Package 'AlphaPowerHazard'

July 29, 2026

Type Package

Title Alpha-Power Hazard Regression Models for Survival Data

Version 0.1.0

Description Implements the alpha-power hazard model and regression frameworks for survival data based on the flexible hazard rate function h(x; alpha, beta) = alpha^x + x^(beta-1) (Pal et al., 2026 <doi:10.1007/s41096-026-00297-5>). Provides standard distribution functions (d, p, q, r, h, H, s) and distributional properties including raw/central moments, variance, skewness, kurtosis, quantile statistics (Bowley's skewness, Moors's kurtosis), Lambert W hazard rate function minimum (Corless et al., 1996), order statistics, and stochastic ordering (Shaked & Shanthikumar, 1994). Computes five classical estimation methods for baseline parameters: Maximum Likelihood Estimation (Casella & Berger, 2002), Least Squares Estimation (Swain et al., 1988), Weighted Least Squares Estimation (Styan, 1973), Maximum Product of Spacings Estimation (Cheng & Amin, 1983 <doi:10.1111/j.2517-6161.1983.tb01241.x>), and Cramer-von Mises Estimation (Macdonald, 1971). Supports four hazard regression models (M1-M4) within proportional hazards and parametric frameworks across uncensored data, right censoring, left censoring, interval censoring, and progressive Type-I and Type-II censoring schemes (Lee & Wang, 2003; Lawless, 2011; Balakrishnan & Aggarwala, 2000). Includes comprehensive model diagnostics, Cox-Snell, martingale, deviance, standardized, and studentized residuals, leverage, Cook's distance, DFFITS, DFBETAS, model comparisons (AIC, BIC, WAIC), k-fold cross-validation, prediction suites, random data generators, and an eight-panel diagnostic visualization suite.

License GPL-3

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>),
Arvind Pandey [aut],
Bhupendra Singh [aut],
Vrijesh Tripathi [aut]

Maintainer Shikhar Tyagi <shikhar1093tyagi@gmail.com>

Repository CRAN

Date/Publication 2026-07-29 16:50:09 UTC

Contents

`alpha_power_distribution`

Alpha-Power Hazard Distribution Functions and Statistical Properties

---

Description

Probability density function (PDF), cumulative distribution function (CDF), quantile function, random variates generation, hazard function, cumulative hazard function, survival function, raw and central moments, quantile statistics (Bowley's skewness and Moors's kurtosis), hazard rate minimum location, order statistics, and stochastic ordering for the Alpha-Power Hazard distribution.

Usage

```
halpha_power(x, alpha, beta)

Halpha_power(x, alpha, beta)

salpha_power(x, alpha, beta)

dalpha_power(x, alpha, beta, log = FALSE)

palpha_power(q, alpha, beta, lower.tail = TRUE, log.p = FALSE)

qalpha_power(p, alpha, beta, lower.tail = TRUE, log.p = FALSE)

ralpha_power(n, alpha, beta)

r_alpha_power(n, alpha, beta)
```

Arguments

`x`, `q`	numeric vector of quantiles or values.
`alpha`	positive numeric (> 1); alpha shape parameter.
`beta`	positive numeric (> 0); beta shape parameter.
`log`, `log.p`	logical; if `TRUE`, probabilities/densities are returned on the log scale.
`lower.tail`	logical; if `TRUE` (default), probabilities are $P[X \leq x]$, otherwise $P[X > x]$.
`p`	numeric vector of probabilities.
`n`	integer; number of random variates to generate.

Details

The baseline hazard rate function is given by:

$$h(x;\alpha,\beta) = \alpha^x + x^{\beta-1}, \quad \alpha > 1, \beta > 0, x > 0.$$

The cumulative hazard function is:

$$H(x;\alpha,\beta) = \frac{\alpha^x - 1}{\log \alpha} + \frac{x^\beta}{\beta}.$$

The survival function is:

$$S(x;\alpha,\beta) = \exp\left(-H(x;\alpha,\beta)\right).$$

The probability density function is:

$$f(x;\alpha,\beta) = (\alpha^x + x^{\beta-1}) \exp\left(-\left[\frac{\alpha^x - 1}{\log \alpha} + \frac{x^\beta}{\beta}\right]\right).$$

Value

`dalpha_power` returns density values. `palpha_power` returns cumulative probabilities. `qalpha_power` returns quantiles. `ralpha_power` and `r_alpha_power` return random variates. `halpha_power` returns hazard values. `Halpha_power` returns cumulative hazard values. `salpha_power` returns survival probabilities.

