

Package: FrailtyPackPlus (via r-universe)

October 28, 2026

Type Package

Title Shared, Correlated, Additive, and Zero-Inflated Frailty Models

Version 0.1.0

Description Fits and compares shared, correlated, additive, zero-inflated promotion cure-rate, and k-out-of-M one-shot device frailty survival models under Bayesian Markov chain Monte Carlo (MCMC) and Maximum Likelihood Estimation (MLE) frameworks. Supports ten baseline hazard distributions (Weibull, Exponential, Generalized Exponential, Exponential Power, Linear Failure Rate, Generalized Log-Logistic, Generalized Weibull, Generalized Rayleigh, Weighted Exponential, Extended Weibull) and eight frailty distributions (Gamma, Inverse Gaussian, Positive Stable, Compound Poisson, Compound Negative Binomial, Generalized Lindley, Weighted Lindley, Mixture of Inverse Gaussians). Implements reversed hazard rate (RHR) models for bivariate left-censored survival data, additive shared frailty models under random right censoring, and one-shot device reliability analysis. Provides goodness-of-fit testing, information criteria (AIC, BIC, AICc, HQIC, DIC, EAIC, EBIC, LPML), convergence diagnostics (Geweke, Gelman-Rubin, Heidelberger-Welch), Bayes factors, residual analyses, and diagnostic visualizations. Methodologies are based on Hanagal and Dabade (2013) [doi:10.1016/j.spl.2013.07.015](https://doi.org/10.1016/j.spl.2013.07.015), Hanagal and Dabade (2015) [doi:10.1080/03610918.2012.661909](https://doi.org/10.1080/03610918.2012.661909), Hanagal and Bhambure (2015) [doi:10.1080/03610926.2013.804561](https://doi.org/10.1080/03610926.2013.804561), Hanagal and Bhambure (2016) [doi:10.3233/MAS-140308](https://doi.org/10.3233/MAS-140308), Hanagal, Pandey, and Ganguly (2017) [doi:10.1080/03610918.2015.1085559](https://doi.org/10.1080/03610918.2015.1085559), Hanagal and Pandey (2018) [doi:10.1080/03610926.2018.1549256](https://doi.org/10.1080/03610926.2018.1549256), Hanagal (2020) [doi:10.1080/03610926.2020.1736305](https://doi.org/10.1080/03610926.2020.1736305), Tyagi et al. (2021) [doi:10.1007/s40314-021-01666-5](https://doi.org/10.1007/s40314-021-01666-5), Pandey, Hanagal, Tyagi, and Gupta (2021) [doi:10.1142/S0218539321500406](https://doi.org/10.1142/S0218539321500406), Tyagi, Pandey, and Chesneau (2022) [doi:10.1007/s42519-021-00241-9](https://doi.org/10.1007/s42519-021-00241-9), Pandey, Hanagal, and Tyagi (2022) [doi:10.2478/stattrans-2022-0048](https://doi.org/10.2478/stattrans-2022-0048),

Tyagi, Pandey, and Chesneau (2022)
[doi:10.1007/s13571-022-00278-1](https://doi.org/10.1007/s13571-022-00278-1), Tyagi, Pandey, Hanagal, and
 Chesneau (2024) [doi:10.1080/03610918.2024.2391874](https://doi.org/10.1080/03610918.2024.2391874), Kiprotich
 et al. (2025) [doi:10.1080/02664763.2025.2560510](https://doi.org/10.1080/02664763.2025.2560510), Assuncao et
 al. (2026) [doi:10.1002/bimj.70162](https://doi.org/10.1002/bimj.70162), Chen et al. (1999)
[doi:10.1080/01621459.1999.10474198](https://doi.org/10.1080/01621459.1999.10474198), Dunn and Smyth (1996)
[doi:10.1080/10618600.1996.10474708](https://doi.org/10.1080/10618600.1996.10474708), Ling et al. (2021)
[doi:10.3390/math9233032](https://doi.org/10.3390/math9233032), Yu, So, and Balakrishnan (2025)
[doi:10.1016/B978-0-44-331694-4.00006-2](https://doi.org/10.1016/B978-0-44-331694-4.00006-2), Balakrishnan, Ling,
 and So (2021, ISBN:978-1-119-77795-3), and McGilchrist and
 Aisbett (1991) [doi:10.2307/2532138](https://doi.org/10.2307/2532138).

License GPL (>= 3)

Encoding UTF-8

LazyData true

ByteCompile true

Depends R (>= 4.0.0)

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

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

Language en-US

RoxygenNote 7.3.3

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 2026-10-27 14:30:02 UTC

RemoteUrl <https://github.com/cran/FrailtyPackPlus>

RemoteRef HEAD

RemoteSha 94638b8574366dcd081edc82029ef4d1457d8983

Contents

austrian_twin	5
base_cumhaz	6
base_haz	7
bayes_factor	7
bayes_factor_rhr	8
calc_loss_estimates	9
classH	10
coef.oneshot_kofm	11

compare_rhr_models	11
component_moments	12
component_moments.oneshot_frailty	13
component_moments.oneshot_kofm	14
compute_mcmc_diagnostics	15
compute_p0_p1	16
compute_probabilities	17
cond_prob_rhr	18
confint.oneshot_kofm	19
corrFrailty	20
corrFrailty.control	22
cpo	23
crhr_baseline	24
criteria	25
cross_ratio_rhr	26
d_rhr_baseline	27
d2laplace_frailty	28
dens_proper	28
dependence	29
diagnostics_mcmc	30
dic	31
dlaplace_frailty	32
eaic	33
ebic	33
em_oneshot_frailty	34
em_oneshot_kofm	35
evaluate_ziprom	37
fb_sim_frailty	38
fisher_info_oneshot	39
fit_add_frailty_bayes	41
fit_add_frailty_mle	43
fit_rhr_frailty_bayes	45
fit_rhr_frailty_mle	48
fitted.oneshot_kofm	49
forecast_ziprom	50
frailty_bayes	51
frailty_compare	53
frailty_fit	54
frailty_mle	54
generate_powerset	56
hpd_interval	56
insulin_gdm	57
kendall_tau_rhr	58
kidney_bivariate	59
kidney_frailty	60
kidney_infection	61
laplace_frailty	62
logLik.frailty_bayes	63

logLik.oneshot_kofm	63
loglik_add_frailty	64
loglik_biv_rhr	65
loglik_oneshot	66
lpml	68
mice_tumor	68
mttf_k_out_of_m	69
mttf_k_out_of_m.oneshot_frailty	69
mttf_k_out_of_m.oneshot_kofm	71
oneshot_frailty	73
oneshot_kofm	75
p_rhr_baseline	78
plot.add_frailty	78
plot.frailty_bayes	80
plot.oneshot_frailty	80
plot.oneshot_kofm	82
plot.rhr_bayes_fit	83
plot.rhr_mle_fit	84
plot.ziprom_fit	84
plot_activation_compare	86
predict.frailty_bayes	87
predict.oneshot_kofm	88
predict.ziprom_fit	88
print.frailty_bayes	90
print.oneshot_kofm	91
print.summary.oneshot_kofm	91
print.ziprom_fit	92
q_rhr_baseline	93
quantile_proper	94
reliability_k_out_of_m	95
reliability_k_out_of_m.oneshot_frailty	95
reliability_k_out_of_m.oneshot_kofm	97
residuals.frailty_bayes	98
residuals.oneshot_kofm	99
retinopathy_frailty	99
rhr_baseline	100
rzicr_promotion	100
sim_add_frailty	101
sim_biv_rhr_frailty	103
sim_copula	104
sim_corrFrailty	105
sim_data4	107
sim_frailty	108
sim_oneshot	109
sim_oneshot_frailty	111
sim_oneshot_frailty_lifetimes	111
sim_ziprom	112
summary.frailty_bayes	114

summary.frailty_mle	114
summary.oneshot_kofm	115
surv_proper	115
test_baseline	116
test_independence	117
var_frailty	117
vcov.oneshot_kofm	118
ziprom_core_dens	119
ziprom_core_surv	120
ziprom_fit	121
ziprom_loglik	123

Index	126
--------------	------------

austrian_twin	<i>Dataset documentation for Frailty package Australian Twin Appendectomy Data</i>
---------------	--

Description

Bivariate survival dataset containing ages at onset of appendectomy in 350 male twin pairs from the Australian Twin Registry, with left-censoring.

Usage

```
austrian_twin
```

Format

A data frame with 350 rows and 7 variables:

- t1** Age at onset of appendectomy (in years) for twin 1 (lifetime).
- t2** Age at onset of appendectomy (in years) for twin 2 (lifetime).
- d1** Censoring indicator for twin 1 (1 = observed, 0 = left-censored).
- d2** Censoring indicator for twin 2 (1 = observed, 0 = left-censored).
- age** Age of the twin pair (covariate K1).
- k2** Binary indicator of appendectomy condition for twin 1 (covariate K2).
- k3** Binary indicator of appendectomy condition for twin 2 (covariate K3).

Source

Duffy, D. L., Martin, N. G., & Mathews, J. D. (1990). Appendectomy in Australian twins. *American Journal of Human Genetics*, 47(3), 590-592.

References

- Tyagi, S., Pandey, A., & Chesneau, C. (2022). Identifying the effects of observed and unobserved risk factors using weighted Lindley shared regression model. *Journal of Statistical Theory and Practice*, 16(1), 16. doi:10.1007/s42519021002419.
- Tyagi, S., Pandey, A., & Chesneau, C. (2022). Weighted Lindley shared regression model for bivariate left censored data. *Sankhya B: The Indian Journal of Statistics*, 84(2), 754-784. doi:10.1007/s13571022002781.
- Pandey, A., Hanagal, D. D., Gupta, P., & Tyagi, S. (2020). Analysis of Australian twin data using generalized inverse Gaussian shared frailty models based on reversed hazard rate. *International Journal of Statistics and Reliability Engineering*, 7(2), 219-235.
- Pandey, A., Hanagal, D. D., Tyagi, S., & Gupta, P. (2021). Generalized Lindley shared frailty based on reversed hazard rate. *International Journal of Reliability, Quality and Safety Engineering*, 28(5), 2150040. doi:10.1142/S0218539321500406.
- Gupta, P., Pandey, A., & Tyagi, S. (2022). Comparison of multiplicative frailty models under generalized log-logistic-II baseline distribution for counter heterogeneous left censored data. In *Statistical Techniques for Interdisciplinary Research* (pp. 97-108). ISBN: 978-93-91248-57-4.

