Package: BetaDanish (via r-universe)

June 4, 2026

Type Package

Title The Beta-Danish Distribution for Lifetime Data Analysis

Version 0.2.0

Description Implements the four-parameter Beta-Danish distribution and its three-parameter submodel for survival and reliability analysis, based on Ahmad and Danish (2025) <doi:10.2478/jamsi-2025-0010>. Provides functions for density, distribution, quantile, hazard, and random generation. Includes maximum likelihood estimation for complete and right-censored data, goodness-of-fit assessment, comparison with standard lifetime distributions, and publication-quality visualizations. Advanced modules support Accelerated Failure Time (AFT) regression, mixture and promotion-time cure models, and competing risks analysis.

URL `https://bilal-aiou.github.io/BetaDanish/`, `https://github.com/bilal-aiou/BetaDanish`

BugReports `https://github.com/bilal-aiou/BetaDanish/issues`

License GPL-3

Encoding UTF-8

LazyData true

Imports maxLik, survival, stats, graphics, utils, tools

Suggests readxl, flexsurv, testthat (>= 3.0.0), knitr, rmarkdown, cmprsk, coda, MCMCpack

VignetteBuilder knitr

Config/testthat/edition 3

Depends R (>= 3.5)

Config/roxygen2/version 8.0.0

Language en-US

Repository https://bilal-aiou.r-universe.dev

Date/Publication 2026-06-04 06:29:00 UTC

RemoteUrl https://github.com/bilal-aiou/betadanish

RemoteRef HEAD

RemoteSha 0ec95a909b6ebd282cd66e85aaa4dbf0161ef3d2

Contents

`aarset` *Aarset Device Failure Times*

Description

Times to failure of 50 devices, exhibiting a classic bathtub-shaped hazard rate. This is a standard benchmark dataset in reliability engineering.

Usage

```
aarset
```

Format

A data frame with 50 rows and 2 columns:

time Failure time

status Event indicator (1 = event occurred)

Source

Aarset, M. V. (1987). How to Identify a Bathtub Hazard Rate. IEEE Transactions on Reliability, R-36(1), 106-108.

Examples

```
data(aarset)

fit <- fit_betadanish(survival::Surv(time, status) ~ 1, data = aarset)
plot(fit, type = "hazard")
```

`analyze_betadanish` *Comprehensive Beta-Danish Analysis Pipeline*

Description

Runs a complete end-to-end analysis: reads data, fits the 4-parameter and 3-parameter models, compares them, benchmarks against standard distributions, and generates diagnostic plots.

Usage

```
analyze_betadanish(file, time_col, status_col = NULL)
```

Arguments

`file`	Path to the CSV or Excel file containing the data.
`time_col`	Name of the time column.
`status_col`	Name of the status column (optional).

Value

Invisibly returns a list containing the fitted full model and submodel objects. The function is mainly called for its side effects of printing an analysis report and producing diagnostic plots.

`bayes_betadanish` *Bayesian Estimation for the Beta-Danish Distribution*

Description

Samples from the posterior of the Beta-Danish or Exponentiated Danish parameters using a random-walk Metropolis sampler with vague $\Gamma(0.01, 0.01)$ priors on the positive parameters.

Usage

```
bayes_betadanish(
  time,
  status = NULL,
  submodel = TRUE,
  burnin = 5000,
  mcmc = 15000,
  tune = 0.5,
  theta_init = NULL,
  seed = NULL,
  verbose = 0
)
```

Arguments

`time`	Numeric vector of observed times.
`status`	Numeric vector of event indicators (1 = event, 0 = right-censored).
`submodel`	Logical; `TRUE` for the 3-parameter ED submodel, `FALSE` for the 4-parameter full model.
`burnin`	Burn-in iterations.
`mcmc`	Post-burnin iterations.
`tune`	Random-walk tuning parameter.
`theta_init`	Optional starting values on the log scale.
`seed`	Optional integer seed.
`verbose`	Integer; passed to MCMCmetrop1R (0 = silent).

Details

Requires **MCMCpack** and **coda** (Suggests).

Value

An object of class `"bd_bayes"` with components `draws` (mcmc object), `summary`, `HPD`, `submodel`, `call`.

