Package: GammaFrailty (via r-universe)

June 19, 2026

Type Package

Title Gamma Frailty Regression Models with Multiple Baseline Distributions

Version 0.1.0

Description Implements univariate gamma frailty regression models for survival data with six different baseline distributions: the Arvind distribution (Pandey et al., 2024), the Lindley distribution (Lindley, 1958), the Linear Failure Rate distribution (Bain, 1974), the Power Xgamma distribution (Tyagi et al., 2022), the Modified Topp-Leone distribution (Singh et al., 2025), and the Power Failure Rate distribution (Mugdadi, 2005). The package supports uncensored (complete) and censored data (right, left, interval, and progressive censoring) with and without covariates. It provides maximum likelihood estimation, standard errors, confidence intervals, t-statistics, p-values, Akaike Information Criterion (AIC), Bayesian Information Criterion (BIC), a bootstrap approximation of the Widely Applicable Information Criterion (WAIC), k-fold cross-validation, variance inflation factors, R-squared, adjusted R-squared, Mean Squared Error (MSE), Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), an overall model F-test, frailty variance estimation, survival probabilities at user-specified time points, median survival, expected survival within a fixed window, risk predictions, marginal predictions, martingale and deviance residuals, standardized and studentized residuals, leverage values, Cook's distance, Difference in Fits (DFFITS), Difference in Betas (DFBETAS), and a comprehensive suite of diagnostic and survival plots including Kaplan-Meier overlays and coefficient forest plots. Random number generation is available for each baseline distribution and the full frailty model, and a simulation study function evaluates parameter recovery across sample sizes and censoring scenarios. References are Lindley (1958) <doi:10.1111/j.2517-6161.1958.tb00278.x>, Mugdadi (2005) <doi:10.1016/j.amc.2004.09.064>, Bain (1974)

<doi:10.1080/00401706.1974.10489237>, Singh, Tyagi, Singh, and Tyagi (2025) <`https://ph02.tci-thaijo.org/index.php/thaistat/article/view/257215`>, Pandey, Singh, Tyagi, and Tyagi (2024) <`https://ssca.org.in/journal.html`>, and Tyagi, Kumar, Pandey, Saha, and Bagariya (2022) <`https://ijsreg.com/`>.

Depends R (>= 4.0.0)

Imports survival, maxLik, numDeriv, MASS, stats, graphics, grDevices, utils

Suggests testthat (>= 3.0.0), knitr, rmarkdown

Config/testthat/edition 3

Encoding UTF-8

Language en-US

RoxygenNote 7.3.3

VignetteBuilder knitr

NeedsCompilation no

Author Shikhar Tyagi [aut, cre] (ORCID: <`https://orcid.org/0000-0003-1606-0844`>)

Maintainer Shikhar Tyagi <`shikhar1093tyagi@gmail.com`>

Repository https://shikhartyagi.r-universe.dev

Date/Publication 2026-06-18 17:00:02 UTC

RemoteUrl https://github.com/cran/GammaFrailty

RemoteRef HEAD

RemoteSha 6d176dd1c53552f82bc75d328891d632bf77f5c7

Contents

`baseline_hazard` *Baseline Cumulative Hazard and Hazard Functions*

Description

Computes the baseline cumulative hazard $H_0(t)$ and baseline hazard $h_0(t)$ for the six supported distributions.

Usage

```
baseline_hazard(t, baseline, par)
```

Arguments

`t`	positive numeric vector of time points.
`baseline`	character; baseline distribution name. One of `"arvind"`, `"lindley"`, `"lfr"`, `"pxg"`, `"mtl"`, `"pfr"`.
`par`	numeric vector of baseline parameters (see Details).

Details

Arvind (`par = c(alpha)`): $H_0(t) = \alpha t^2 + \log(1 + \alpha t)$, $h_0(t) = 2\alpha t + \alpha/(1 + \alpha t)$.

Lindley (`par = c(lambda)`): $H_0(t) = \lambda t - \log(1 + \lambda t/(1 + \lambda))$, $h_0(t) = \lambda^2(1 + t)/(1 + \lambda + \lambda t)$.

LFR (`par = c(a, b)`): $H_0(t) = at + bt^2/2$, $h_0(t) = a + bt$.

Power Xgamma (`par = c(alpha, beta)`): $H_0(t) = \alpha t^{1/\beta} + \log(1 + \alpha) - \log(1 + \alpha + \alpha t^{1/\beta} + \alpha^2 t^{2/\beta}/2)$.

Modified Topp-Leone (`par = c(alpha)`): $H_0(t) = -\log\Big(1 - \Big(\frac{2t+t^2}{(1+t)^2}\Big)^{\alpha}\Big)$.

Power Failure Rate (`par = c(a, k)`): $H_0(t) = \frac{a}{k+1} t^{k+1}$, $h_0(t) = at^k$.

Value

A list with components:

H0 Numeric vector of cumulative hazard values.

h0 Numeric vector of hazard values.

