...)
```

Arguments

<code>fit</code>	A fitted 'pooling_surv' object.
<code>newdata</code>	Optional data frame with columns matching the covariates of the model. If it contains a 'group' column, that is used for the group index; otherwise the first group is used for all rows.
<code>times</code>	Optional numeric vector of times at which to evaluate the curves. If 'NULL', a sensible grid is built from the observed follow-up times.
<code>level</code>	Credible-interval level (default '0.95').
<code>...</code>	Unused.

Details

If 'newdata' is supplied, each row is evaluated at its own group and covariate values; otherwise the covariates are fixed at zero and the first group is used (with a warning if more than one group exists).

Value

An object of class 'survival_exnex' (also a 'data.frame') with long format columns: 'time', 'median', 'lower', 'upper', and 'group'.

Index

`compare_waic`, [2](#)
`compute_waic`, [3](#)
`convergence_diagnostics`, [4](#)

`exnex_surv`, [5](#)

`median_survival`, [5](#)

`plot.pooling_surv`, [6](#)
`plot.survival_exnex`, [7](#)
`pooling_surv`, [7](#)
`posterior_predictive_check`, [11](#)
`print.pooling_surv`, [12](#)
`probability_superiority`, [12](#)

`rank_probabilities`, [13](#)
`rmst`, [14](#)

`shrinkage_plot`, [15](#)
`simulate_data`, [16](#)
`summary.pooling_surv`, [17](#)
`survival_curves`, [17](#)