Examples

```
## Not run:
set.seed(1)
dat <- rbetadanish(100, a = 1.5, b = 2, c = 3, k = 0.5)
fit <- bayes_betadanish(time = dat, submodel = TRUE,
                        burnin = 500, mcmc = 1500)
fit$summary

## End(Not run)
```

`bd_entropy_shannon` *Shannon Entropy of the Beta-Danish Distribution*

Description

Computes the differential Shannon entropy $H(f) = -\int_0^\infty f(t)\log f(t)\,dt$ for the four-parameter Beta-Danish distribution by adaptive Gauss-Kronrod quadrature on the log-pdf.

Usage

```
bd_entropy_shannon(a, b, c, k, subdivisions = 2000, rel.tol = 1e-08)
```

Arguments

`a, b, c, k`	Positive parameters of the Beta-Danish distribution.
`subdivisions, rel.tol`	Passed to `stats::integrate`.

Value

Scalar Shannon entropy (in nats); `NA_real_` on integration failure.

Examples

```
bd_entropy_shannon(a = 1.5, b = 2.5, c = 2, k = 1)
```

`bd_order_stat_pdf` *Density of the r-th Order Statistic*

Description

Evaluates the probability density function of the r-th order statistic from a sample of size n drawn from the Beta-Danish distribution.

Usage

```
bd_order_stat_pdf(x, r, n, a, b, c, k, log = FALSE)
```

Arguments

`x`	Numeric vector of time points.
`r`	Integer order (1 = minimum, n = maximum).
`n`	Integer sample size.
`a, b, c, k`	Positive parameters of the Beta-Danish distribution.
`log`	Logical; if `TRUE` return the log-density.

Value

Numeric vector (or its log).

Examples

```
tgrid <- seq(0.01, 5, length.out = 50)
bd_order_stat_pdf(tgrid, r = 5, n = 20,
                  a = 1.5, b = 2.5, c = 2, k = 1)
```

`BetaDanish` *The Beta-Danish Distribution*

Description

Density, distribution function, quantile function, hazard function, and random generation for the four-parameter Beta-Danish distribution.

Usage

```
dbetadanish(x, a, b, c, k, log = FALSE)

pbetadanish(q, a, b, c, k, lower.tail = TRUE, log.p = FALSE)

qbetadanish(p, a, b, c, k, lower.tail = TRUE, log.p = FALSE)

rbetadanish(n, a, b, c, k)

hbetadanish(x, a, b, c, k, log = FALSE)
```

Arguments

`x, q`	Vector of quantiles (time points).
`a`	Shape parameter (beta generator). Set ‘a = 1‘ for the 3-parameter submodel.
`b`	Shape parameter (beta generator / tail weight).
`c`	Shape parameter (baseline shape).
`k`	Scale parameter (baseline scale).
`log, log.p`	Logical; if TRUE, probabilities/densities are given as log.
`lower.tail`	Logical; if TRUE (default), probabilities are P[X <= x], otherwise P[X > x].
`p`	Vector of probabilities.
`n`	Number of observations to generate.

Details

The Beta-Danish distribution is a highly flexible lifetime distribution capable of modeling decreasing, increasing, unimodal, and bathtub-shaped hazard rates.

Value

‘dbetadanish‘ gives the density, ‘pbetadanish‘ gives the distribution function, ‘qbetadanish‘ gives the quantile function, ‘hbetadanish‘ gives the hazard function, and ‘rbetadanish‘ generates random deviates.

References

Ahmad, B., & Danish, M. Y. (2026). Development and Characterization of a Flexible Three-Parameter Lifetime Distribution.

Examples

```
# Density
dbetadanish(x = 2, a = 1.5, b = 2, c = 3, k = 0.5)

# CDF
pbetadanish(q = 2, a = 1.5, b = 2, c = 3, k = 0.5)
```

```
# Hazard
hbetadanish(x = 2, a = 1.5, b = 2, c = 3, k = 0.5)

# Random generation
rbetadanish(n = 10, a = 1.5, b = 2, c = 3, k = 0.5)
```

`brain_cancer` *Brain Cancer Survival Data*

Description

A comprehensive dataset of 500 brain cancer patients, including survival times, censoring status, and multiple clinical covariates. This dataset was used to demonstrate Accelerated Failure Time (AFT) regression and Cure-Rate models using the Beta-Danish distribution.