Examples

```
data(austrian_twin)
summary(austrian_twin)
```

base_cumhaz

Cumulative Baseline Hazard Function

Description

Computes the cumulative baseline hazard $H_0(t)$ for Weibull, Gompertz, or Exponential baseline distributions.

Usage

```
base_cumhaz(
  t,
  shape,
  scale,
  baseline = c("weibull", "gompertz", "exponential")
)
```

Arguments

t	Numeric vector of positive survival times.
shape	Numeric vector or scalar representing baseline shape parameter(s) (> 0).
scale	Numeric vector or scalar representing baseline scale parameter(s) (> 0).
baseline	Character string indicating the baseline hazard distribution: "weibull" (default), "gompertz", or "exponential".

Value

Numeric vector of cumulative baseline hazard values.

base_haz	<i>Baseline Hazard Function</i>
----------	---------------------------------

Description

Computes the baseline hazard rate $h_0(t)$ for Weibull, Gompertz, or Exponential baseline distributions.

Usage

```
base_haz(t, shape, scale, baseline = c("weibull", "gompertz", "exponential"))
```

Arguments

t	Numeric vector of positive survival times.
shape	Numeric vector or scalar representing baseline shape parameter(s) (> 0).
scale	Numeric vector or scalar representing baseline scale parameter(s) (> 0).
baseline	Character string indicating the baseline hazard distribution: "weibull" (default), "gompertz", or "exponential".

Value

Numeric vector of baseline hazard rate values.

bayes_factor	<i>Bayes Factor Comparison Between Two Models</i>
--------------	---

Description

Computes the Bayes factor comparing Model 1 against Model 2 using the ratio of Pseudo Marginal Likelihoods (LPML) or harmonic mean marginal likelihood.

Usage

```
bayes_factor(model1, model2)
```

Arguments

model1	Fitted object of class "frailty_bayes".
model2	Fitted object of class "frailty_bayes".

Value

A list containing BF12 (Bayes Factor), log_BF12, and verbal interpretation according to Jeffreys' scale.

Examples

```
data(kidney_frailty)
m1 <- frailty_bayes(Surv(time1, time2, status1, status2) ~ age,
  data = kidney_frailty, frailty = "gamma", n_iter = 200, burn_in = 50)
m2 <- frailty_bayes(Surv(time1, time2, status1, status2) ~ age,
  data = kidney_frailty, frailty = "invgauss", n_iter = 200, burn_in = 50)
bayes_factor(m1, m2)
```

 bayes_factor_rhr

Bayes Factor Calculation for Model Comparison

Description

Computes the Bayes factor comparing two Bayesian shared RHR frailty models based on log pseudo-marginal likelihoods (LPML) or marginal deviances:

$$BF_{12} = \exp(LPML_1 - LPML_2)$$

Usage

```
bayes_factor_rhr(model1, model2)
```

Arguments

model1	Fitted object of class "rhr_bayes_fit".
model2	Fitted object of class "rhr_bayes_fit".

Details

Evaluates the strength of evidence between two Bayesian models using the scale of Kass and Raftery (1995) based on $2 \ln(BF_{12})$.

Value

A list containing the following components:

log_BF12 Natural logarithm of the Bayes factor $\ln(BF_{12})$.

BF12 Bayes factor value BF_{12} .

two_log_BF12 Twice the natural logarithm of the Bayes factor $2 \ln(BF_{12})$.

LPML_Model1 Log pseudo-marginal likelihood of Model 1.

LPML_Model2 Log pseudo-marginal likelihood of Model 2.

Evidence Verbal interpretation of evidence according to the Kass and Raftery (1995) scale.

Examples

```
data(austrian_twin)
fit1 <- fit_rhr_frailty_bayes(cbind(t1, t2, d1, d2) ~ age, data = austrian_twin[1:30, ],
                             frailty = "WL", baseline = "GE", n_iter = 40, burn_in = 10)
fit2 <- fit_rhr_frailty_bayes(cbind(t1, t2, d1, d2) ~ age, data = austrian_twin[1:30, ],
                             frailty = "WL", baseline = "EG", n_iter = 40, burn_in = 10)
bayes_factor_rhr(fit1, fit2)
```

calc_loss_estimates	<i>Bayesian Point Estimation and Posterior Risks under Multiple Loss Functions</i>
---------------------	--

Description

Computes Bayes point estimates and corresponding posterior risks under Squared Error Loss Function (SELF), Modified Quadratic Squared Error Loss Function (MQSELF), and Precautionary Loss Function (PLF), along with equal-tailed Bayesian Credible Intervals and Highest Posterior Density (HPD) bounds.

Usage

```
calc_loss_estimates(samples, level = 0.95)
```

Arguments

<code>samples</code>	Matrix or data frame of MCMC posterior draws (rows = iterations, columns = parameters).
<code>level</code>	Confidence level for credible intervals (default is 0.95).

Value

A data frame containing:

Parameter Parameter name.

Est_SELF Bayes estimate under SELF (posterior mean).

Risk_SELF Posterior risk under SELF (posterior variance).

Est_MQSELF Bayes estimate under MQSELF.

Risk_MQSELF Posterior risk under MQSELF.

Est_PLF Bayes estimate under PLF.

Risk_PLF Posterior risk under PLF.

LCL Lower bound of equal-tailed credible interval.

UCL Upper bound of equal-tailed credible interval.

References

- Tyagi, S., Pandey, A., Hanagal, D. D., and Chesneau, C. (2024). A study on comparisons of additive regression frailty models to counter heterogeneity: Bayesian strategies and case study. *Communications in Statistics - Simulation and Computation*. doi:[10.1080/03610918.2024.2391874](https://doi.org/10.1080/03610918.2024.2391874)
- Norstrom, J. G. (1996). The use of precautionary loss functions in risk analysis. *IEEE Transactions on Reliability*, 45(3), 400-403. doi:[10.1109/24.536992](https://doi.org/10.1109/24.536992)

Examples

```
# Simulated posterior samples
set.seed(123)
post_draws <- matrix(rgamma(1000 * 3, shape = 4, rate = 2), ncol = 3)
colnames(post_draws) <- c("zeta1", "delta1", "xi1")
calc_loss_estimates(post_draws)
```

classH	<i>Modified Class-H Failure Modes Data</i>
--------	--

Description

The modified Class-H failure modes data presented in Table 10 of Ling et al. (2021), adapted from motorette insulation testing originally reported by Nelson (2009). The dataset records the outcomes of one-shot inspections across 8 test groups under four elevated temperature conditions with $M = 2$ failure modes: Turn failure (T) and Ground failure (G).

Usage

```
classH
```

Format

A data frame with 8 rows and 7 variables:

- group Test group index ($i = 1, \dots, 8$).
- stress Accelerated temperature stress level in degrees Fahrenheit ($s_i = 374, 428, 464, 500$).
- time Pre-specified inspection time in hours ($\tau_i = 800, \dots, 10000$).
- n_none Number of functioning devices (neither component malfunctioned, $X = \emptyset$).
- n_T Number of devices with only Turn component failure ($X = \{T\}$).
- n_G Number of devices with only Ground component failure ($X = \{G\}$).
- n_both Number of devices with both Turn and Ground failures ($X = \{T, G\}$).

Source

- Ling, M.-H., Balakrishnan, N., Yu, C., and So, H. Y. (2021). Inference for One-Shot Devices with Dependent k-Out-of-M Structured Components under Gamma Frailty. *Mathematics*, 9(23), 3032. doi:[10.3390/math9233032](https://doi.org/10.3390/math9233032).
- Nelson, W. B. (2009). *Accelerated Testing: Statistical Models, Test Plans, and Data Analysis*. John Wiley & Sons, Hoboken, NJ.

Examples

```
data(classH)
head(classH)
```

coef.oneshot_kofm	<i>Extract Model Coefficients</i>
-------------------	-----------------------------------

Description

Extracts estimated parameters from a fitted "oneshot_kofm" object.

Usage

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

Arguments

object	A fitted model object of class "oneshot_kofm".
...	Additional arguments.

Value

A named numeric vector of parameter estimates.

compare_rhr_models	<i>Compare Competing Shared RHR Frailty Models</i>
--------------------	--

Description

Tabulates and compares multiple fitted Bayesian or MLE shared frailty models using information criteria (AIC, BIC, AICc, HQIC, DIC, LPML), Kendall's tau, and frailty variance.

Usage

```
compare_rhr_models(..., models_list = NULL)
```

Arguments

...	Fitted objects of class "rhr_bayes_fit" or "rhr_mle_fit".
models_list	Optional list of fitted model objects.

Details

Compares goodness-of-fit and penalized likelihood criteria across competing models. Models with lower AIC, BIC, AICc, HQIC, and DIC, and higher LPML (closer to zero) represent superior fits.

Value

A data frame with one row per model containing:

Model Name or label of the model.

Frailty Frailty distribution.

Baseline Baseline distribution.

Method Estimation methodology ("Bayes" or "MLE").

LogLik Log-likelihood at parameter point estimates.

AIC Akaike Information Criterion.

BIC Bayesian Information Criterion.

AICc Corrected Akaike Information Criterion.

HQIC Hannan-Quinn Information Criterion.

DIC Deviance Information Criterion (for Bayesian fits).

LPML Log Pseudo Marginal Likelihood (for Bayesian fits).

Kendall_Tau Estimated Kendall's tau rank correlation.

Frailty_Var Estimated theoretical variance of the frailty variable.

Examples

```
data(austrian_twin)
fit1 <- fit_rhr_frailty_bayes(cbind(t1, t2, d1, d2) ~ age, data = austrian_twin[1:30, ],
                             frailty = "WL", baseline = "GE", n_iter = 40, burn_in = 10)
fit2 <- fit_rhr_frailty_bayes(cbind(t1, t2, d1, d2) ~ age, data = austrian_twin[1:30, ],
                             frailty = "WL", baseline = "EG", n_iter = 40, burn_in = 10)
compare_rhr_models(WL_GE = fit1, WL_EG = fit2)
```

component_moments	<i>Expected Component Lifetimes and Moments</i>
-------------------	---

Description

Calculates the expected mean failure times and moments for individual components under gamma frailty.