Examples

```
bh <- baseline_hazard(seq(0.1, 2, by = 0.1), baseline = "arvind", par = 0.5)
plot(seq(0.1, 2, by = 0.1), bh$h0, type = "l", main = "Arvind hazard")
```

`bootstrap_waic` *Bootstrap WAIC for the Gamma Frailty Model*

Description

Approximates the Widely Applicable Information Criterion (WAIC) via bootstrap re-sampling of the log-likelihood. Since the package uses frequentist MLE, this serves as a computationally tractable approximation to the Bayesian WAIC.

Usage

```
bootstrap_waic(fit, B = 200)
```

Arguments

`fit`	An object of class `"gamma_frailty_fit"`.
`B`	integer; number of bootstrap iterations (default 200).

Value

A named list:

WAIC $-2 \times (\text{lppd} - p_{\text{WAIC}})$.

lppd Log pointwise predictive density.

p_waic Effective number of parameters.

B_effective Number of successful bootstrap draws used.

Examples

```
set.seed(1)
dat <- r_gamma_frailty(80, "arvind", par = 0.5, theta = 0.3,
                        cen_type = "right")
fit <- fit_gamma_frailty(dat$time, dat$status, baseline = "arvind")
bootstrap_waic(fit, B = 50)
```

`compare_models` *Compare Multiple Gamma Frailty Models*

Description

Compiles AIC, BIC, log-likelihood, frailty variance, and (optionally) WAIC for a list of fitted gamma frailty models.

Usage

```
compare_models(..., compute_waic = FALSE, waic_B = 100L)
```

Arguments

`...`	named objects of class `"gamma_frailty_fit"`. Names are used as row labels in the output table. You may also pass a single named list.
`compute_waic`	logical; if `TRUE`, compute bootstrap WAIC for each model (slow; default `FALSE`).
`waic_B`	integer; bootstrap iterations for WAIC (default 100).

Value

A data frame with columns `Baseline`, `logLik`, `K`, `AIC`, `BIC`, `theta`, and optionally `WAIC`.

Examples

```
set.seed(8)
dat <- r_gamma_frailty(100, "arvind", par = 0.5, theta = 0.3,
                        cen_type = "right")
f1 <- fit_gamma_frailty(dat$time, dat$status, baseline = "arvind")
f2 <- fit_gamma_frailty(dat$time, dat$status, baseline = "lindley")
compare_models(Arvind = f1, Lindley = f2)
```

`cv_frailty` *K-Fold Cross-Validation for the Gamma Frailty Model*

Description

Performs k-fold cross-validation and reports out-of-sample log-likelihood and RMSE.

Usage

```
cv_frailty(
  time,
  status,
  x = matrix(nrow = length(time), ncol = 0),
  baseline = "arvind",
  k = 5L,
  time2 = NULL
)
```

Arguments

`time`	positive numeric vector of event/censoring times.
`status`	integer vector of censoring indicators.
`x`	numeric matrix of covariates (may have zero columns).
`baseline`	character; baseline distribution.
`k`	integer; number of folds (default 5).
`time2`	numeric vector; upper bound for interval-censored obs.

Value

A named list:

mean_oos_loglik Mean out-of-sample log-likelihood.

sd_oos_loglik Standard deviation across folds.

mean_oos_rmse Mean out-of-sample RMSE.

sd_oos_rmse SD of RMSE across folds.

fold_logliks, fold_rmses Per-fold values.

Examples

```
set.seed(7)
dat <- r_gamma_frailty(120, "lfr", par = c(0.5, 0.2), theta = 0.4,
                       cen_type = "right")
cv_frailty(dat$time, dat$status, baseline = "lfr", k = 5)
```

`diagnostics_table` *Summary Diagnostics Table*

Description

Returns a consolidated data frame of all diagnostic measures for each observation, suitable for identifying outliers and influential points.

Usage

```
diagnostics_table(fit)
```

Arguments

`fit`	An object of class "`gamma_frailty_fit`".

Value

A data frame with columns: `time`, `status`, `cox_snell`, `martingale`, `deviance`, `raw`, `standardized`, `studentized`, `leverage`, `cooks_distance`, `DFFITS`, and one column per covariate for DFBETAS.

Examples

```
set.seed(3)
dat <- r_gamma_frailty(60, "pfr", par = c(0.5, 1), theta = 0.3,
                        cen_type = "right")
fit <- fit_gamma_frailty(dat$time, dat$status, baseline = "pfr")
diag_tbl <- diagnostics_table(fit)
head(diag_tbl)
```

`fit_gamma_frailty` *Fit a Gamma Frailty Regression Model*

Description

Fits a gamma frailty regression model with the chosen baseline distribution via maximum likelihood estimation (MLE). Handles uncensored and right-, left-, interval-, and progressive-censored data, with and without covariates.

Usage

```
fit_gamma_frailty(
  time,
  status,
  x = matrix(nrow = length(time), ncol = 0),
  baseline = "arvind",
  time2 = NULL,
  prog_cen = NULL,
  init = NULL
)
```

Arguments

`time`	positive numeric vector of event or censoring times.
`status`	integer vector indicating the observation type: `1` = exact event, `0` = right-censored, `2` = left-censored, `3` = interval-censored.
`x`	numeric matrix or data frame of covariates. Use an empty matrix (zero columns) for a no-covariate model. Default: `matrix(nrow = length(time), ncol = 0)`.
`baseline`	character; baseline distribution. One of `"arvind"`, `"lindley"`, `"lfr"`, `"pxg"`, `"mtl"`, `"pfr"`.
`time2`	numeric vector; upper bound for interval-censored observations (required when any `status == 3`).
`prog_cen`	integer vector; number of items progressively removed at the j-th failure time (progressive censoring adjustment).
`init`	numeric vector of initial parameter values on the log scale for baseline parameters and log scale for θ. If `NULL`, sensible defaults are used.