Usage

```
brain_cancer
```

Format

A data frame with 500 rows and 16 columns:

ID	Patient identifier
Gender	Patient gender (1 = Male, 0 = Female)
Age	Age group (1 = Young, 2 = Middle, 3 = Old)
Area	Geographic area (1 = Urban, 0 = Rural)
FH	Family history of cancer (1 = Yes, 0 = No)
CMH	Comorbid history (1 = Yes, 0 = No)
Grade	Tumor grade (1 = I/II, 2 = III, 3 = IV)
Surgery	Surgical intervention (1 = Yes, 0 = No)
Radiotherapy	Radiotherapy treatment (1 = Yes, 0 = No)
Chemotherapy	Chemotherapy treatment (1 = Yes, 0 = No)
Treatment	Treatment type
Morphology	Tumor morphology
Survstatus	Survival status (1 = Event/Death, 0 = Censored)
Survtime	Survival time in months
Types	Tumor types classification
Morphology1	Alternative morphology classification

Source

Atomic Energy Cancer Hospital (NORI), Islamabad, Pakistan.

Examples

```
data(brain_cancer)

# Fit an AFT model using the brain cancer data
fit <- fit_bd_aft(survival::Surv(Survtime, Survstatus) ~ Age + Grade + Surgery,
                  data = brain_cancer, n_starts = 2)
summary(fit)
```

`carbon_fibres` *Breaking Stress of Carbon Fibres*

Description

Breaking stress (in Gba) of 100 carbon fibre specimens. This dataset exhibits a unimodal (increasing-then-decreasing) hazard pattern that classical distributions like the Weibull cannot adequately capture.

Usage

```
carbon_fibres
```

Format

A data frame with 100 rows and 2 columns:

time Breaking stress in Gba

status Event indicator (1 = event occurred)

Source

Nichols, M. D., & Padgett, W. J. (2006). A bootstrap control chart for Weibull percentiles. Quality and Reliability Engineering International, 22(2), 141-151.

Examples

```
data(carbon_fibres)

fit <- fit_betadanish(survival::Surv(time, status) ~ 1, data = carbon_fibres)
```

`cif_betadanish` *Compute Cumulative Incidence Function (CIF)*

Description

Computes the CIF for a specific cause from a fitted Beta-Danish competing risks model using numerical integration.

Usage

```
cif_betadanish(fit, tvec, cause_idx)
```

Arguments

`fit`	An object of class ‘bd_competing‘.
`tvec`	Numeric vector of times at which to evaluate the CIF.
`cause_idx`	Integer; the specific cause to evaluate (must match one of the causes in the fitted model).

Value

A numeric vector of CIF probabilities corresponding to ‘tvec‘.

`cif_compare` *Compare Model-Based CIF to the Aalen-Johansen Estimator*

Description

Computes the nonparametric Aalen-Johansen CIF (via **cmprsk**) for each competing-risks cause, overlays it on the fitted Beta-Danish CIF, and returns the Aalen-Johansen times/estimates, the fitted CIF values on a common grid, and Gray’s CIF-equality test where applicable.

Usage

```
cif_compare(fit, tmax = NULL, n_grid = 160, plot = TRUE)
```

Arguments

`fit`	A fitted object of class `"bd_competing"`.
`tmax`	Optional upper time for evaluation; default the 95th percentile of observed times.
`n_grid`	Number of time points on the evaluation grid (default 160).
`plot`	Logical; if `TRUE` (default) draws a panel of overlays.

Details

Requires **cmprsk** (Suggests).

Value

A list with elements `tgrid`, `cif_fit` (data frame long format), `cif_aj` (data frame long format) and optionally `gray_test`.

Examples

```
## Not run:
set.seed(1)
T1 <- rbetadanish(200, 1.2, 1.5, 1.0, 0.4)
T2 <- rbetadanish(200, 1.0, 2.0, 1.0, 0.2)
C  <- stats::rexp(200, 0.05)
time  <- pmin(T1, T2, C)
cause <- ifelse(time == C, 0L, ifelse(T1 <= T2, 1L, 2L))
fit <- fit_bd_competing(time = time, cause = cause)
cif_compare(fit)   # requires cmprsk to be installed

## End(Not run)
```

`coef.betadanish` *Extract Coefficients*

Description

Extract Coefficients

Usage

```
## S3 method for class 'betadanish'
coef(object, ...)
```

Arguments

`object`	An object of class ‘betadanish‘.
`...`	Further arguments passed to or from other methods.