References

Pal, S., Tyagi, V., Chesneau, C., & Sharma, V. K. (2026). A New Flexible Hazard Regression Model for Survival Data Analysis Alternative to Cox and AFT Regression Models. Journal of the Indian Society for Probability and Statistics. doi:10.1007/s41096026002975

Examples

```
dalpha_power(c(0.5, 1, 1.5), alpha = 1.5, beta = 0.8)
palpha_power(c(0.5, 1, 1.5), alpha = 1.5, beta = 0.8)
qalpha_power(c(0.25, 0.5, 0.75), alpha = 1.5, beta = 0.8)
set.seed(123)
ralpha_power(5, alpha = 1.5, beta = 0.8)
```

---

alpha_power_functions *Alpha-Power Hazard Model Functions (Models M1-M4)*

Description

Computes the survival function $S(t)$, density $f(t)$, hazard $h(t)$, and cumulative hazard $H(t)$ for regression models M1, M2, M3, and M4 with optional gamma frailty.

Usage

```
alpha_power_functions(
  t,
  eta = rep(0, length(t)),
  theta_frailty = NULL,
  par = c(1.5, 1.1),
  model = "M1"
)
```

Arguments

t	positive numeric vector of time points.
eta	numeric vector of linear predictors $\eta_j = Z_j^\top \theta$.
theta_frailty	positive numeric; frailty variance parameter (if NULL or 0, no frailty).
par	numeric vector of baseline parameters c(alpha, beta).
model	character; one of "M1", "M2", "M3", "M4".

Value

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

---

alpha_power_hazard *Alpha-Power Hazard Regression Model (Formula Interface)*

Description

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

Usage

```
alpha_power_hazard(
  formula,
  data,
  model = c("M1", "M2", "M3", "M4"),
  frailty = FALSE,
  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.
model	character; one of "M1", "M2", "M3", "M4". Default is "M1".
frailty	logical; whether to include frailty. Default is FALSE.
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("alpha_power", "alpha_power_fit") with an additional call and formula component.

Examples

```
set.seed(1)
dat <- data.frame(
  time   = c(1.2, 0.8, 2.3, 1.5, 3.1),
  status = c(1, 1, 0, 1, 0),
  X1     = c(0.5, -0.2, 0.8, 0.1, -0.5)
)
fit <- alpha_power_hazard(survival::Surv(time, status) ~ X1, data = dat, model = "M1")
summary(fit)
```

---

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 alpha-power hazard distribution.

Usage

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

Arguments

t	positive numeric vector of time points.
par	numeric vector of baseline parameters c(alpha, beta).

Details

Alpha-Power Hazard (par = c(alpha, beta)): $h_0(t) = \alpha^t + t^{\beta-1}$, $H_0(t) = (\alpha^t - 1)/\log(\alpha) + t^\beta/\beta$.

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), par = c(1.5, 0.8))
plot(seq(0.1, 2, by = 0.1), bh$h0, type = "l", main = "Alpha-Power hazard")
```

---

bootstrap_waic *Bootstrap WAIC for the Alpha-Power Hazard Model*

Description

Approximates the Widely Applicable Information Criterion (WAIC) via bootstrap re-sampling of the log-likelihood.

Usage

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

Arguments

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

Value

A named list with components WAIC, lppd, p_waic, B_effective.

---

compare_models *Model Comparison Table (AIC, BIC, WAIC)*

Description

Compares multiple fitted alpha-power models in a single table.

Usage

```
compare_models(..., B_waic = 100)
```

Arguments

...	fitted objects of class "alpha_power_fit".
B_waic	integer; bootstrap replicates for WAIC calculation.

Value

A data frame summarizing model comparisons.

---

cv_alpha_power *K-Fold Cross-Validation for the Alpha-Power Hazard Model*

Description

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

Usage

```
cv_alpha_power(
  time,
  status,
  x = matrix(nrow = length(time), ncol = 0),
  model = "M1",
  frailty = FALSE,
  K = 5L
)
```

Arguments

time	positive numeric vector of event/censoring times.
status	integer vector of censoring indicators.
x	numeric matrix of covariates.
model	character; model specification "M1", "M2", "M3", or "M4".
frailty	logical; whether to include frailty.
K	integer; number of folds (default 5).

Value

A list with cross-validation performance metrics.

diagnostics_table *Summary Table of Diagnostics*

Description

Creates a summary table of model diagnostic statistics.

Usage

```
diagnostics_table(fit)
```

Arguments

fit	An object of class "alpha_power_fit".