Value

An object of class `"gamma_frailty_fit"` (a named list) with components:

coefficients Data frame with columns Estimate, StdErr, t_stat, p_value, CI_lower, CI_upper, Signif.

logLik Maximised log-likelihood.

AIC Akaike Information Criterion.

BIC Bayesian Information Criterion.

vcov Variance-covariance matrix of estimates (log scale).

baseline Name of the baseline distribution.

theta Estimated frailty variance.

theta_se Standard error of the frailty variance (delta method).

VIF Named numeric vector of variance inflation factors (when `ncol(x)` > 1).

Tolerance Tolerance = 1/VIF.

F_stat Wald F-statistic for overall covariate significance.

p_F_stat P-value for the F-statistic.

n Sample size.

n_cov Number of covariates.

n_par_base Number of baseline parameters.

time, status, x, time2, prog_cen Input data stored for post-fit diagnostics.

raw_est Named numeric vector of estimates on the log/original scale used internally for prediction and diagnostics.

Examples

```
set.seed(1)
dat <- r_gamma_frailty(100, "arvind", par = 0.5, theta = 0.3,
                       cen_type = "right", cen_rate = 0.2)
fit <- fit_gamma_frailty(dat$time, dat$status, baseline = "arvind")
summary(fit)
```

`forecast_frailty` *Forecast Survival Curves*

Description

Extends the model-based survival curve beyond the observed training range to a specified forecast horizon.

Usage

```
forecast_frailty(fit, horizon, n_grid = 200, newdata = NULL)
```

Arguments

`fit`	An object of class "`gamma_frailty_fit`".
`horizon`	positive numeric; the maximum time to forecast to.
`n_grid`	integer; number of equally spaced time points in `[0, horizon]` to evaluate (default 200).
`newdata`	optional covariate matrix for new subjects.

Value

A list with `time` (grid of time points) and `survival` (matrix, subjects x n_grid).

Examples

```
set.seed(1)
dat <- r_gamma_frailty(80, "arvind", par = 0.5, theta = 0.3,
                        cen_type = "right")
fit <- fit_gamma_frailty(dat$time, dat$status, baseline = "arvind")
fc  <- forecast_frailty(fit, horizon = 5)
plot(fc$time, fc$survival[1, ], type = "l", xlab = "Time",
     ylab = "S(t)", main = "Forecast")
```

`gamma_frailty` *Gamma Frailty Regression Model (Formula Interface)*

Description

A formula-based interface to `fit_gamma_frailty`, modelled after `lm` and `glm`. The response must be a `Surv` object.

Usage

```
gamma_frailty(
  formula,
  data,
  baseline = "arvind",
  time2 = NULL,
  prog_cen = NULL,
  init = NULL
)
```

Arguments

`formula`	a `formula` object. The left-hand side must be a `Surv(time, status)` or `Surv(time, time2, status, type = "interval")` object.
`data`	a data frame containing the variables in the formula.
`baseline`	character; baseline distribution. One of `"arvind"`, `"lindley"`, `"lfr"`, `"pxg"`, `"mtl"`, `"pfr"`.
`time2`	numeric vector; upper bound for interval-censored observations.
`prog_cen`	integer vector; progressive censoring scheme.
`init`	numeric vector; initial parameter values.

Value

An object of class `c("gamma_frailty", "gamma_frailty_fit")` with an additional `call` and `formula` component.

Examples

```
set.seed(1)
dat <- r_gamma_frailty(100, "arvind", par = 0.5, theta = 0.3,
                        x = matrix(rnorm(100), ncol = 1),
                        beta = 0.5, cen_type = "right", cen_rate = 0.2)
colnames(dat)[4] <- "X1"
fit <- gamma_frailty(survival::Surv(time, status) ~ X1,
                      data = dat, baseline = "arvind")
summary(fit)
```

`gamma_frailty_functions` *Gamma Frailty Model Functions*

Description

Computes the survival function $S(t)$, density $f(t)$, hazard $h(t)$, and cumulative hazard $\mathcal{H}(t)$ for the gamma frailty model.

Usage

```
gamma_frailty_functions(t, eta, theta, baseline, par)
```

Arguments

`t`	positive numeric vector of time points.
`eta`	positive numeric vector of linear predictors $\eta_j = e^{X_j \beta}$. Must have the same length as `t` or be a scalar.
`theta`	positive numeric; frailty variance.
`baseline`	character; baseline distribution name.
`par`	numeric vector of baseline parameters.

Details

The marginal survival function (after integrating out the gamma frailty) is

$$S(t_j) = [1 + \theta \eta_j H_0(t_j)]^{-1/\theta} .$$

The corresponding density, hazard, and cumulative hazard are derived analytically from this expression.

Value

A list with components `S`, `f`, `h`, `H`.