Usage

```
component_moments(object, ...)
```

Arguments

object	A fitted model object of class "oneshot_kofm" or "oneshot_frailty".
...	Additional arguments passed to methods.

Value

A data frame or list containing component mean lifetimes and standard deviations.

component_moments.oneshot_frailty

Marginal Moments for Components under Shared Gamma Frailty

Description

Computes the marginal scale, shape, mean lifetime, variance, and standard deviation for individual components under the Power Lomax marginal distribution.

Usage

```
## S3 method for class 'oneshot_frailty'
component_moments(object, stress = NULL, ...)
```

Arguments

object	A fitted model object of class "oneshot_frailty".
stress	Optional numeric vector or matrix of stress covariate values.
...	Additional arguments passed to other methods.

Details

Under Weibull baseline lifetime and shared gamma frailty with variance β , the marginal unconditional lifetime distribution of component m is Power Lomax with survival function

$$S(t) = (1 + \beta(\lambda t)^\alpha)^{-1/\beta}$$

The r -th raw moment is given in closed form by:

$$E[T^r] = \frac{\Gamma(1 + r/\alpha)\Gamma(1/\beta - r/\alpha)}{\lambda^r \beta^{r/\alpha} \Gamma(1/\beta)}$$

valid when $\beta < \alpha/r$.

Value

A data frame containing:

- component** Component name.
- lambda** Estimated scale parameter λ .
- alpha** Estimated shape parameter α .
- beta** Frailty variance parameter β .
- mean** Marginal mean lifetime (or Inf if $\beta \geq \alpha$).
- variance** Marginal variance (or Inf if $\beta \geq \alpha/2$).
- sd** Marginal standard deviation.

References

Yu, C., So, H. Y., and Balakrishnan, N. (2025). *Inference for one-shot devices with Weibull component lifetimes under gamma frailty model*. In: Stochastic Modeling and Statistical Methods, Elsevier Inc., Chapter 1, pp. 1–34. doi:10.1016/B9780443316944.000062.

Examples

```
data(classH)
start_par <- c(-3.997, -0.009, -14.412, 0.003, 0.004, -0.001, -5.493e-4, -0.005, 0.082)
fit_w <- oneshot_frailty(
  data = classH,
  time = "time",
  stress = "stress",
  counts = c("n_none", "n_T", "n_G", "n_both"),
  baseline = "weibull",
  frailty = TRUE,
  start = start_par
)
component_moments(fit_w, stress = 400)
```

component_moments.oneshot_kofm

Marginal Component Moments

Description

Computes the marginal scale, mean, and variance for each failure mode under Lomax / Exponential distributions with gamma frailty.

Usage

```
## S3 method for class 'oneshot_kofm'
component_moments(object, stress, ...)
```

Arguments

object	A fitted model object of class "oneshot_kofm".
stress	A numeric scalar or vector representing operating conditions.
...	Additional arguments passed to other methods.

Details

Under gamma frailty, the marginal distribution of component m is Lomax (Pareto Type II) with:

$$\mu_{T_m} = \frac{1}{\lambda_m(1-\beta)}, \quad \beta < 1$$

$$\sigma_{T_m}^2 = \frac{1}{\lambda_m^2(1-\beta)^2(1-2\beta)}, \quad \beta < 0.5$$

Missing values are not permitted.

Value

A data frame containing:

component Component name.
stress Operating stress level.
lambda Estimated scale parameter $\lambda_{0,m}$.
mean Marginal mean lifetime μ_{T_m} .
variance Marginal lifetime variance $\sigma_{T_m}^2$ (if $\beta < 0.5$).
sd Marginal standard deviation.

References

Ling, M.-H., Balakrishnan, N., Yu, C., and So, H. Y. (2021). Inference for One-Shot Devices with Dependent k-Out-of-M Structured Components under Gamma Frailty. *Mathematics*, 9(23), 3032. [doi:10.3390/math9233032](https://doi.org/10.3390/math9233032).

Examples

```
data(classH)
fit <- oneshot_kofm(
  data = classH,
  time = "time",
  stress = "stress",
  counts = c("n_none", "n_T", "n_G", "n_both"),
  baseline = "exponential"
)
component_moments(fit, stress = 356)
```

compute_mcmc_diagnostics

Convergence and Summary Diagnostics for MCMC Chains

Description

Computes Geweke z-score, Gelman-Rubin diagnostic (multichain), Effective Sample Size (ESS), Monte Carlo Standard Error (MCSE), and Highest Posterior Density (HPD) intervals.

Usage

```
compute_mcmc_diagnostics(
  chain,
  param_names = NULL,
  frac1 = 0.1,
  frac2 = 0.5,
  prob = 0.95
)
```

Arguments

chain	Numeric matrix of MCMC samples (iterations x parameters).
param_names	Character vector of parameter names.
frac1	Fraction of chain to use for Geweke test first part (default 0.1).
frac2	Fraction of chain to use for Geweke test second part (default 0.5).
prob	Credibility level for HPD intervals (default 0.95).

Value

A data frame containing posterior summary statistics and diagnostics.

compute_p0_p1	<i>Compute Multinomial Logit Probabilities for Zero-Inflation and Cure Fraction</i>
---------------	---

Description

Evaluates the probabilities of zero-failure p_0 , cure fraction p_1 , susceptible fraction $1 - p_0 - p_1$, and the latent Poisson parameter $\theta = -\log(p_1)$.

Usage

compute_p0_p1(X1, X2, beta1, beta2)

Arguments

X1	Design matrix for zero-inflation linear predictor.
X2	Design matrix for cure fraction linear predictor.
beta1	Numeric vector of coefficients for zero-inflation.
beta2	Numeric vector of coefficients for cure fraction.

Value

A list containing p0, p1, theta, and susceptible.

Description

Computes the unconditional joint survival functions $g_i(S, \tau_i)$ and the observation probabilities $P_i(X, \tau_i)$ for all subsets of malfunctioned components $X \subseteq \Omega$ under the shared gamma frailty model.

Usage

```
compute_probabilities(
  par,
  X_cov,
  tau,
  M,
  baseline = c("weibull", "exponential"),
  frailty = TRUE,
  powerset = NULL
)
```

Arguments

par	A numeric vector of model parameters: scale coefficients a, shape coefficients b (for Weibull baseline), and frailty variance beta.
X_cov	A numeric matrix of covariates (with leading column of ones for intercept), having I rows corresponding to test groups.
tau	A non-negative numeric vector of inspection times.
M	An integer specifying the number of components ($M \geq 1$).
baseline	A character string: "exponential" (default) or "weibull".
frailty	A logical value: TRUE (default) if frailty parameter $\beta > 0$ is included, or FALSE for the independence model ($\beta = 0$).
powerset	An optional list of power set subsets generated by generate_powerset .

Details

For a subset $S \subseteq \Omega$, the unconditional joint survival probability is

$$g_i(S, \tau_i) = \left(1 + \beta \sum_{m \in S} (\lambda_{i,m} \tau_i)^{\alpha_{i,m}} \right)^{-1/\beta}$$

when $\beta > 0$, and $\exp(-\sum_{m \in S} (\lambda_{i,m} \tau_i)^{\alpha_{i,m}})$ when $\beta = 0$. By the principle of inclusion-exclusion, the probability of observing exactly the malfunctioned subset X is

$$P_i(X, \tau_i) = \sum_{Y \subseteq X} (-1)^{|Y|} g_i(Y \cup X^c, \tau_i)$$

where $X^c = \Omega \setminus X$.

Missing values are not permitted in `par`, `X_cov`, or `tau`.

Value

A list containing:

prob_matrix An $I \times 2^M$ numeric matrix of subset probabilities $P_i(X, \tau_i)$.

lambda_mat An $I \times M$ numeric matrix of scale parameters $\lambda_{i,m}$.

alpha_mat An $I \times M$ numeric matrix of shape parameters $\alpha_{i,m}$.

beta A non-negative numeric scalar representing frailty variance.

powerset The list of 2^M power set subsets.

References

Ling, M.-H., Balakrishnan, N., Yu, C., and So, H. Y. (2021). Inference for One-Shot Devices with Dependent k-Out-of-M Structured Components under Gamma Frailty. *Mathematics*, 9(23), 3032. [doi:10.3390/math9233032](https://doi.org/10.3390/math9233032).

Examples

```
X_cov <- matrix(c(1, 1, 374, 428), ncol = 2)
tau <- c(8000, 2500)
par <- c(-4.99, -0.005, -16.44, 0.018, 0.5)
res <- compute_probabilities(par, X_cov, tau, M = 2, baseline = "exponential", frailty = TRUE)
res$prob_matrix
```

cond_prob_rhr

Conditional Probability Measure under Reversed Hazard Rate

Description

Computes the bivariate conditional dependence probability ratio $\chi(t_1, t_2) = S(t_1, t_2) / (S(t_1)S(t_2))$.

Usage

```
cond_prob_rhr(
  t1,
  t2,
  x1 = NULL,
  x2 = NULL,
  beta = NULL,
  frailty = "WL",
  theta = 1,
  mu = NULL,
  theta2 = NULL,
  baseline = "GE",
```

```

    params1 = c(zeta = 1, delta = 1),
    params2 = c(zeta = 1, delta = 1)
  )

```

Arguments

t1	Numeric scalar or vector of first component times.
t2	Numeric scalar or vector of second component times.
x1	Numeric matrix or vector of covariates for first component.
x2	Numeric matrix or vector of covariates for second component.
beta	Numeric vector of regression coefficients.
frailty	Character string indicating frailty distribution.
theta	Numeric parameter for frailty distribution.
mu	Optional secondary parameter for "GL".
theta2	Optional second parameter for "GIG".
baseline	Character string indicating baseline distribution.
params1	Numeric named vector of baseline parameters for first component.
params2	Numeric named vector of baseline parameters for second component.

Value

Numeric vector of conditional probability ratios.

Examples

```

cond_prob_rhr(15, 18, frailty = "WL", theta = 3.4, baseline = "GE",
  params1 = c(zeta = 12.4, delta = 0.11),
  params2 = c(zeta = 11.8, delta = 0.11))

```

confint.oneshot_kofm *Confidence Intervals for Model Parameters*

Description

Computes asymptotic confidence intervals for parameters of a "oneshot_kofm" model.

Usage

```

## S3 method for class 'oneshot_kofm'
confint(object, parm, level = 0.95, ...)