Value

A data frame of diagnostic metrics.

fit_alpha_power *Fit an Alpha-Power Hazard Regression Model (Models M1-M4)*

Description

Fits an alpha-power hazard regression model via maximum likelihood estimation (MLE). Supports models M1, M2, M3, and M4 from Pal et al. (2026). Handles uncensored and right-, left-, interval-, and progressive-censored data.

Usage

```
fit_alpha_power(
  time,
  status,
  x = matrix(nrow = length(time), ncol = 0),
  model = c("M1", "M2", "M3", "M4"),
  frailty = FALSE,
  time2 = NULL,
  prog_cen = NULL,
  init = NULL
)
```

Arguments

time	positive numeric vector of event or censoring times.
status	integer vector indicating observation type: 1 = exact event, 0 = right-censored, 2 = left-censored, 3 = interval-censored.
x	numeric matrix or data frame of covariates.
model	character; one of "M1", "M2", "M3", "M4". Default is "M1".
frailty	logical; whether to include a gamma frailty parameter θ. Default is FALSE.
time2	numeric vector; upper bound for interval-censored observations.
prog_cen	integer vector; progressive censoring scheme.
init	numeric vector of initial parameter values on estimation scale.

Value

An object of class "alpha_power_fit" containing parameter estimates, standard errors, log-likelihood, AIC, BIC, and model diagnostics.

---

fit_alpha_power_base *Classical Baseline Parameter Estimation Methods*

Description

Computes point estimates and standard errors for the parameters α and β of the Alpha-Power Hazard distribution using five classical estimation methods: Maximum Likelihood Estimation (MLE), Least Squares Estimation (LSE), Weighted Least Squares Estimation (WLSE), Maximum Product of Spacings Estimation (MPSE), and Cramér-von Mises Estimation (CME).

Usage

```
fit_alpha_power_base(
  x,
  method = c("mle", "lse", "wlse", "mpse", "cme"),
  init = NULL
)
```

Arguments

x	positive numeric vector of failure time observations.
method	character; estimation method. One of "mle", "lse", "wlse", "mpse", "cme".
init	numeric vector of initial parameters c(alpha, beta). If NULL, heuristic initial values are computed.

Value

An object of class "alpha_power_base_fit" containing:

par	Estimated parameters c(alpha, beta).
method	Estimation method name.
value	Optimized objective function value.
convergence	Integer code (0 = success).
hessian	Hessian matrix at minimum/maximum.
se	Standard errors of parameters (if available).

References

Pal, S., Tyagi, V., Chesneau, C., & Sharma, V. K. (2026). A New Flexible Hazard Regression Model for Survival Data Analysis Alternative to Cox and AFT Regression Models. Journal of the Indian Society for Probability and Statistics. doi:10.1007/s41096026002975

Examples

```
set.seed(123)
dat <- ralpha_power(50, alpha = 1.5, beta = 0.8)
fit_mle  <- fit_alpha_power_base(dat, method = "mle")
fit_lse  <- fit_alpha_power_base(dat, method = "lse")
fit_wlse <- fit_alpha_power_base(dat, method = "wlse")
fit_mpse <- fit_alpha_power_base(dat, method = "mpse")
fit_cme  <- fit_alpha_power_base(dat, method = "cme")
fit_mle$par
```

---

hrf_min_alpha_power *Hazard Rate Function Minimum Location and Value*

Description

Computes the unique minimum point $x_{\min}$ and minimum hazard rate $h(x_{\min})$ for the Alpha-Power Hazard distribution when $0 < \beta < 1$.

Usage

```
hrf_min_alpha_power(alpha, beta)
```

Arguments

alpha	positive numeric (> 1); alpha parameter.
beta	positive numeric in (0, 1); beta shape parameter.

Value

A list containing xmin and hmin.

influence_alpha_power *Influence Diagnostics for the Alpha-Power Hazard Model*

Description

Computes leverage values, Cook's distance, DFFITS, and DFBETAS.

Usage

```
influence_alpha_power(fit)
```

Arguments

fit	An object of class "alpha_power_fit".

Value

A named list containing influence metrics.

loglik_alpha_power *Log-Likelihood for the Alpha-Power Hazard Regression Models*

Description

Internal function computing total log-likelihood for uncensored and censored (right, left, interval, progressive) data under models M1-M4.

Usage

```
loglik_alpha_power(
  par_all,
  time,
  status,
  x,
  model = "M1",
  has_frailty = FALSE,
  time2 = NULL,
  prog_cen = NULL
)
```

Arguments

par_all	numeric vector of parameters on estimation scale.
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.
model	character; one of "M1", "M2", "M3", "M4".
has_frailty	logical; whether frailty parameter is included as last element.
time2	numeric vector; upper bound for interval censoring.
prog_cen	integer vector; progressive censoring scheme.