Examples

```
gf <- gamma_frailty_functions(1:5, eta = 1, theta = 0.5,
                               baseline = "arvind", par = 0.5)
plot(1:5, gf$S, type = "l", main = "Survival")
```

`influence_frailty` *Influence Diagnostics for the Gamma Frailty Model*

Description

Computes leverage values, Cook's distance, DFFITS, and DFBETAS for identifying outliers and influential observations.

Usage

```
influence_frailty(fit)
```

Arguments

`fit`	An object of class "`gamma_frailty_fit`".

Value

A named list with components:

leverage Approximate hat values h_{ii}.

std_residuals Standardized deviance residuals.

stu_residuals Studentized deviance residuals (LOO approx).

cooks_distance Cook's distance D_i.

DFFITS DFFITS values.

DFBETAS Matrix of DFBETAS, one column per covariate.

Examples

```
set.seed(2)
dat <- r_gamma_frailty(80, "lfr", par = c(0.5, 0.2), theta = 0.4,
                        x = matrix(rnorm(80), ncol = 1),
                        beta = 0.5, cen_type = "right")
fit <- fit_gamma_frailty(dat$time, dat$status, dat[, 4, drop = FALSE],
                          baseline = "lfr")
inf <- influence_frailty(fit)
plot(inf$cooks_distance, type = "h", main = "Cook's Distance")
```

`loglik_gamma_frailty` *Log-Likelihood for the Gamma Frailty Model*

Description

Internal function computing the total log-likelihood for uncensored and censored (right, left, interval, progressive) data.

Usage

```
loglik_gamma_frailty(
  par_all,
  time,
  status,
  x,
  baseline,
  time2 = NULL,
  prog_cen = NULL
)
```

Arguments

`par_all`	numeric vector of ALL parameters on the estimation scale (log-transformed for positivity constraints): `c(log(baseline_pars), beta_covariates, log(theta))`.
`time`	numeric vector of event or censoring times.
`status`	integer vector: 1 = exact event, 0 = right-censored, 2 = left-censored, 3 = interval-censored.
`x`	numeric matrix of covariates (can have zero columns).
`baseline`	character; baseline distribution name.
`time2`	numeric vector; upper bound for interval-censored observations (`status == 3`). Ignored otherwise.
`prog_cen`	integer vector; number of items progressively removed at each observed failure time. If `NULL`, no progressive adjustment.

Value

Scalar log-likelihood value (or `-1e12` if infeasible).

`plot_all` *Plot All Diagnostics*

Description

Generates all 8 diagnostic plots for a fitted gamma frailty model and displays them in the **active R graphics device** (e.g., the RStudio Plots pane). Plots are **never** saved to a file automatically. To save, wrap the call in `pdf()` / `dev.off()` yourself, or use the `save_to_file` argument.

Usage

```
plot_all(fit, ask = grDevices::dev.interactive(), save_to_file = NULL)
```

Arguments

`fit`	An object of class `"gamma_frailty_fit"`.
`ask`	logical; if `TRUE` (the default in interactive sessions) R pauses and waits for the user to press Enter before drawing each new plot. Set `ask = FALSE` to cycle through all plots without pausing.
`save_to_file`	character or `NULL` (default). If a file path is supplied (e.g., `"diagnostics.pdf"` or `"plot.png"`), the 8 plots are written to that file instead of the screen. The file type is detected from the extension (`.pdf`, `.png`, `.svg`, `.jpeg` / `.jpg`). When `NULL` (default) the plots go to the active device only.

Value

Invisibly returns `fit`.

Note

This function **never** saves plots automatically. The `save_to_file` argument exists purely for user convenience and is `NULL` by default.

Examples

```
set.seed(1)
dat <- r_gamma_frailty(80, "arvind", par = 0.5, theta = 0.3,
                        cen_type = "right", cen_rate = 0.2)
fit <- fit_gamma_frailty(dat$time, dat$status, baseline = "arvind")
# Display all 8 plots in the R graphics window (no file saved):
 plot_all(fit, ask = FALSE)
```

`plot_baseline` *Plot Baseline Distribution Functions*

Description

Plots the PDF, CDF, survival function, and hazard function for a specified baseline distribution over a time grid.

Usage

```
plot_baseline(baseline, par, t_range = c(0.01, 3), n_grid = 300)
```

Arguments

`baseline`	character; baseline distribution name.
`par`	numeric vector of baseline parameters.
`t_range`	numeric vector of length 2 giving the time range.
`n_grid`	integer; number of grid points (default 300).

Value

Invisibly returns a list with components `t` (time grid), `f` (PDF), `F` (CDF), `S` (survival), and `h` (hazard) evaluated over the time grid.

Examples

```
plot_baseline("arvind", par = 0.5, t_range = c(0.01, 3))
```

`plot_coef_forest` *Coefficient Forest Plot*

Description

Displays all parameter estimates with 95% confidence intervals as a horizontal forest (dot-and-whisker) plot.

Usage

```
plot_coef_forest(fit, ...)
```

Arguments

`fit`	An object of class "`gamma_frailty_fit`".
`...`	additional graphical parameters.

Value

No return value, called for side effects (plotting).

`plot_dfbetas` *DFBETAS Dot Plot*

Description

Plots DFBETAS values as a stem plot for each covariate, highlighting observations that exceed the $2/\sqrt{n}$ threshold in red.