```

Arguments

object	A fitted model object of class "oneshot_kofm".
parm	A specification of which parameters are to be given confidence intervals.
level	The confidence level required (default 0.95).
...	Additional arguments.

Value

A numeric matrix with columns LCI and UCI.

corrFrailty

Fit Correlated Frailty Models for Bivariate Survival Data

Description

Fits parametric correlated frailty models to bivariate right-censored survival data using Bayesian Markov Chain Monte Carlo (MCMC) simulation or Maximum Likelihood Estimation (MLE). Supports Correlated Gamma, Inverse Gaussian, Positive Stable, and Compound Poisson frailty distributions paired with Generalized Weibull, Generalized Log-Logistic, Weibull, Exponential, Pareto, Linear Failure Rate, and Burr (Type XII) baseline hazards.

Usage

```
corrFrailty(
  formula,
  data,
  id = NULL,
  frailty = c("gamma", "invgaussian", "posstable", "comppoisson", "none"),
  baseline = c("genweibull", "genloglogistic", "weibull", "exponential", "pareto", "lfr",
    "burr"),
  baseline2 = baseline,
  method = c("mcmc", "mle"),
  control = corrFrailty.control(),
  ...
)
```

Arguments

formula	A formula object. For wide format data, use $\text{Surv}(t1, d1, t2, d2) \sim x1 + x2$ or $\text{cbind}(t1, d1, t2, d2) \sim x1 + x2$. For long format data with paired clusters, use $\text{Surv}(\text{time}, \text{status}) \sim x1 + x2$ with argument <code>id</code> .
data	A data.frame containing the survival times, censoring indicators, and covariates.
id	Optional variable identifying paired observations in long format data (must have exactly 2 observations per cluster).

frailty	Character specifying the frailty distribution. One of "gamma" (default), "invgaussian", "posstable", "comppoisson", or "none".
baseline	Character specifying the baseline hazard distribution for the first survival time T1. One of "genweibull", "genloglogistic", "weibull" (default), "exponential", "pareto", "lfr", or "burr".
baseline2	Character specifying the baseline hazard distribution for the second survival time T2. Defaults to the same distribution as baseline.
method	Character specifying the estimation method: "mcmc" (Bayesian MCMC, default) or "mle" (Maximum Likelihood Estimation).
control	List of control parameters created by corrFrailty.control .
...	Additional arguments passed to control parameters.

Value

An object of class "corrFrailty" containing:

coefficients	Point estimates (posterior means for MCMC, MLEs for MLE).
summary.table	Detailed table of estimates, standard errors, credible/confidence intervals, and diagnostics.
method	Estimation method used ("mcmc" or "mle").
frailty	Frailty distribution.
baseline1	Baseline hazard distribution for T1.
baseline2	Baseline hazard distribution for T2.
logLik	Log-likelihood evaluated at parameter estimates.
deviance	Deviance ($-2 * \logLik$).
AIC	Akaike Information Criterion.
BIC	Bayesian Information Criterion.
DIC	Deviance Information Criterion (for MCMC).
pD	Effective number of parameters (for MCMC).
LPML	Log Pseudo-Marginal Likelihood (for MCMC).
chains	List of MCMC chains (for MCMC).
all.samples	Matrix of pooled posterior samples (for MCMC).
call	The matched function call.
data	The processed data matrix used for model fitting.

Author(s)

Shikhar Tyagi, Arvind Pandey, Vrijesh Tripathi

References

- Hanagal, D. D., Pandey, A., and Ganguly, A. (2017). Correlated gamma frailty models for bivariate survival data. *Communications in Statistics - Simulation and Computation*, 46(5), 3627-3644. <doi:10.1080/03610918.2015.1085559>
- Hanagal, D. D. and Pandey, A. (2018). Correlated inverse Gaussian frailty models for bivariate survival data. *Communications in Statistics - Theory and Methods*, 49(14), 3455-3475. <doi:10.1080/03610926.2018.1549256>
- Hanagal, D. D. (2020). Correlated positive stable frailty models. *Communications in Statistics - Theory and Methods*, 51(8), 2419-2436. <doi:10.1080/03610926.2020.1736305>
- Hanagal, D. D. (2020). Analysis of Kidney infection Data Using Correlated Inverse Gaussian Frailty Model. *Statistics and Applications*, 18(1), 1-19.

Examples

```
# Simulate small bivariate survival dataset
set.seed(123)
dat <- sim_corrFrailty(n = 30, frailty = "gamma",
                      frailty.params = list(sigma = 0.5, rho = 0.4),
                      baseline = "exponential",
                      baseline.params = list(lambda = 0.05),
                      beta = c(0.3), seed = 123)

# Fit Correlated Gamma Frailty Model using MCMC
fit <- corrFrailty(Surv(t1, d1, t2, d2) ~ x1, data = dat,
                  frailty = "gamma", baseline = "exponential",
                  method = "mcmc",
                  control = corrFrailty.control(n.iter = 100, n.burnin = 20,
                                                n.thin = 1, n.chains = 1))

print(fit)
summary(fit)
```

corrFrailty.control	<i>Control Parameters for corrFrailty Estimation</i>
---------------------	--

Description

Auxiliary function for setting MCMC and MLE optimization control parameters in [corrFrailty](#).

Usage

```
corrFrailty.control(
  n.iter = 5000,
  n.burnin = 1000,
  n.thin = 5,
  n.chains = 2,
  init = NULL,
  tune = NULL,
  adapt.iter = 500,
```

```

priors = list(),
seed = NULL,
optim.method = "BFGS",
optim.control = list()
)

```

Arguments

n.iter	Integer, total number of MCMC iterations per chain. Default is 5000.
n.burnin	Integer, number of burn-in iterations to discard. Default is 1000.
n.thin	Integer, thinning interval. Default is 5.
n.chains	Integer, number of parallel/independent MCMC chains. Default is 2.
init	Optional list of initial value vectors (one per chain).
tune	Optional numeric vector of initial proposal standard deviations.
adapt.iter	Integer, number of iterations during burn-in to adapt proposal standard deviations. Default is 500.
priors	Optional list specifying prior hyperparameters.
seed	Optional integer random seed for reproducibility.
optim.method	Character, optimization algorithm for MLE. Default is "BFGS".
optim.control	List of control parameters passed to optim .

Value

A list with specified control parameters.

Author(s)

Shikhar Tyagi, Arvind Pandey, Vrijesh Tripathi

Examples

```
ctrl <- corrFrailty.control(n.iter = 1000, n.burnin = 200, n.chains = 2)
```

cpo	<i>Conditional Predictive Ordinate (CPO) and Log Pseudo Marginal Likelihood (LPML)</i>
-----	--

Description

Conditional Predictive Ordinate (CPO) and Log Pseudo Marginal Likelihood (LPML)

Usage

```
cpo(object)
```

Arguments

object An object of class "frailty_bayes".

Value

A list containing vector CPO and scalar LPML.

crhr_baseline	<i>Baseline Cumulative Reversed Hazard Rate Function</i>
---------------	--

Description

Computes the cumulative reversed hazard rate (CRHR), defined as $R_0(t) = -\ln F_0(t)$.

Usage

```
crhr_baseline(
  t,
  dist = c("GE", "EG", "GIR", "MIW", "GLL-I", "GLL-II", "GIE", "Weibull", "Exponential",
    "Log-Logistic"),
  params
)
```

Arguments

t Numeric vector of positive failure times.

dist Character string specifying baseline distribution.

params Numeric named vector of baseline parameters.

Value

Numeric vector of cumulative reversed hazard rate values.

Examples

```
crhr_baseline(c(10, 20, 30), dist = "GE", params = c(zeta = 12.4, delta = 0.11))
```

criteria	<i>Model Selection Criteria and Bayes Factor Comparisons</i>
----------	--

Description

Computes AIC, BIC, Corrected AIC (AICc), Hannan-Quinn Criterion (HQIC), Deviance Information Criterion (DIC), and Bayes Factors for additive shared frailty models.

Usage

```
calc_model_criteria(  
  loglik,  
  k,  
  n,  
  deviance_samples = NULL,  
  deviance_at_mean = NULL  
)  
  
calc_bayes_factor(fit1, fit2)
```

Arguments

- loglik Scalar maximized or posterior mean log-likelihood value.
- k Number of estimated model parameters.
- n Sample size (number of subjects/pairs).
- deviance_samples Optional numeric vector of posterior deviances (-2 * logLik) for DIC calculation.
- deviance_at_mean Optional scalar deviance evaluated at the posterior mean of parameters.
- fit1, fit2 Fitted model objects of class "add_frailty".

Value

For `calc_model_criteria`, a named list of information criteria. For `calc_bayes_factor`, a scalar Bayes factor comparing `fit1` against `fit2`.

References

Pandey, A., Hanagal, D. D., and Tyagi, S. (2022). Generalised Lindley shared additive frailty regression model for bivariate survival data. *Statistics in Transition new series*, 23(4), 161-176. [doi:10.2478/stattrans20220048](https://doi.org/10.2478/stattrans20220048)

Tyagi, S., Pandey, A., Hanagal, D. D., and Chesneau, C. (2024). A study on comparisons of additive regression frailty models to counter heterogeneity: Bayesian strategies and case study. *Communications in Statistics - Simulation and Computation*. [doi:10.1080/03610918.2024.2391874](https://doi.org/10.1080/03610918.2024.2391874)

Examples

```
# Calculate information criteria
calc_model_criteria(loglik = -330.5, k = 13, n = 38)
```

cross_ratio_rhr

*Clayton-Oakes Cross-Ratio Function under Reversed Hazard Rate***Description**

Computes the bivariate Clayton-Oakes cross-ratio association measure $\kappa(t_1, t_2)$.

Usage

```
cross_ratio_rhr(
  t1,
  t2,
  x1 = NULL,
  x2 = NULL,
  beta = NULL,
  frailty = "WL",
  theta = 1,
  mu = NULL,
  theta2 = NULL,
  baseline = "GE",
  params1 = c(zeta = 1, delta = 1),
  params2 = c(zeta = 1, delta = 1)
)
```

Arguments

t1	Numeric scalar or vector of first component times.
t2	Numeric scalar or vector of second component times.
x1	Numeric matrix or vector of covariates for first component.
x2	Numeric matrix or vector of covariates for second component.
beta	Numeric vector of regression coefficients.
frailty	Character string indicating frailty distribution.
theta	Numeric parameter for frailty distribution.
mu	Optional secondary parameter for "GL".
theta2	Optional second parameter for "GIG".
baseline	Character string indicating baseline distribution.
params1	Numeric named vector of baseline parameters for first component.
params2	Numeric named vector of baseline parameters for second component.