Value

Scalar log-likelihood value.

---

moments_alpha_power *Statistical Moments of the Alpha-Power Hazard Distribution*

Description

Computes the raw moments μ_k, central moments, variance, coefficient of skewness, and coefficient of kurtosis for the Alpha-Power Hazard distribution.

Usage

```
moments_alpha_power(k = 4, alpha = 1.5, beta = 1.1)
```

Arguments

k	integer >= 1; maximum moment order to compute.
alpha	positive numeric (> 1); alpha parameter.
beta	positive numeric (> 0); beta parameter.

Value

A list with components:

raw_moments	Numeric vector of raw moments $\mu_1, \ldots, \mu_k$.
mean	First raw moment μ_1.
variance	Variance $\sigma^2 = \mu_2 - \mu_1^2$.
skewness	Coefficient of skewness $S = \mu_3^*/\sigma^3$.
kurtosis	Coefficient of kurtosis $K = \mu_4^*/\sigma^4$.
central_moments	Numeric vector of central moments $\mu_1^*, \ldots, \mu_k^*$.

`order_stats_alpha_power` *Order Statistics of the Alpha-Power Hazard Distribution*

Description

Computes the PDF or CDF of the r-th order statistic from an independent sample of size n drawn from the Alpha-Power Hazard distribution.

Usage

```
order_stats_alpha_power(x, r, n, alpha, beta, type = c("pdf", "cdf"))
```

Arguments

`x`	numeric vector of values.
`r`	integer; rank of order statistic ($1 \leq r \leq n$).
`n`	integer; sample size.
`alpha`	positive numeric (> 1); alpha parameter.
`beta`	positive numeric (> 0); beta parameter.
`type`	character; either `"pdf"` or `"cdf"`.

Value

Numeric vector of PDF or CDF values of $X_{(r)}$.

`plot_all` *Plot All Diagnostics*

Description

Generates all 8 diagnostic plots for a fitted alpha-power hazard 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 "alpha_power_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_alpha_power_reg(80, par = c(1.5, 0.8), theta = 0.3,
                     cen_type = "right", cen_rate = 0.2)
fit <- fit_alpha_power(dat$time, dat$status)
# 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 the alpha-power hazard distribution over a time grid.

Usage

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

Arguments

par	numeric vector of baseline parameters c(alpha, beta).
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(par = c(1.5, 0.8), 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 `"alpha_power_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 `"alpha_power_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 `"alpha_power_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 `"alpha_power_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 "alpha_power_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 "alpha_power_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 "alpha_power_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 "alpha_power_fit".
...	additional graphical parameters.

Value

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

predict_alpha_power *Predictions from an Alpha-Power Hazard 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 alpha-power hazard model.

Usage

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

survival_at(fit, time_points, newdata = NULL)

risk_predict(fit, newdata = NULL)

forecast_alpha_power(fit, horizon, newdata = NULL, n_grid = 100)
```

Arguments

fit	An object of class "alpha_power_fit".
newdata	numeric matrix or data frame of covariates for prediction.
newtime	numeric vector of time points for prediction.
type	character; prediction type: "survival", "hazard", "median", "expected", "risk", "marginal", "forecast".
window	numeric vector of length 2 giving time interval for type = "expected".
time_points	numeric vector of time points for survival_at.
horizon	positive numeric; time horizon for forecast_alpha_power.
n_grid	integer; number of grid points for forecasting.

Value

Prediction outputs depending on type or helper function invoked.

Examples

```
set.seed(1)
dat <- r_alpha_power_reg(50, par = c(1.5, 0.8), cen_type = "right")
fit <- fit_alpha_power(dat$time, dat$status)
pred_s <- predict_alpha_power(fit, type = "survival")
```

quantile_stats_alpha_power *Quantile Statistics of the Alpha-Power Hazard Distribution*

Description

Computes key quantiles (0.125, 0.25, 0.375, 0.5, 0.625, 0.75, 0.875), Bowley's coefficient of skewness, and Moors's coefficient of kurtosis.

Usage

```
quantile_stats_alpha_power(alpha = 1.5, beta = 1.1)
```

Arguments

alpha	positive numeric (> 1); alpha parameter.
beta	positive numeric (> 0); beta parameter.

Value

A list with quantiles, Bowley's skewness (bowley_skewness), and Moors's kurtosis (moors_kurtosis).

residuals_alpha_power *Residuals for the Alpha-Power Hazard Model*

Description

Computes a comprehensive set of residuals and goodness-of-fit metrics for a fitted alpha-power hazard model.

Usage