Usage

```
plot_dfbetas(fit, ...)
```

Arguments

`fit`	An object of class "`gamma_frailty_fit`".
`...`	additional graphical parameters.

Value

No return value, called for side effects (plotting).

`plot_leverage` *Leverage Histogram*

Description

Displays a histogram of leverage values with a threshold line at $2(p+1)/n$.

Usage

```
plot_leverage(fit, ...)
```

Arguments

`fit`	An object of class "`gamma_frailty_fit`".
`...`	additional graphical parameters.

Value

No return value, called for side effects (plotting).

`plot_qq_residuals` *Q-Q Plot of Cox-Snell Residuals*

Description

Compares sorted Cox-Snell residuals against theoretical quantiles of the standard exponential distribution. Deviations from the diagonal indicate model misfit.

Usage

```
plot_qq_residuals(fit, ...)
```

Arguments

`fit`	An object of class `"gamma_frailty_fit"`.
`...`	additional graphical parameters.

Value

No return value, called for side effects (plotting).

`plot_residuals_fitted` *Residuals vs Fitted Values Plot*

Description

Plots deviance residuals against fitted survival probabilities with a LOWESS smoother to assess linearity and heteroscedasticity.

Usage

```
plot_residuals_fitted(fit, ...)
```

Arguments

`fit`	An object of class `"gamma_frailty_fit"`.
`...`	additional graphical parameters passed to `plot`.

Value

No return value, called for side effects (plotting).

`plot_residuals_leverage` *Residuals vs Leverage Plot*

Description

Plots deviance residuals against leverage values with Cook's distance contours (D = 0.5 and D = 1) to identify influential points.

Usage

```
plot_residuals_leverage(fit, ...)
```

Arguments

`fit`	An object of class `"gamma_frailty_fit"`.
`...`	additional graphical parameters.

Value

No return value, called for side effects (plotting).

`plot_scale_location` *Scale-Location Plot*

Description

Plots the square root of absolute deviance residuals against fitted survival probabilities to detect heteroscedasticity.

Usage

```
plot_scale_location(fit, ...)
```

Arguments

`fit`	An object of class `"gamma_frailty_fit"`.
`...`	additional graphical parameters.

Value

No return value, called for side effects (plotting).

`plot_survival_km` *Kaplan-Meier vs Model-Based Survival Plot*

Description

Overlays the Kaplan-Meier curve (empirical) with the model-based survival curve evaluated at the mean covariate vector.

Usage

```
plot_survival_km(fit, ...)
```

Arguments

`fit`	An object of class `"gamma_frailty_fit"`.
`...`	additional graphical parameters.

Value

No return value, called for side effects (plotting).

`predict_frailty` *Predictions from a Gamma Frailty Model*

Description

Computes survival probabilities, hazard rates, median survival times, expected survival in a window, risk scores, marginal survival, and forecasted survival curves for a fitted gamma frailty model.

Usage

```
predict_frailty(
  fit,
  newdata = NULL,
  newtime = NULL,
 type = c("survival", "hazard", "median", "expected", "risk", "marginal", "forecast"),
  window = NULL
)
```

Arguments

`fit`	An object of class `"gamma_frailty_fit"`.
`newdata`	numeric matrix or data frame of covariates for prediction. If `NULL` the training covariates are used.
`newtime`	numeric vector of time points for prediction. If `NULL` the training times are used.
`type`	character; the type of prediction: `"survival"` Matrix of survival probabilities (subjects x times). `"hazard"` Matrix of hazard rates (subjects x times). `"median"` Numeric vector of median survival times. `"expected"` Expected survival probability drop in `window`. `"risk"` Risk score $\eta_j = \exp(X_j \hat{\beta})$. `"marginal"` Marginal survival (population-average, averaged over frailty and observed covariates). `"forecast"` Extended survival curve beyond the training range; use `newtime` to specify forecast horizon.
`window`	numeric vector of length 2 (t_1, t_2) giving the time interval for `type = "expected"`.

Value

Depends on `type`:

- `"survival"`, `"hazard"`, `"forecast"`: a list with element `survival` or `hazard` - a matrix (subjects x times).
- `"median"`: numeric vector (one value per subject).
- `"expected"`: numeric vector (one value per subject).
- `"risk"`: numeric vector of risk scores.
- `"marginal"`: numeric vector of population-average survival at each time in `newtime`.

Examples

```
set.seed(5)
dat <- r_gamma_frailty(100, "pxg", par = c(1, 0.8), theta = 0.4,
                       x = matrix(rnorm(200), ncol = 2),
                       beta = c(0.3, -0.2),
                       cen_type = "right", cen_rate = 0.2)
fit <- fit_gamma_frailty(dat$time, dat$status, dat[, 4:5],
                         baseline = "pxg")
# Survival at training times
pred_S <- predict_frailty(fit, type = "survival")
# Median survival
med    <- predict_frailty(fit, type = "median")
```

`r_arvind` *Random samples from the Arvind distribution*

Description

Generates `n` random observations from the Arvind distribution with parameter `alpha` using the inverse-CDF method. The survival function is $S(t) = \exp(-\alpha t^2)/(1 + \alpha t)$.