Value

A named numeric vector of maximum likelihood parameter estimates.

`compare_distributions` *Compare Beta-Danish with Standard Distributions*

Description

Fits standard lifetime distributions (Weibull, Log-Normal, Log-Logistic, Gamma, Exponential) using the ‘flexsurv‘ package and compares them to the Beta-Danish fit.

Usage

```
compare_distributions(object)
```

Arguments

`object`	A fitted ‘betadanish‘ object.

Value

A ranked data frame of model comparisons.

`compare_models` *Compare Nested Beta-Danish Models*

Description

Performs a Likelihood Ratio Test (LRT) between the 4-parameter full model and the 3-parameter submodel.

Usage

```
compare_models(full_model, sub_model)
```

Arguments

`full_model`	A fitted 4-parameter ‘betadanish‘ object.
`sub_model`	A fitted 3-parameter ‘betadanish‘ object.

Value

A data frame with the test statistic and p-value.

`fit_bd_aft` *Fit Beta-Danish AFT Regression Model*

Description

Fits an Accelerated Failure Time (AFT) regression model using the Complementary Exponentiated Danish (CED) baseline (Beta-Danish with a=1).

Usage

```
fit_bd_aft(formula, data, n_starts = 10, method = "BFGS")
```

Arguments

`formula`	A survival formula (e.g., ‘Surv(time, status) ~ age + treatment‘).
`data`	A data frame containing the variables.
`n_starts`	Integer; number of random starts for optimization.
`method`	Optimization method passed to ‘maxLik‘.

Details

To ensure identifiability, the shape parameter ‘a‘ is fixed to 1. The scale parameter ‘k‘ is linked to covariates via ‘k_i = exp(X_i Positive coefficients in ‘delta‘ indicate a larger ‘k‘, which corresponds to shorter survival times (accelerated failure).

Value

An object of class ‘bd_aft‘.

`fit_bd_competing` *Fit Beta-Danish Competing Risks Model*

Description

Fits a parametric competing risks model assuming independent latent failure times, where each cause-specific baseline follows a 4-parameter Beta-Danish distribution.

Usage

```
fit_bd_competing(time, cause, n_starts = 5, method = "BFGS")
```

Arguments

`time`	Numeric vector of observed times.
`cause`	Integer vector of event causes. ‘0‘ indicates right-censored, and ‘1, 2, ..., m‘ indicate specific event causes.
`n_starts`	Integer; number of random starts for the joint optimization.
`method`	Optimization method passed to ‘maxLik‘.

Details

Under the assumption of independent latent failure times, the joint likelihood factorizes. The function first fits independent Beta-Danish models for each cause (treating other causes as censored) to find robust starting values, then optimizes the joint likelihood.

Value

An object of class ‘bd_competing‘.

`fit_bd_cure` *Fit Beta-Danish Cure Models*

Description

Fits mixture and promotion-time (non-mixture) cure models using the Beta-Danish AFT baseline.

Usage

```
fit_bd_cure(
  formula_aft,
  formula_cure,
  data,
  type = c("mixture", "promotion"),
  n_starts = 10,
  method = "BFGS"
)
```

Arguments

`formula_aft`	A formula for the latency component (e.g., ‘Surv(time, status) ~ age‘).
`formula_cure`	A one-sided formula for the incidence/cure component (e.g., ‘~ treatment‘).
`data`	A data frame containing the variables.
`type`	Character; either ‘"mixture"‘ or ‘"promotion"‘.
`n_starts`	Integer; number of random starts for optimization.
`method`	Optimization method passed to ‘maxLik‘.

Details

In the **mixture** model, the population is split into susceptible and cured fractions. The susceptible probability is modeled via logistic regression: ‘pi = exp(Z

In the **promotion-time** (non-mixture) model, the cure fraction is derived from a latent Poisson process of clonogenic cells: ‘theta = exp(Z The cure fraction is ‘exp(-theta)‘.

Value

An object of class ‘bd_cure‘.

`fit_betadanish` *Fit the Beta-Danish Distribution to Survival Data*

Description

Fits the Beta-Danish distribution using Maximum Likelihood Estimation (MLE). Supports both complete and right-censored data via ‘survival::Surv‘ objects.