```
residuals_alpha_power(fit)
```

Arguments

fit	An object of class "alpha_power_fit".

Value

A named list containing residuals, leverage, and fit metrics.

`r_alpha_power_reg` *Random Data Generation for Alpha-Power Hazard Regression Models*

Description

Generates synthetic survival data under the Alpha-Power Hazard regression models (M1-M4) with optional gamma frailty and censoring schemes (right, left, interval, progressive Type-I & Type-II).

Usage

```
r_alpha_power_reg(
  n,
  par = c(1.5, 1.1),
  theta = NULL,
  x = matrix(nrow = n, ncol = 0),
  beta = numeric(0),
  model = "M1",
  cen_type = c("none", "right", "left", "interval"),
  cen_rate = 0.1,
  time2 = NULL
)
```

Arguments

`n`	integer; sample size.
`par`	numeric vector of baseline parameters `c(alpha, beta)`.
`theta`	positive numeric; frailty variance parameter (optional).
`x`	numeric matrix of covariates.
`beta`	numeric vector of regression coefficients.
`model`	character; one of `"M1"`, `"M2"`, `"M3"`, `"M4"`. Default is `"M1"`.
`cen_type`	character; censoring type: `"none"`, `"right"`, `"left"`, `"interval"`.
`cen_rate`	positive numeric; censoring rate parameter for exponential censoring distribution.
`time2`	numeric vector; upper bound for interval censoring.

Value

A data frame with columns `time`, `status`, covariates, and optionally `time2`.

Examples

```
set.seed(123)
dat <- r_alpha_power_reg(50, par = c(1.5, 0.8), x = matrix(rnorm(50), ncol = 1),
                         beta = 0.5, cen_type = "right", cen_rate = 0.2)
head(dat)
```

simulate_mle_performance *Quick Simulation Performance Check*

Description

A lightweight single-scenario simulation to quickly verify MLE parameter recovery.

Usage

```
simulate_mle_performance(
  n_sim = 50L,
  n = 50L,
  par = c(1.5, 1.1),
  beta = numeric(0),
  model = "M1",
  frailty = FALSE,
  theta = 0.5,
  cen_rate = 0.1
)
```

Arguments

n_sim	integer; number of replicates.
n	integer; sample size.
par	numeric vector of true baseline parameters c(alpha, beta).
beta	numeric vector of true regression coefficients.
model	character; model specification.
frailty	logical; whether frailty is included.
theta	positive numeric; true frailty variance.
cen_rate	numeric; exponential censoring rate.

Value

A data frame of simulation summary metrics.

simulation_study *Monte Carlo Simulation Study for Alpha-Power Hazard Models*

Description

Evaluates MLE performance (bias, MSE, coverage, and convergence rate) for a specified alpha-power hazard model across multiple sample sizes and censoring settings.

Usage

```
simulation_study(
  n_sim = 100L,
  n_vec = c(50L, 100L),
  par = c(1.5, 1.1),
  beta = numeric(0),
  model = "M1",
  frailty = FALSE,
  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.
n_vec	integer vector; sample sizes to evaluate.
par	numeric vector of true baseline parameters c(alpha, beta).
beta	numeric vector of true regression coefficients.
model	character; model specification "M1", "M2", "M3", or "M4".
frailty	logical; whether frailty parameter is included.
theta	positive numeric; true frailty variance (if frailty = TRUE).
cen_type	character; censoring type: "none", "right", "left", "interval".
cen_rate	positive numeric; censoring rate.
conf_level	numeric; nominal coverage probability.
verbose	logical; print progress.

Value

A data frame summarizing simulation metrics.

`stochastic_ordering_alpha_power` *Stochastic Ordering Comparison*

Description

Verifies stochastic ordering $F(x; \alpha_1, \beta_1) \leq F(x; \alpha_2, \beta_2)$ for $\alpha_1 \leq \alpha_2$ and $\beta_1 \leq \beta_2$.

Usage

```
stochastic_ordering_alpha_power(
  alpha1,
  beta1,
  alpha2,
  beta2,
  x = seq(0.1, 5, by = 0.1)
)
```

Arguments

`alpha1`, `alpha2`	positive numeric (> 1); alpha parameters.
`beta1`, `beta2`	positive numeric (> 0); beta parameters.
`x`	numeric vector of positive evaluation points.

Value

A list with `F1`, `F2`, and logical `is_ordered`.

Index