Usage

```
r_arvind(n, alpha)
```

Arguments

`n`	integer; number of observations.
`alpha`	positive numeric; shape/rate parameter.

Value

Numeric vector of length `n`.

References

Pandey, A., Singh, R. P., Tyagi, S., & Tyagi, A. (2024). Modelling climate, COVID-19, and reliability data: A new continuous lifetime model under different methods of estimation. Stat. Appl., 22(2).

Examples

```
set.seed(1)
x <- r_arvind(50, alpha = 0.5)
hist(x, main = "Arvind samples")
```

`r_gamma_frailty` *Generate Random Survival Times from the Gamma Frailty Model*

Description

Generates `n` random survival times from the specified gamma frailty model using the inverse-CDF method. Seven censoring mechanisms are supported:

`"none"` Complete (uncensored) data.

`"right"` Random right censoring with exponential censoring times at rate `cen_rate`.

`"left"` Left censoring at a fixed threshold `left_threshold`.

`"interval"` Interval censoring with window width `int_width`.

`"type1"` Type-I (time-terminated) censoring: all subjects observed up to a fixed time `cen_time`; those who have not failed by `cen_time` are right-censored at `cen_time`.

`"type2"` Type-II (failure-terminated) censoring: the experiment terminates at the `r_failures`-th observed failure; the remaining $n - r$ subjects are right-censored at that time.

`"progressive"` Progressive Type-II censoring: at each of the first `length(prog_scheme)` failure times, `prog_scheme[j]` surviving units are randomly withdrawn.

`"progressive_type1"` Progressive Type-I censoring: at each pre-specified time in `prog_times`, `prog_scheme[j]` surviving units are randomly withdrawn; the rest are observed until failure.

Usage

```
r_gamma_frailty(
  n,
  baseline,
  par,
  x = matrix(nrow = n, ncol = 0),
  beta = numeric(0),
  theta,
  cen_type = c("none", "right", "left", "interval", "type1", "type2", "progressive",
    "progressive_type1"),
  cen_rate = 0.2,
  left_threshold = NULL,
  int_width = NULL,
  cen_time = NULL,
  r_failures = NULL,
  prog_scheme = NULL,
  prog_times = NULL
)
```

Arguments

`n`	integer; number of subjects.
`baseline`	character; baseline distribution name. One of `"arvind"`, `"lindley"`, `"lfr"`, `"pxg"`, `"mtl"`, `"pfr"`.
`par`	numeric vector of true baseline parameters.
`x`	numeric matrix of covariates (default: no covariates).
`beta`	numeric vector of true regression coefficients.
`theta`	positive numeric; true frailty variance.
`cen_type`	character; censoring mechanism. One of `"none"`, `"right"`, `"left"`, `"interval"`, `"type1"`, `"type2"`, `"progressive"`, `"progressive_type1"`.
`cen_rate`	positive numeric; exponential censoring rate used for `"right"` and `"progressive"` censoring (default 0.2).
`left_threshold`	numeric; fixed left-censoring threshold (used when `cen_type = "left"`). Defaults to the 20th percentile of the true event times.
`int_width`	positive numeric; width of interval-censoring windows (used when `cen_type = "interval"`). Defaults to the 15th percentile of the true event times.

`cen_time`	positive numeric; fixed censoring time for Type-I censoring (`cen_type = "type1"`). Defaults to the 70th percentile of the true event times.
`r_failures`	positive integer; number of failures to observe before terminating the study for Type-II censoring (`cen_type = "type2"`). Defaults to $\lfloor 0.7n \rfloor$.
`prog_scheme`	integer vector; for `cen_type = "progressive"`, the number of units withdrawn at each of the first `length(prog_scheme)` failure times. For `cen_type = "progressive_type1"`, the number of units withdrawn at each time in `prog_times`. Defaults to a vector of 1s.
`prog_times`	positive numeric vector; pre-specified withdrawal times for Progressive Type-I censoring (`cen_type = "progressive_type1"`). Defaults to the 25th, 50th, and 75th percentiles of the true event times.

Value

A data frame with columns:

time event or censoring time.

time2 right end of interval (only for `cen_type = "interval"`, otherwise `NA`).

status 1 = exact event, `0` = right-censored (includes Type-I, Type-II, and progressive withdrawals), 2 = left-censored, 3 = interval-censored.

plus any covariate columns from `x`.

Examples

```
set.seed(42)
# Random right censoring
dat <- r_gamma_frailty(n = 100, baseline = "arvind", par = 0.5,
                        theta = 0.3, cen_type = "right", cen_rate = 0.2)
head(dat)

# Type-I censoring (fixed censoring time)
dat_t1 <- r_gamma_frailty(n = 100, baseline = "arvind", par = 0.5,
                           theta = 0.3, cen_type = "type1", cen_time = 2.0)
table(dat_t1$status)   # 0 = censored at 2.0, 1 = failed before 2.0

# Type-II censoring (fixed failure count)
dat_t2 <- r_gamma_frailty(n = 100, baseline = "arvind", par = 0.5,
                           theta = 0.3, cen_type = "type2", r_failures = 70L)
table(dat_t2$status)   # exactly 70 events (status=1)

# Progressive Type-I censoring
dat_pt1 <- r_gamma_frailty(n = 100, baseline = "arvind", par = 0.5,
                            theta = 0.3, cen_type = "progressive_type1",
                            prog_times = c(1, 2, 3), prog_scheme = c(5L, 5L, 5L))
table(dat_pt1$status)
```

`r_lfr` *Random samples from the Linear Failure Rate distribution*

Description

Generates `n` random observations from the Linear Failure Rate (LFR) distribution with parameters `a` and `b` via the inverse-CDF method. The survival function is $S(t) = \exp(-at - bt^2/2)$.