Usage

```
fit_betadanish(formula, data, submodel = FALSE, n_starts = 10, method = "BFGS")
```

Arguments

`formula`	A formula object, with the response on the left of a ‘~‘ operator, and the terms on the right. The response must be a survival object as returned by the ‘Surv‘ function. Use ‘~ 1‘ for models without covariates.
`data`	A data frame containing the variables named in the formula.
`submodel`	Logical; if ‘TRUE‘, fits the 3-parameter submodel by fixing ‘a = 1‘.
`n_starts`	Integer; the number of random starting points to use for the optimization to ensure global convergence. Default is 10.
`method`	Character; the optimization method passed to ‘maxLik‘. Default is "BFGS".

Details

The optimization is performed on the log-transformed parameters to strictly enforce positivity constraints. The returned coefficients and variance-covariance matrix are transformed back to the natural scale using the Delta method.

Value

An object of S3 class ‘"betadanish"‘, containing the parameter estimates, log-likelihood, variance-covariance matrix, and convergence diagnostics.

Examples

```
# Simulate some data
set.seed(123)
sim_time <- rbetadanish(100, a = 1.5, b = 2, c = 3, k = 0.5)
sim_status <- sample(c(0, 1), 100, replace = TRUE, prob = c(0.2, 0.8))
dat <- data.frame(time = sim_time, status = sim_status)

# Fit the 4-parameter model
fit <- fit_betadanish(survival::Surv(time, status) ~ 1, data = dat)
summary(fit)

# Fit the 3-parameter submodel
fit_sub <- fit_betadanish(survival::Surv(time, status) ~ 1, data = dat, submodel = TRUE)
summary(fit_sub)
```

`gof_betadanish` *Goodness-of-Fit Statistics for Beta-Danish Models*

Description

Computes Information Criteria (AIC, BIC, HQIC, AICC) and the Kolmogorov-Smirnov (K-S) statistic for a fitted Beta-Danish model.

Usage

```
gof_betadanish(object)
```

Arguments

`object`	A fitted ‘betadanish‘ object.

Value

A list containing the Information Criteria and the K-S statistic.

`leukemia` *Acute Myelogenous Leukemia Survival*

Description

Survival times (in weeks) for 23 patients with acute myelogenous leukemia. A classic, small dataset perfect for fast testing of censored data workflows.

Usage

```
leukemia
```

Format

A data frame with 23 rows and 3 columns:

time Survival time in weeks

status Event indicator (1 = event, 0 = censored)

group Treatment group (Maintained vs Non-maintained)

Source

Miller, R. G. (1997). Survival Analysis. Wiley.

Examples

```
data(leukemia)

fit <- fit_betadanish(survival::Surv(time, status) ~ 1, data = leukemia)
```

`logLik.betadanish` *Extract Log-Likelihood*

Description

Extract Log-Likelihood

Usage

```
## S3 method for class 'betadanish'
logLik(object, ...)
```

Arguments

`object`	An object of class ‘betadanish‘.
`...`	Further arguments passed to or from other methods.

Value

An object of class logLik containing the maximized log-likelihood value, with degrees of freedom and number of observations stored as attributes.

`melanoma` *Malignant Melanoma Survival After Surgery*

Description

Survival times for 205 patients with malignant melanoma after surgery. This rich clinical dataset includes multiple covariates and heavy censoring.

Usage

```
melanoma
```

Format

A data frame with 205 rows and 6 columns:

time Survival time in days

status Event indicator (1 = died from melanoma, 0 = alive, 2 = other death)

thickness Tumor thickness in mm

sex Patient sex (1 = male, 0 = female)

age Patient age in years

ulcer Ulceration indicator (1 = present, 0 = absent)

year Year of operation

Source

Andersen, P. K., Borgan, O., Gill, R. D., & Keiding, N. (1993). Statistical Models Based on Counting Processes. Springer.

Examples

```
data(melanoma)

# Treat status 1 as event, others as censored
melanoma$event <- ifelse(melanoma$status == 1, 1, 0)
fit <- fit_betadanish(survival::Surv(time, event) ~ age + thickness, data = melanoma)
```

`plot.bd_aft` *Cox-Snell Residual Plot for AFT and Cure Fits*

Description

Diagnostic Cox-Snell residual plot for a fitted AFT or cure model.