Value

Numeric vector of cross-ratio values.

Examples

```
cross_ratio_rhr(15, 18, frailty = "WL", theta = 3.4, baseline = "GE",
  params1 = c(zeta = 12.4, delta = 0.11),
  params2 = c(zeta = 11.8, delta = 0.11))
```

d_rhr_baseline	<i>Baseline Probability Density Function under Reversed Hazard Rate</i>
----------------	---

Description

Computes the probability density function (PDF) for various baseline distributions used in reversed hazard rate frailty models.

Usage

```
d_rhr_baseline(
  t,
  dist = c("GE", "EG", "GIR", "MIW", "GLL-I", "GLL-II", "GIE", "Weibull", "Exponential",
    "Log-Logistic"),
  params
)
```

Arguments

- t Numeric vector of positive failure times.
- dist Character string specifying baseline distribution. One of "GE", "EG", "GIR", "MIW", "GLL-I", "GLL-II", "GIE", "Weibull", "Exponential", "Log-Logistic".
- params Numeric named vector of baseline parameters.

Value

Numeric vector of density values.

Examples

```
d_rhr_baseline(c(10, 20, 30), dist = "GE", params = c(zeta = 12.4, delta = 0.11))
d_rhr_baseline(c(10, 20, 30), dist = "EG", params = c(zeta = 9.1, delta = 0.10))
```

d2laplace_frailty

Second Derivative of Laplace Transform of Frailty Distributions

Description

Computes $L''_W(s) = d^2 L_W(s)/ds^2$. Note that $L''_W(s) > 0$.

Usage

```
d2laplace_frailty(
  s,
  frailty = c("WL", "GL", "GIG", "Gamma", "IG", "Lindley", "None"),
  theta,
  mu = NULL,
  theta2 = NULL
)
```

Arguments

s	Numeric vector of non-negative arguments.
frailty	Character string indicating the frailty distribution.
theta	Numeric parameter for frailty distribution.
mu	Optional secondary parameter for "GL".
theta2	Optional second parameter for "GIG".

Value

Numeric vector of second derivative values.

Examples

```
d2laplace_frailty(c(0.1, 0.5, 1.0), frailty = "WL", theta = 3.4)
```

dens_proper

Proper Conditional Density Function for Susceptible Group

Description

Computes $f_p^*(t)$ for $t > 0$ under Minimum, Maximum, and Random activation schemes with Gamma frailty.

Usage

```
dens_proper(
  t,
  theta,
  alpha,
  shape,
  scale,
  activation = c("random", "minimum", "maximum"),
  baseline = c("weibull", "gompertz", "exponential")
)
```

Arguments

t	Numeric vector of positive survival times.
theta	Numeric vector or scalar representing the latent Poisson parameter (> 0).
alpha	Numeric scalar representing Gamma frailty variance (≥ 0).
shape	Numeric vector or scalar representing baseline shape parameter(s) (> 0).
scale	Numeric vector or scalar representing baseline scale parameter(s) (> 0).
activation	Character string indicating the activation scheme: "random" (default), "minimum", or "maximum".
baseline	Character string indicating the baseline hazard distribution: "weibull" (default), "gompertz", or "exponential".

Value

Numeric vector of proper conditional density values.

dependence

Dependence Measures and Theoretical Frailty Variance

Description

Computes Kendall's tau (rank-based association) and theoretical frailty variance for additive shared frailty models under Generalized Lindley, Weighted Lindley, Gamma, and Inverse Gaussian frailty distributions.

Usage

```
kendall_tau(
  frailty_type = c("genlindley", "wlindley", "gamma", "invgaussian"),
  theta = NULL,
  mu = NULL,
  s_param = NULL
)
```

```
frailty_variance(  
  frailty_type = c("genlindley", "wlindley", "gamma", "invgaussian"),  
  theta = NULL,  
  mu = NULL,  
  s_param = NULL  
)
```

Arguments

frailty_type	Character string specifying the frailty distribution: "genlindley", "wlindley", "gamma", or "invgaussian".
theta	Positive parameter for Generalized Lindley, Gamma, or Inverse Gaussian frailty.
mu	Positive parameter for Generalized Lindley frailty (with $0 < \mu < 1 + \theta$).
s_param	Positive shape parameter for Weighted Lindley frailty.

Value

A scalar numeric value representing Kendall’s tau or theoretical frailty variance.

References

Pandey, A., Hanagal, D. D., and Tyagi, S. (2022). Generalised Lindley shared additive frailty regression model for bivariate survival data. *Statistics in Transition new series*, 23(4), 161-176. [doi:10.2478/stattrans20220048](https://doi.org/10.2478/stattrans20220048)

Tyagi, S., Pandey, A., Hanagal, D. D., and Chesneau, C. (2024). A study on comparisons of additive regression frailty models to counter heterogeneity: Bayesian strategies and case study. *Communications in Statistics - Simulation and Computation*. [doi:10.1080/03610918.2024.2391874](https://doi.org/10.1080/03610918.2024.2391874)

Examples

```
# Kendall's tau for Generalized Lindley frailty  
kendall_tau(frailty_type = "genlindley", theta = 2.29258, mu = 1.38391)  
  
# Kendall's tau for Weighted Lindley frailty  
kendall_tau(frailty_type = "wlindley", s_param = 1.0955)  
  
# Frailty variances  
frailty_variance(frailty_type = "genlindley", theta = 2.29, mu = 1.38)  
frailty_variance(frailty_type = "wlindley", s_param = 1.10)
```

diagnostics_mcmc	MCMC Convergence Diagnostics
------------------	------------------------------

Description

Computes Geweke z-score, Heidelberger-Welch stationarity test p-value, and effective sample size (ESS).

Usage

```
diagnostics_mcmc(mcmc_matrix, frac1 = 0.1, frac2 = 0.5)
```

Arguments

<code>mcmc_matrix</code>	Numeric matrix or data frame where each column represents posterior draws for a parameter.
<code>frac1</code>	Fraction of samples to use from the beginning for the Geweke test. Default is 0.10.
<code>frac2</code>	Fraction of samples to use from the end for the Geweke test. Default is 0.50.

Details

Evaluates MCMC chain convergence and stationarity for each parameter column in `mcmc_matrix`. The Geweke diagnostic compares sample means from the beginning (`frac1`) and end (`frac2`) of the chain. The Heidelberg-Welch test performs a Cramer-von Mises stationarity test on the cumulative sum of centered draws. Effective sample size (ESS) adjusts for chain autocorrelation. Missing values (NA or NaN) in input matrices are not permitted.

Value

A data frame with one row per parameter containing the following columns:

`Parameter` Name of the parameter.

`Geweke_Z` Geweke test z-score.

`Geweke_p` Two-sided p-value for the Geweke test (p-value > 0.05 indicates convergence).

`Heidelberg_p` Stationarity test p-value (p-value > 0.05 indicates stationary chain).

`ESS` Effective sample size adjusted for autocorrelation.

Examples

```
set.seed(123)
mat <- cbind(param1 = stats::rnorm(500), param2 = stats::rgamma(500, 2, 1))
diagnostics_mcmc(mat)
```

dic

Deviance Information Criterion (DIC)

Description

Calculates DIC for a Bayesian frailty model.

Usage

```
dic(object)
```

Arguments

object An object of class "frailty_bayes".

Value

A list with elements DIC, D_bar, D_hat, and p_D.

Examples

```
data(kidney_frailty)
fit <- frailty_bayes(Surv(time1, time2, status1, status2) ~ age,
                    data = kidney_frailty, n_iter = 200, burn_in = 50)
dic(fit)
```

dlaplace_frailty	<i>First Derivative of Laplace Transform of Frailty Distributions</i>
------------------	---

Description

Computes $L'_W(s) = dL_W(s)/ds$. Note that $L'_W(s) \leq 0$.

Usage

```
dlaplace_frailty(
  s,
  frailty = c("WL", "GL", "GIG", "Gamma", "IG", "Lindley", "None"),
  theta,
  mu = NULL,
  theta2 = NULL
)
```

Arguments

s Numeric vector of non-negative arguments.

frailty Character string indicating the frailty distribution.

theta Numeric parameter for frailty distribution.

mu Optional secondary parameter for "GL".

theta2 Optional second parameter for "GIG".

Value

Numeric vector of first derivative values.

Examples

```
dlaplace_frailty(c(0.1, 0.5, 1.0), frailty = "WL", theta = 3.4)
```

eaic	<i>Expected Akaike Information Criterion (EAIC)</i>
------	---

Description

Expected Akaike Information Criterion (EAIC)

Usage

```
eaic(object)
```

Arguments

object An object of class "frailty_bayes".

Value

Numeric EAIC value.

ebic	<i>Expected Bayesian Information Criterion (EBIC)</i>
------	---

Description

Expected Bayesian Information Criterion (EBIC)

Usage

```
ebic(object)
```

Arguments

object An object of class "frailty_bayes".

Value

Numeric EBIC value.

em_oneshot_frailty	<i>Expectation-Maximization Algorithm for One-Shot Device Testing Data</i>
--------------------	--

Description

Implements the EM algorithm developed by Yu, So, and Balakrishnan (2025) for computing maximum likelihood estimates in one-shot device testing under a shared gamma frailty model with Weibull or Exponential baseline lifetime distributions.

Usage

```
em_oneshot_frailty(
  count_mat,
  X_cov,
  tau,
  M,
  baseline = c("weibull", "exponential"),
  frailty = TRUE,
  start = NULL,
  max_iter = 500,
  tol = 1e-05,
  trace = FALSE
)
```

Arguments

count_mat	An $I \times 2^M$ matrix of observed counts for each malfunction subset.
X_cov	An $I \times p$ matrix of covariates with intercept.
tau	A positive numeric vector of length I containing inspection times.
M	An integer specifying the number of components ($M \geq 1$).
baseline	A character string: "weibull" (default) or "exponential".
frailty	A logical value: TRUE (default) for frailty model, FALSE for independence.
start	Optional numeric vector of initial parameter values.
max_iter	An integer specifying the maximum number of EM iterations (default 500).
tol	A positive numeric scalar for convergence tolerance (default 1e-5).
trace	A logical value: TRUE to print iteration progress, FALSE (default) to suppress.