Usage

```
r_lfr(n, a, b)
```

Arguments

`n`	integer; number of observations.
`a`	non-negative numeric; intercept of the hazard.
`b`	non-negative numeric; slope of the hazard.

Value

Numeric vector of length `n`.

References

Bain, L. J. (1974). Analysis for the linear failure-rate life-testing distribution. Technometrics, 16(4), 551-559.

Examples

```
set.seed(1)
x <- r_lfr(100, a = 0.5, b = 0.2)
hist(x, main = "LFR samples")
```

`r_lindley` *Random samples from the Lindley distribution*

Description

Generates `n` random observations from the Lindley distribution with parameter `lambda` using a mixture representation: $X \sim p \cdot \text{Exp}(\lambda) + (1-p) \cdot \text{Gamma}(2, \lambda)$ where $p = \lambda/(1+\lambda)$.

Usage

```
r_lindley(n, lambda)
```

Arguments

`n`	integer; number of observations.
`lambda`	positive numeric; rate parameter.

Value

Numeric vector of length `n`.

References

Lindley, D. V. (1958). Fiducial distributions and Bayes' theorem. Journal of the Royal Statistical Society. Series B (Methodological), 102-107.

Examples

```
set.seed(1)
x <- r_lindley(100, lambda = 1.5)
hist(x, main = "Lindley samples")
```

`r_mtl` *Random samples from the Modified Topp-Leone distribution*

Description

Generates `n` random observations from the Modified Topp-Leone (MTL) distribution with parameter `alpha` using the inverse-CDF method via numerical root-finding.

Usage

```
r_mtl(n, alpha)
```

Arguments

`n`	integer; number of observations.
`alpha`	positive numeric; shape parameter.

Value

Numeric vector of length `n`.

References

Singh, B., Tyagi, S., Singh, R. P., & Tyagi, A. (2025). Modified Topp-Leone distribution: properties, classical and Bayesian estimation with application to COVID-19 and reliability data. Thailand Statistician, 23(1), 72-96.

Examples

```
set.seed(1)
x <- r_mtl(50, alpha = 2.0)
hist(x, main = "Modified Topp-Leone samples")
```

`r_pfr` *Random samples from the Power Failure Rate distribution*

Description

Generates `n` random observations from the Power Failure Rate (PFR) distribution with parameters `a` and `k` using the closed-form inverse-CDF: $t = \left(\frac{(k+1)(-\log U)}{a}\right)^{1/(k+1)}$.

Usage

```
r_pfr(n, a, k)
```

Arguments

`n`	integer; number of observations.
`a`	positive numeric; scale parameter.
`k`	non-negative numeric; shape parameter ($k \neq -1$).

Value

Numeric vector of length `n`.

References

Mugdadi, A. R. (2005). The least squares type estimation of the parameters in the power hazard function. Applied Mathematics and Computation, 169(2), 737-748.

Examples

```
set.seed(1)
x <- r_pfr(100, a = 0.5, k = 1.0)
hist(x, main = "Power Failure Rate samples")
```

`r_pxg` *Random samples from the Power Xgamma distribution*

Description

Generates `n` random observations from the Power Xgamma distribution with parameters `alpha` and `beta` using the inverse-CDF method via numerical root-finding.

Usage

```
r_pxg(n, alpha, beta)
```

Arguments

`n`	integer; number of observations.
`alpha`	positive numeric; first shape parameter.
`beta`	positive numeric; second shape parameter.

Value

Numeric vector of length `n`.

References

Tyagi, S., Kumar, S., Pandey, A., Saha, M., & Bagariya, H. Power xgamma distribution: Properties and its applications to cancer data. Int J Stat Reliab Eng. 2022; 9(1): 51-60.

Examples

```
set.seed(1)
x <- r_pxg(50, alpha = 1.0, beta = 0.8)
hist(x, main = "Power Xgamma samples")
```

`residuals_frailty` *Residuals for the Gamma Frailty Model*

Description

Computes a comprehensive set of residuals and goodness-of-fit metrics for a fitted gamma frailty model.

Usage

```
residuals_frailty(fit)
```

Arguments

`fit`	An object of class "`gamma_frailty_fit`".

Value

A named list containing:

cox_snell Cox-Snell residuals $r_i = -\log S(t_i)$.

martingale Martingale residuals $M_i = \delta_i - r_i$.

deviance Deviance residuals.

raw Raw residuals (model CDF minus empirical CDF on sorted data).

standardized Standardized raw residuals.

studentized Studentized (leave-one-out approximation) residuals.

fitted_S Fitted survival probabilities.

MSE, RMSE, MAE Mean squared / root mean squared / mean absolute error of raw residuals.