Usage

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

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

Arguments

`x`	A fitted "`bd_aft`" or "`bd_cure`" object.
`...`	Further graphical parameters.

Value

Invisibly returns `x`.

Examples

```
set.seed(42)
n <- 300
x <- stats::rnorm(n)
k <- exp(-0.5 - 0.3 * x)
t_sim <- rbetadanish(n, a = 1, b = 2, c = 1.5, k = k)
status <- stats::rbinom(n, 1, 0.85)  # ~15% censoring
dat <- data.frame(time = t_sim, status = status, x = x)
fit <- fit_bd_aft(survival::Surv(time, status) ~ x, data = dat,
                  n_starts = 5)
plot(fit)
```

`plot.betadanish` *Plot Diagnostics for Beta-Danish Fit*

Description

Generates diagnostic plots for a fitted Beta-Danish model, including survival, hazard, density, PP, and QQ plots.

Usage

```
## S3 method for class 'betadanish'
plot(x, type = c("survival", "hazard", "density", "pp", "qq", "all"), ...)
```

Arguments

`x`	A fitted ‘betadanish‘ object.
`type`	Character string specifying the plot type: ‘"survival"‘, ‘"hazard"‘, ‘"density"‘, ‘"pp"‘, ‘"qq"‘, or ‘"all"‘.
`...`	Additional arguments passed to the base ‘plot‘ function.

Value

Invisibly returns the input betadanish object. Called mainly for its side effect of producing diagnostic plots.

`print.betadanish` *Print Method for Beta-Danish Fit*

Description

Print Method for Beta-Danish Fit

Usage

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

Arguments

`x`	An object of class ‘betadanish‘.
`...`	Further arguments passed to or from other methods.

Value

Invisibly returns the input betadanish object. Called mainly for its side effect of printing the fitted model summary.

`print.summary.betadanish` *Print Summary Method for Beta-Danish Fit*

Description

Print Summary Method for Beta-Danish Fit

Usage

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

Arguments

`x`	An object of class ‘summary.betadanish‘.
`...`	Further arguments passed to or from other methods.

Value

Invisibly returns the input summary.betadanish object. Called mainly for its side effect of printing the coefficient table and fit statistics.

`read_survival_data` *Read and Prepare Survival Data*

Description

A helper function to read survival data from CSV or Excel files and prepare it for analysis with the Beta-Danish package. It automatically handles missing status columns by assuming all observations are complete (uncensored).

Usage

```
read_survival_data(file, time_col, status_col = NULL, covar_cols = NULL)
```

Arguments

`file`	Character string specifying the path to the file.
`time_col`	Character string specifying the name of the time/survival column.
`status_col`	Character string specifying the name of the event/censoring indicator column. If ‘NULL‘ (default), the function assumes all observations are uncensored and creates a status column filled with 1s.
`covar_cols`	Character vector specifying the names of covariate columns to keep. If ‘NULL‘ (default), no covariates are kept.

Details

The function checks the file extension to determine how to read the data. For ‘.xlsx‘ or ‘.xls‘ files, the ‘readxl‘ package must be installed. Missing values (‘NA‘) in the specified columns will cause those rows to be dropped with a warning.

Value

A clean ‘data.frame‘ containing the ‘time‘, ‘status‘, and any specified covariates, ready to be passed to ‘fit_betadanish()‘.

Examples

```
tmp <- tempfile(fileext = ".csv")
example_data <- data.frame(
  survival_time = c(5, 8, 12, 16),
  status = c(1, 1, 0, 1)
)
write.csv(example_data, tmp, row.names = FALSE)
dat <- read_survival_data(tmp, time_col = "survival_time", status_col = "status")
unlink(tmp)
```

`remission` *Bladder Cancer Remission Times*

Description

Remission times (in months) for 128 bladder cancer patients. This is a complete (uncensored) sample widely used in lifetime distribution literature to demonstrate decreasing or right-skewed hazard rates.

Usage

```
remission
```

Format

A data frame with 128 rows and 2 columns:

time Remission time in months

status Event indicator (1 = event occurred)

Source

Lee, E. T., & Wang, J. W. (2003). Statistical Methods for Survival Data Analysis (3rd ed.). Wiley.