Details

The EM algorithm handles the completely censored one-shot device test data by treating latent component failure times $T_{i,j,m}$ and latent frailty terms $\gamma_{i,j}$ as missing data.

In the ****E-step****, the conditional expectations $E[\gamma_{i,j}|X]$, $E[\ln \gamma_{i,j}|X]$, $E[\gamma_{i,j} T_{i,j,m}^{\alpha_{i,m}}|X]$, and $E[\ln T_{i,j,m}|X]$ are computed using the closed-form and incomplete beta expressions derived in Sections 1.4 and Appendix 1.B of the paper.

In the ****M-step****, parameters a_m , b_m , and β are updated to maximize the expected complete log-likelihood $Q(\theta|\theta^{(h)})$.

Missing values are not permitted in any input data.

Value

A list containing:

par Numeric vector of maximum likelihood parameter estimates.

loglik Maximized observed log-likelihood value.

iterations Number of EM iterations completed.

converged Logical flag indicating whether the algorithm converged (TRUE or FALSE).

history A data frame recording the log-likelihood and parameter norm across iterations.

References

Yu, C., So, H. Y., and Balakrishnan, N. (2025). *Inference for one-shot devices with Weibull component lifetimes under gamma frailty model*. In: Stochastic Modeling and Statistical Methods, Elsevier Inc., Chapter 1, pp. 1–34. doi:10.1016/B9780443316944.000062.

Examples

```
data(classH)
X_cov <- cbind(1, classH$stress)
tau <- classH$time
count_mat <- as.matrix(classH[, c("n_none", "n_T", "n_G", "n_both")])
fit_em <- em_oneshot_frailty(count_mat, X_cov, tau, M = 2, baseline = "weibull",
                           frailty = TRUE, max_iter = 25, tol = 1e-4)
fit_em$par
```

em_oneshot_kofm

Expectation-Maximization Algorithm for One-Shot Devices

Description

Implements the Expectation-Maximization (EM) algorithm developed by Ling et al. (2021) for determining the maximum likelihood estimates of model parameters for one-shot devices with multiple dependent failure modes under gamma frailty.

Usage

```
em_oneshot_kofm(
  count_mat,
  X_cov,
  tau,
  M,
  baseline = c("exponential", "weibull"),
```

```

    frailty = TRUE,
    start = NULL,
    max_iter = 500,
    tol = 1e-05,
    trace = FALSE
)

```

Arguments

count_mat	An $I \times 2^M$ matrix of observed counts for each malfunction subset.
X_cov	An $I \times p$ matrix of covariates with intercept.
tau	A positive numeric vector of length I containing inspection times.
M	An integer specifying the number of components ($M \geq 1$).
baseline	A character string: "exponential" (default) or "weibull".
frailty	A logical value: TRUE (default) for frailty model, FALSE for independence.
start	Optional numeric vector of initial parameter values.
max_iter	An integer specifying the maximum number of EM iterations (default 500).
tol	A positive numeric scalar for convergence tolerance (default 1e-5).
trace	A logical value: TRUE to print iteration progress, FALSE (default) to suppress.

Details

The EM algorithm treats latent component failure times $T_{i,j,m}$ and latent device frailties $\gamma_{i,j}$ as unobserved data.

In the ****E-step****, exact closed-form expressions for the conditional expectations $E[\gamma \mid X]$, $E[\ln \gamma \mid X]$, and $E[\gamma T_{i,m} \mid X]$ are computed via inclusion-exclusion formulas (Equations 12–15 of Ling et al., 2021).

In the ****M-step****, the $2M + 1$ parameter maximization problem decouples into M separate 2-parameter optimization subproblems for each failure mode (Equations 9–10) and one 1-parameter root-finding problem for $\beta \in (0, 0.5]$ (Equation 11), ensuring rapid convergence and numerical stability.

Missing values are not permitted.

Value

A list containing:

par Numeric vector of maximum likelihood parameter estimates.

loglik Maximized observed log-likelihood value.

iterations Number of EM iterations completed.

converged Logical flag indicating whether the algorithm converged (TRUE or FALSE).

history A data frame recording the log-likelihood and parameter norm across iterations.

References

Ling, M.-H., Balakrishnan, N., Yu, C., and So, H. Y. (2021). Inference for One-Shot Devices with Dependent k-Out-of-M Structured Components under Gamma Frailty. *Mathematics*, 9(23), 3032. [doi:10.3390/math9233032](https://doi.org/10.3390/math9233032).

Examples

```
data(classH)
X_cov <- cbind(1, classH$stress)
tau <- classH$time
count_mat <- as.matrix(classH[, c("n_none", "n_T", "n_G", "n_both")])
fit_em <- em_oneshot_kofm(count_mat, X_cov, tau, M = 2, baseline = "exponential",
                        frailty = TRUE, max_iter = 50, tol = 1e-4)
fit_em$par
```

evaluate_ziprom	<i>Evaluation Metrics and Residual Diagnostics for Zero-Inflated Promotion Models</i>
-----------------	---

Description

Computes statistical goodness-of-fit and forecast accuracy measures including Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), Brier Score, and Harrell’s Concordance Index (C-index), as well as exact normalized randomized quantile residuals.

Usage

```
evaluate_ziprom(object, newdata = NULL, times = NULL, ...)

ziprom_residuals(object, ...)
```

Arguments

- object An object of class "ziprom_fit".
- newdata Optional new data frame to evaluate model performance on out-of-sample data.
- times Optional numeric evaluation times for Brier Score / MSE.
- ... Additional arguments.

Value

- For evaluate_ziprom: A list containing:
- MSE Mean Squared Error between observed status and fitted survival probabilities.
 - RMSE Root Mean Squared Error ($\sqrt{\text{MSE}}$).
 - MAE Mean Absolute Error.
 - Brier_Score Integrated / average Brier score across time horizon.

C_index Harrell's Concordance Index (discrimination metric).

LogLik Maximum Log-Likelihood value.

AIC Akaike Information Criterion.

BIC Bayesian Information Criterion.

CAIC Consistent Akaike Information Criterion.

HQIC Hannan-Quinn Information Criterion.

For ziprom_residuals: A numeric vector of normalized randomized quantile residuals.

References

Dunn, P. K., and Smyth, G. K. (1996). Randomized Quantile Residuals. *Journal of Computational and Graphical Statistics*, 5(3), 236-244. doi:10.1080/10618600.1996.10474708

Assuncao, D., Ramos, P. L., Tomazella, V., Leao, J., Osses, L., and Rodrigues, A. (2026). Zero-Inflated Promotion Cure-Rate Models With Frailty and Flexible Activation Schemes. *Biometrical Journal*, 68:e70162. doi:10.1002/bimj.70162

Examples

```
set.seed(42)
sim_dat <- sim_ziprom(n = 80, activation = "random")
fit <- ziprom_fit(survival::Surv(time, status) ~ x, data = sim_dat, activation = "random")
evals <- evaluate_ziprom(fit)
print(evals)
```

fb_sim_frailty

Simulate Survival Data with Shared or Univariate Frailty

Description

Generates synthetic survival datasets under any supported baseline hazard and frailty distribution with right censoring.

Usage

```
fb_sim_frailty(
  n = 100,
  baseline = "weibull",
  frailty = "gamma",
  type = "multiplicative_bivariate",
  beta = c(0.5, -0.5),
  theta = 1,
  baseline_params = c(0.01, 1.2),
  censor_rate = 0.2
)
```

Arguments

<code>n</code>	Sample size (number of subjects/clusters).
<code>baseline</code>	Baseline hazard name (default "weibull").
<code>frailty</code>	Frailty distribution name (default "gamma").
<code>type</code>	Model type (default "multiplicative_bivariate").
<code>beta</code>	Regression coefficients vector.
<code>theta</code>	Frailty parameter(s).
<code>baseline_params</code>	Baseline parameters vector.
<code>censor_rate</code>	Right-censoring probability.

Value

A data frame containing simulated survival times, status indicators, covariates, and true frailty values.

Examples

```
sim_data <- sim_frailty(
  n = 50,
  baseline = "weibull",
  frailty = "gamma",
  type = "multiplicative_bivariate",
  beta = c(0.5, -0.5),
  theta = 1.5,
  baseline_params = c(0.02, 1.2)
)
head(sim_data)
```

<code>fisher_info_oneshot</code>	<i>Observed Fisher Information Matrix for One-Shot Devices</i>
----------------------------------	--

Description

Computes the observed Fisher information matrix for model parameters in one-shot device testing based on the missing information principle and exact analytical derivatives derived in Ling et al. (2021).

Usage

```
fisher_info_oneshot(
  par,
  count_mat,
  X_cov,
  tau,
  M,
```

```

baseline = c("weibull", "exponential"),
frailty = TRUE,
powerset = NULL
)

```

Arguments

par	Numeric vector of estimated model parameters.
count_mat	An $I \times 2^M$ matrix of observed device counts.
X_cov	An $I \times p$ matrix of covariates.
tau	A positive numeric vector of inspection times.
M	An integer specifying the number of components.
baseline	A character string: "exponential" (default) or "weibull".
frailty	A logical value: TRUE (default) for frailty model, FALSE for independence.
powerset	An optional list of power set subsets.

Details

The missing information principle gives the observed Fisher information matrix as:

$$I(\hat{\theta}) = \sum_{i=1}^I \sum_{X \in P(\Omega)} \frac{K_i}{P_i(X, \tau_i)} \left(\frac{\partial P_i(X, \tau_i)}{\partial \theta} \right) \left(\frac{\partial P_i(X, \tau_i)}{\partial \theta} \right)^T$$

Missing values are not permitted.

Value

A symmetric positive semi-definite numeric matrix representing the observed Fisher information matrix.

References

Ling, M.-H., Balakrishnan, N., Yu, C., and So, H. Y. (2021). Inference for One-Shot Devices with Dependent k-Out-of-M Structured Components under Gamma Frailty. *Mathematics*, 9(23), 3032. [doi:10.3390/math9233032](https://doi.org/10.3390/math9233032).