R_square, Adj_R_square R-squared of model CDF vs empirical CDF.

KS_stat, KS_pvalue Kolmogorov-Smirnov test statistic and p-value (Cox-Snell residuals vs Exp(1)).

Examples

```
set.seed(1)
dat <- r_gamma_frailty(80, "lindley", par = 1.5, theta = 0.5,
                       cen_type = "right", cen_rate = 0.3)
fit <- fit_gamma_frailty(dat$time, dat$status, baseline = "lindley")
res <- residuals_frailty(fit)
plot(res$fitted_S, res$deviance, xlab = "Fitted S(t)", ylab = "Deviance")
```

`risk_predict` *Risk Prediction for New Subjects*

Description

Computes risk scores $\eta_j = \exp(X_j\hat{\beta})$ and corresponding survival curves for new observations.

Usage

```
risk_predict(fit, newdata, times = NULL)
```

Arguments

`fit`	An object of class "`gamma_frailty_fit`".
`newdata`	numeric matrix or data frame of covariates for new subjects.
`times`	numeric vector of time points for survival prediction.

Value

A list with `risk` (risk scores) and `survival` (matrix of survival probabilities, subjects x times).

Examples

```
set.seed(4)
dat <- r_gamma_frailty(100, "arvind", par = 0.5, theta = 0.3,
                        x = matrix(rnorm(200), ncol = 2),
                        beta = c(0.4, -0.3),
                        cen_type = "right")
fit  <- fit_gamma_frailty(dat$time, dat$status, dat[, 4:5],
                           baseline = "arvind")
new  <- matrix(c(1, -0.5, 0.5, 1), nrow = 2)
risk_predict(fit, newdata = new, times = c(0.5, 1, 2))
```

`simulate_mle_performance` *Quick Simulation Performance Check*

Description

A lightweight single-scenario simulation to quickly verify MLE parameter recovery. Prints bias and MSE for each parameter.

Usage

```
simulate_mle_performance(
  n_sim = 100L,
  n = 100L,
  baseline = "arvind",
  par = 0.5,
  beta = numeric(0),
  theta = 0.5,
  cen_rate = 0.1
)
```

Arguments

`n_sim`	integer; number of replicates (default 100).
`n`	integer; sample size (default 100).
`baseline`	character; baseline distribution.
`par`	numeric vector of true baseline parameters.
`beta`	numeric vector of true regression coefficients.
`theta`	positive numeric; true frailty variance.
`cen_rate`	numeric; exponential censoring rate.

Value

A data frame (invisibly) with columns True, Mean, Bias, MSE.

Examples

```
set.seed(99)
simulate_mle_performance(n_sim = 50, n = 80, baseline = "pfr",
                         par = c(0.5, 1), theta = 0.4)
```

`simulation_study` *Monte Carlo Simulation Study for Gamma Frailty Models*

Description

Evaluates MLE performance (bias, MSE, coverage, and convergence rate) for a specified gamma frailty model across multiple sample sizes and censoring settings.

Usage

```
simulation_study(
  n_sim = 500L,
  n_vec = c(50L, 100L, 200L),
  baseline = "arvind",
  par = 0.5,
  beta = numeric(0),
  theta = 0.5,
  cen_type = "right",
  cen_rate = 0.2,
  conf_level = 0.95,
  verbose = TRUE
)
```

Arguments

`n_sim`	integer; number of Monte Carlo replicates per scenario (default 500).
`n_vec`	integer vector; sample sizes to evaluate (default `c(50, 100, 200)`).
`baseline`	character; baseline distribution.
`par`	numeric vector of true baseline parameters.
`beta`	numeric vector of true regression coefficients (or `numeric(0)` for no-covariate models).
`theta`	positive numeric; true frailty variance.
`cen_type`	character; censoring type passed to `r_gamma_frailty`.
`cen_rate`	positive numeric; censoring rate (for right/progressive).
`conf_level`	numeric; nominal coverage probability for CIs (default 0.95).
`verbose`	logical; print progress (default `TRUE`).

Value

A data frame summarising, for each sample size, the mean estimate, bias, MSE, and 95% CI coverage for each parameter.

Examples

```
set.seed(42)
sim_res <- simulation_study(
  n_sim = 100, n_vec = c(50, 100),
  baseline = "arvind", par = 0.5,
  beta = 0.5, theta = 0.3,
  cen_type = "right", cen_rate = 0.2
)
print(sim_res)
```

`survival_at` *Survival Probability at Specified Time Points*

Description

Returns a tidy data frame of predicted survival probabilities at user-supplied time points, one row per (subject, time) combination.

Usage

```
survival_at(fit, times, newdata = NULL)
```

Arguments

`fit`	An object of class "`gamma_frailty_fit`".
`times`	numeric vector of time points.
`newdata`	optional covariate matrix / data frame for new subjects.

Value

A data frame with columns `subject`, `time`, `survival`.

Examples

```
set.seed(1)
dat <- r_gamma_frailty(60, "arvind", par = 0.5, theta = 0.3,
                        cen_type = "right")
fit <- fit_gamma_frailty(dat$time, dat$status, baseline = "arvind")
survival_at(fit, times = c(0.5, 1, 2))
```

Index