Examples

```
data(remission)

fit <- fit_betadanish(survival::Surv(time, status) ~ 1, data = remission)
summary(fit)
```

`report_betadanish` *Create a compact report from a BetaDanish model fit*

Description

Create a compact report from a BetaDanish model fit

Usage

```
report_betadanish(fit)
```

Arguments

`fit`	A fitted BetaDanish model object.

Value

A list containing model summary information.

`simulate_bd_cure_data` *Simulate Beta-Danish Cure Data*

Description

Generates synthetic survival data from a Beta-Danish mixture or promotion-time cure model, incorporating covariates.

Usage

```
simulate_bd_cure_data(
  n,
  type = c("mixture", "promotion"),
  a = 1,
  b = 2,
  c = 1.5,
  delta,
  gamma,
  X,
  Z,
  target_censor = 0.3,
  seed = NULL
)
```

Arguments

`n`	Integer; number of observations.
`type`	Character; ‘"mixture"‘ or ‘"promotion"‘.
`a, b, c`	Numeric; baseline shape parameters.
`delta`	Numeric vector; coefficients for the latency scale ‘k‘.
`gamma`	Numeric vector; coefficients for the incidence/cure component.
`X`	Matrix; design matrix for latency (must match length of ‘delta‘).
`Z`	Matrix; design matrix for incidence (must match length of ‘gamma‘).
`target_censor`	Numeric; target proportion of censoring to calibrate the exponential censoring rate. Default is 0.3.
`seed`	Integer; optional seed.

Value

A list containing the simulated ‘data‘ (time, status), the ‘cured‘ indicator, and the true parameters.

`simulate_bd_data` *Simulate Data from the Beta-Danish Distribution*

Description

Generates synthetic survival data from the Beta-Danish distribution, with optional right-censoring.

Usage

```
simulate_bd_data(n, a, b, c, k, censor_rate = 0, seed = NULL)
```

Arguments

`n`	Integer; number of observations to simulate.
`a, b, c, k`	Numeric; parameters of the Beta-Danish distribution.
`censor_rate`	Numeric; rate parameter for the exponential censoring distribution. If ‘0‘ (default), no censoring is applied.
`seed`	Integer; optional seed for reproducibility.

Value

A data frame with columns ‘time‘ and ‘status‘.

Examples

```
# Simulate complete data
dat <- simulate_bd_data(n = 100, a = 1.5, b = 2, c = 3, k = 0.5)

# Simulate censored data
dat_cens <- simulate_bd_data(n = 100, a = 1.5, b = 2, c = 3, k = 0.5, censor_rate = 0.1)
```

`summary.betadanish` *Summary Method for Beta-Danish Fit*

Description

Summary Method for Beta-Danish Fit

Usage

```
## S3 method for class 'betadanish'
summary(object, ...)
```

Arguments

`object`	An object of class ‘betadanish‘.
`...`	Further arguments passed to or from other methods.

Value

An object of class summary.betadanish containing coefficient estimates, standard errors, test statistics, p-values, log-likelihood, and model selection criteria.

`transplant` *Bone Marrow Transplant Survival*

Description

Survival times for 91 patients with refractory acute lymphoblastic leukemia who received either an allogeneic or autologous bone marrow transplant. This dataset includes right-censoring and a treatment covariate, making it ideal for demonstrating cure-rate models and AFT regression.

Usage

```
transplant
```

Format

A data frame with 91 rows and 3 columns:

time Survival time in days

status Event indicator (1 = death/relapse, 0 = censored)

group Treatment group (0 = Allogeneic, 1 = Autologous)

Source

Klein, J. P., & Moeschberger, M. L. (2003). Survival Analysis: Techniques for Censored and Truncated Data (2nd ed.). Springer.

Examples

```
data(transplant)

# Fit a model with a covariate
fit <- fit_bd_aft(survival::Surv(time, status) ~ group, data = transplant)
```

`vcov.betadanish` *Extract Variance-Covariance Matrix*

Description

Extract Variance-Covariance Matrix

Usage

```
## S3 method for class 'betadanish'
vcov(object, ...)
```

Arguments

`object`	An object of class ‘betadanish‘.
`...`	Further arguments passed to or from other methods.

Value

A numeric variance-covariance matrix for the estimated model parameters.

Index