Examples

```

data(classH)
X_cov <- cbind(1, classH$stress)
tau <- classH$time
count_mat <- as.matrix(classH[, c("n_none", "n_T", "n_G", "n_both")])
par <- c(-4.9897, -0.0058, -16.4365, 0.0183, 0.5)
fisher_info_oneshot(par, count_mat, X_cov, tau, M = 2, baseline = "exponential", frailty = TRUE)

```

fit_add_frailty_bayes *Bayesian Estimation for Additive Shared Frailty Models via MCMC*

Description

Fits additive shared frailty regression models for bivariate survival data using an adaptive hybrid Metropolis-Hastings Markov Chain Monte Carlo (MCMC) sampler with multiple chains, informative or flat priors, and point estimation under Squared Error, Modified Quadratic Squared Error, and Precautionary loss functions.

Usage

```
fit_add_frailty_bayes(
  formula = NULL,
  data,
  t1_col = "t1",
  delta1_col = "delta1",
  t2_col = "t2",
  delta2_col = "delta2",
  frailty_type = c("genlindley", "wlindley", "gamma", "invgaussian"),
  baseline_dist = c("gw", "gll2"),
  n_iter = 10000,
  burn_in = 2000,
  thin = 5,
  n_chains = 2,
  prior = c("flat", "informative"),
  prior_params = NULL,
  init = NULL,
  seed = NULL,
  verbose = FALSE
)
```

Arguments

formula	Model formula for covariates (e.g. $\sim \text{age} + \text{sex} + \text{gn} + \text{an} + \text{pkd}$). If NULL, an intercept-only or single-covariate model is fitted from the data.
data	A data frame containing survival times, censoring indicators, and covariates.
t1_col	Character string naming the first survival time column in data.
delta1_col	Character string naming the first censoring indicator column in data.
t2_col	Character string naming the second survival time column in data.
delta2_col	Character string naming the second censoring indicator column in data.
frailty_type	Character string specifying the frailty distribution: "genlindley" (Generalized Lindley), "wlindley" (Weighted Lindley), "gamma" (Gamma), or "invgaussian" (Inverse Gaussian).

baseline_dist	Character string specifying the baseline distribution: "gw" (Generalized Weibull) or "gll2" (Generalized Log-Logistic Type-II).
n_iter	Total number of MCMC iterations per chain (default 10000).
burn_in	Number of initial burn-in iterations to discard (default 2000).
thin	Thinning interval / autocorrelation lag (default 5).
n_chains	Number of parallel/independent Markov chains to run (default 2).
prior	Character string specifying prior distribution: "flat" (non-informative) or "informative".
prior_params	Optional list of hyperparameters for informative priors.
init	Optional list or vector of initial parameter values.
seed	Optional integer seed for reproducibility.
verbose	Logical; if TRUE, prints MCMC chain progress.

Value

An object of class `c("add_frailty_bayes", "add_frailty")` containing:

estimates Data frame of Bayes point estimates (SELF, MQSELF, PLF), risks, and credible intervals.

chains List of thinned posterior sample matrices for each chain.

posterior_samples Combined posterior sample matrix across all chains.

deviance_samples Posterior deviance samples ($-2 * \log\text{Lik}$).

diagnostics List containing Geweke, Gelman-Rubin, and Heidelberger-Welch convergence statistics.

criteria Model selection criteria (AIC, BIC, AICc, HQIC, DIC, LogLik).

kendall_tau Kendall's tau dependence measure evaluated at the Bayes estimate.

frailty_variance Theoretical frailty variance at the Bayes estimate.

model_info List containing model settings, formula, call, and data dimensions.

References

Pandey, A., Hanagal, D. D., and Tyagi, S. (2022). Generalised Lindley shared additive frailty regression model for bivariate survival data. *Statistics in Transition new series*, 23(4), 161-176. [doi:10.2478/stattrans20220048](https://doi.org/10.2478/stattrans20220048)

Tyagi, S., Pandey, A., Hanagal, D. D., and Chesneau, C. (2024). A study on comparisons of additive regression frailty models to counter heterogeneity: Bayesian strategies and case study. *Communications in Statistics - Simulation and Computation*. [doi:10.1080/03610918.2024.2391874](https://doi.org/10.1080/03610918.2024.2391874)

Examples

```
data(kidney_bivariate)
sub_data <- kidney_bivariate[1:20, ]
fit_bayes <- fit_add_frailty_bayes(
  formula = ~ age,
  data = sub_data,
```

```

    frailty_type = "wlindley", baseline_dist = "gw",
    n_iter = 200, burn_in = 50, thin = 2, n_chains = 2,
    seed = 123
  )
  print(fit_bayes)

```

fit_add_frailty_mle	<i>Maximum Likelihood Estimation for Additive Shared Frailty Models</i>
---------------------	---

Description

Estimates unknown parameters of additive shared frailty models under bivariate random censoring using numerical optimization of the log-likelihood with transformation for bounded parameters and Hessian-based standard errors.

Usage

```

fit_add_frailty_mle(
  formula = NULL,
  data,
  t1_col = "t1",
  delta1_col = "delta1",
  t2_col = "t2",
  delta2_col = "delta2",
  frailty_type = c("genlindley", "wlindley", "gamma", "invgaussian"),
  baseline_dist = c("gw", "gll2"),
  init = NULL,
  method = "Nelder-Mead",
  hessian = TRUE,
  control = list(maxit = 5000)
)

```

Arguments

formula	Model formula for covariates (e.g. $\sim \text{age} + \text{sex} + \text{gn} + \text{an} + \text{pkd}$). If NULL, an intercept-only or single-covariate model is fitted.
data	A data frame containing survival times, censoring indicators, and covariates.
t1_col	Character string naming the first survival time column in data.
delta1_col	Character string naming the first censoring indicator column in data.
t2_col	Character string naming the second survival time column in data.
delta2_col	Character string naming the second censoring indicator column in data.
frailty_type	Character string specifying the frailty distribution: "genlindley" (Generalized Lindley), "wlindley" (Weighted Lindley), "gamma" (Gamma), or "invgaussian" (Inverse Gaussian).

baseline_dist	Character string specifying the baseline distribution: "gw" (Generalized Weibull) or "gll2" (Generalized Log-Logistic Type-II).
init	Optional numeric vector of initial parameter values.
method	Optimization method passed to stats::optim (default "Nelder-Mead").
hessian	Logical; if TRUE, computes the numerical Hessian for standard errors.
control	List of control parameters passed to stats::optim.

Value

An object of class c("add_frailty_mle", "add_frailty") containing:

coefficients Named vector of Maximum Likelihood parameter estimates.

estimates Data frame containing estimates, standard errors, z-values, p-values, and 95% confidence bounds.

vcov Asymptotic covariance matrix of the estimates.

loglik Maximized log-likelihood value.

criteria Named list of AIC, BIC, AICc, HQIC values.

kendall_tau Kendall's tau dependence measure evaluated at the MLE.

frailty_variance Theoretical frailty variance evaluated at the MLE.

convergence Convergence code from stats::optim.

model_info List containing model specifications and call.

References

Pandey, A., Hanagal, D. D., and Tyagi, S. (2022). Generalised Lindley shared additive frailty regression model for bivariate survival data. *Statistics in Transition new series*, 23(4), 161-176. doi:10.2478/stattrans20220048

Tyagi, S., Pandey, A., Hanagal, D. D., and Chesneau, C. (2024). A study on comparisons of additive regression frailty models to counter heterogeneity: Bayesian strategies and case study. *Communications in Statistics - Simulation and Computation*. doi:10.1080/03610918.2024.2391874

Examples

```
data(kidney_bivariate)
sub_data <- kidney_bivariate[1:20, ]
fit_mle <- fit_add_frailty_mle(
  formula = ~ age,
  data = sub_data,
  frailty_type = "wlindley", baseline_dist = "gw",
  control = list(maxit = 100)
)
print(fit_mle)
```

fit_rhr_frailty_bayes	<i>Bayesian Estimation of Shared Frailty Models under Reversed Hazard Rate</i>
-----------------------	--

Description

Fits shared frailty regression models for bivariate left-censored survival data based on reversed hazard rates using an adaptive Metropolis-within-Gibbs MCMC algorithm. Provides posterior estimates under squared error (SELF), entropy (ELF), and precautionary (PLF) loss functions, highest posterior density (HPD) credible intervals, and model comparison criteria.

Usage

```
fit_rhr_frailty_bayes(
  formula = NULL,
  data = NULL,
  t1 = NULL,
  t2 = NULL,
  d1 = NULL,
  d2 = NULL,
  covars = NULL,
  frailty = c("WL", "GL", "GIG", "Gamma", "IG", "Lindley", "None"),
  baseline = c("GE", "EG", "GIR", "MIW", "GLL-I", "GLL-II", "GIE", "Weibull",
    "Exponential", "Log-Logistic"),
  n_iter = 2000,
  burn_in = 500,
  thin = 1,
  n_chains = 1,
  prior = c("flat", "informative"),
  prior_hyper = NULL,
  loss_functions = c("SELF", "ELF", "PLF"),
  init = NULL,
  tune_sd = NULL,
  verbose = FALSE,
  seed = NULL
)
```

Arguments

formula	An optional object of class "formula" of the form <code>cbind(t1, t2, d1, d2) ~ x1 + x2</code> .
data	An optional data frame containing the variables in the model.
t1	Numeric vector of first failure/censoring times (if formula is not provided). Must be positive.
t2	Numeric vector of second failure/censoring times. Must be positive.

d1	Numeric or integer vector of censoring indicators for first component (1 = observed, 0 = left-censored).
d2	Numeric or integer vector of censoring indicators for second component (1 = observed, 0 = left-censored).
covars	Optional numeric matrix or data frame of covariates.
frailty	Character string specifying frailty distribution. One of "WL" (Weighted Lindley), "GL" (Generalized Lindley), "GIG" (Generalized Inverse Gaussian mixture), "Gamma" (Gamma), "IG" (Inverse Gaussian), "Lindley" (Lindley), or "None" (Independent baseline). Default is "WL".
baseline	Character string specifying baseline distribution. One of "GE" (Generalized Exponential), "EG" (Exponentiated Gumbel), "GIR" (Generalized Inverse Rayleigh), "MIW" (Modified Inverse Weibull), "GLL-I" (Generalized Log-Logistic Type-I), "GLL-II" (Generalized Log-Logistic Type-II), "GIE" (Generalized Inverted Exponential), "Weibull", "Exponential", or "Log-Logistic". Default is "GE".
n_iter	Total number of MCMC iterations. Default is 2000.
burn_in	Number of initial burn-in iterations to discard. Default is 500.
thin	Thinning interval. Default is 1.
n_chains	Number of parallel MCMC chains. Default is 1.
prior	Character string indicating prior specification: "flat" (non-informative) or "informative". Default is "flat".
prior_hyper	Optional named list of hyperparameters for informative gamma and normal priors.
loss_functions	Character vector of loss functions under which Bayes estimates are computed. Default is c("SELF", "ELF", "PLF").
init	Optional numeric vector of initial values for the parameters.
tune_sd	Optional numeric vector of proposal standard deviations for random-walk Metropolis updates.
verbose	Logical; if TRUE, progress messages are printed. Default is FALSE.
seed	Optional integer seed for reproducibility. Default is NULL.

Details

The model assumes that the bivariate lifetime random variables (T_{1j}, T_{2j}) share an unobserved frailty variable W_j that acts multiplicatively on the baseline reversed hazard rates:

$$r(t_{ij} \mid W_j, X_{ij}) = W_j r_{0i}(t_{ij}) \exp(X_{ij}^T \beta)$$

Under left censoring, observations with $d_{ij} = 0$ indicate that the event occurred before or at the observation time t_{ij} ($T_{ij} \leq t_{ij}$), while $d_{ij} = 1$ indicates an exact observed failure time. Missing values (NA or NaN) in survival times or censoring indicators are not allowed and will produce an informative error.

Value

An object of class "rhr_bayes_fit" with the following components:

summary A data frame summarizing parameter estimates under SELF, ELF, and PLF loss functions, posterior standard deviations, Monte Carlo standard errors (MCSE), medians, 95% HPD credible intervals, Geweke test p-values, and Heidelberger-Welch stationarity test p-values.

draws A numeric matrix of kept posterior parameter draws of dimension $((n_iter - burn_in) / thin) * n_chains$ by n_params .

chains A list of chains containing kept parameter draws, log-likelihoods, and inverse pointwise likelihoods.

estimates A named list containing numeric vectors of Bayes point estimates under SELF (Posterior Mean), ELF (Harmonic Mean), and PLF (Root Mean Square).

loss_functions Character vector of loss functions evaluated.

diagnostics A data frame containing MCMC convergence diagnostics including Geweke z-scores and p-values, Heidelberger-Welch p-values, and effective sample sizes (ESS).

criteria A list of model fit and selection criteria: AIC, BIC, AICc, HQIC, DIC (Deviance Information Criterion), pD (effective number of parameters), EAIC, EBIC, LPML (Log Pseudo Marginal Likelihood), mean_deviance, and dev_at_mean.

cpo Numeric vector of conditional predictive ordinate (CPO) values for each observation pair.

kendall_tau Numeric scalar representing estimated Kendall's rank correlation coefficient τ .

frailty_variance Numeric scalar representing estimated variance of the frailty variable $V(W)$.

model_info A list containing model specifications: frailty distribution, baseline distribution, sample size n , number of iterations, burn-in, thinning, number of chains, prior type, covariate names, and parameter names.

data A list containing the input data vectors and covariate matrix.

call The matched call.

Examples

```
data(austrian_twin)
# Demonstration with a small subset and few iterations
fit <- fit_rhr_frailty_bayes(
  cbind(t1, t2, d1, d2) ~ age + k2 + k3,
  data = austrian_twin[1:50, ],
  frailty = "WL", baseline = "GE",
  n_iter = 60, burn_in = 10, thin = 1
)
print(fit)
summary(fit)
```

fit_rhr_frailty_mle *Maximum Likelihood Estimation for Shared RHR Frailty Models*

Description

Fits shared frailty regression models for bivariate left-censored survival data based on reversed hazard rates using maximum likelihood estimation.

Usage

```
fit_rhr_frailty_mle(
  formula = NULL,
  data = NULL,
  t1 = NULL,
  t2 = NULL,
  d1 = NULL,
  d2 = NULL,
  covars = NULL,
  frailty = c("WL", "GL", "GIG", "Gamma", "IG", "Lindley", "None"),
  baseline = c("GE", "EG", "GIR", "MIW", "GLL-I", "GLL-II", "GIE", "Weibull",
    "Exponential", "Log-Logistic"),
  init = NULL,
  hessian = TRUE,
  control = list()
)
```

Arguments

formula	An optional object of class "formula" of the form <code>cbind(t1, t2, d1, d2) ~ x1 + x2</code> .
data	An optional data frame containing variables.
t1	Numeric vector of first failure/censoring times. Must be positive.
t2	Numeric vector of second failure/censoring times. Must be positive.
d1	Numeric or integer vector of censoring indicators for first component (1 = observed, 0 = left-censored).
d2	Numeric or integer vector of censoring indicators for second component (1 = observed, 0 = left-censored).
covars	Optional numeric matrix of covariates.
frailty	Character string specifying frailty distribution. Default is "WL".
baseline	Character string specifying baseline distribution. Default is "GE".
init	Optional numeric vector of initial parameters.
hessian	Logical indicating whether to compute observed Fisher information Hessian. Default is TRUE.
control	Optional list of control parameters passed to optim .

Details

Maximizes the four-part left-censored log-likelihood function using the BFGS algorithm on unconstrained log-transformed parameter space to guarantee positive parameters. Standard errors are computed from the numerical Hessian using the multivariate Delta Method. Missing values (NA or NaN) in survival times or censoring indicators are not permitted.

Value

An object of class "rhr_mle_fit" with the following components:

`summary` A data frame containing parameter estimates, standard errors, z-scores, p-values, and Wald 95% confidence intervals.

`coefficients` A named numeric vector of parameter estimates.

`se` A named numeric vector of asymptotic standard errors computed via the Delta Method.

`vcov` The estimated asymptotic variance-covariance matrix.

`loglik` The maximized log-likelihood value.

`criteria` A list containing AIC, BIC, AICc, and HQIC information criteria.

`kendall_tau` Numeric scalar representing estimated Kendall's tau.

`frailty_variance` Numeric scalar representing estimated variance of the frailty variable.

`opt` The raw optimization result returned by [optim](#).

`model_info` A list containing model specifications: frailty, baseline, sample size, number of covariates, covariate names, and parameter names.

`data` A list containing the input data vectors and covariate matrix.

`call` The matched call.

Examples

```
data(austrian_twin)
fit <- fit_rhr_frailty_mle(
  cbind(t1, t2, d1, d2) ~ age + k2 + k3,
  data = austrian_twin[1:50, ],
  frailty = "WL", baseline = "GE"
)
print(fit)
summary(fit)
```

fitted.oneshot_kofm	<i>Extract Fitted Values</i>
---------------------	------------------------------

Description

Extracts fitted cell probabilities or expected counts from a "oneshot_kofm" object.

Usage

```
## S3 method for class 'oneshot_kofm'
fitted(object, type = c("probs", "counts"), ...)
```

Arguments

object	A fitted model object of class "oneshot_kofm".
type	A character string: "probs" (default) or "counts".
...	Additional arguments.

Value

A numeric matrix of fitted probabilities or expected counts.

forecast_ziprom	<i>Multi-Level Significance Survival Forecasting</i>
-----------------	--

Description

Helper function for extracting survival forecasts with 90%, 95%, and 99% level of significance intervals.

Usage

```
forecast_ziprom(
  object,
  newdata = NULL,
  times = NULL,
  level = c(0.9, 0.95, 0.99),
  ...
)
```

Arguments

object	An object of class "ziprom_fit".
newdata	Optional data frame containing covariate values for forecasting.
times	Optional numeric vector of forecast time points.
level	Numeric vector of confidence levels. Default is c(0.90, 0.95, 0.99).
...	Additional arguments passed to predict.ziprom_fit .

Value

A data frame of survival forecast probabilities with standard errors and multi-level significance intervals.

Examples

```
set.seed(123)
dat <- sim_ziprom(n = 60, activation = "random")
fit <- ziprom_fit(survival::Surv(time, status) ~ x, data = dat, activation = "random")
fc_res <- forecast_ziprom(fit, times = c(0.5, 1.0, 2.0, 3.0, 5.0))
print(fc_res)
```

frailty_bayes	<i>Bayesian Estimation of Shared, Additive, and Multiplicative Frailty Models</i>
---------------	---

Description

Fits parametric shared multiplicative, shared additive, univariate multiplicative, univariate additive, and merged frailty survival models using adaptive Markov Chain Monte Carlo (MCMC) sampling.

Usage

```
frailty_bayes(
  formula,
  data,
  baseline = c("weibull", "exponential", "genexponential", "exppower", "lfr",
    "genloglogistic", "genweibull", "genrayleigh", "weightedexp", "extendedweibull"),
  frailty = c("gamma", "invgauss", "posstable", "compoundpoisson", "compoundnb",
    "genlindley", "weightedlindley", "mixture_invgauss"),
  type = c("multiplicative_bivariate", "additive_bivariate", "multiplicative_univariate",
    "additive_univariate", "merged_univariate"),
  n_iter = 3000,
  burn_in = 1000,
  thin = 1,
  priors = NULL,
  init = NULL,
  adapt_freq = 50,
  verbose = FALSE,
  ...
)
```

Arguments

formula	A formula expression, e.g. <code>Surv(time1, time2, status1, status2) ~ x1 + x2</code> for bivariate data, or <code>Surv(time, status) ~ x1 + x2</code> for univariate data.
data	A <code>data.frame</code> containing all survival variables and covariates.
baseline	Character specifying baseline distribution: "weibull" (default), "exponential", "genexponential", "exppower", "lfr", "genloglogistic", "genweibull", "genrayleigh", "weightedexp", or "extendedweibull".

frailty	Character specifying frailty distribution: "gamma" (default), "invgauss", "posstable", "compoundpoisson", "compoundnb", "genlindley", "weightedlindley", or "mixture_invgauss".
type	Model formulation: "multiplicative_bivariate" (default), "additive_bivariate", "multiplicative_univariate", "additive_univariate", or "merged_univariate".
n_iter	Number of MCMC iterations saved post burn-in (default 3000).
burn_in	Number of burn-in iterations (default 1000).
thin	Thinning interval (default 1).
priors	List of prior distribution specifications (optional).
init	Vector of initial parameter values (optional).
adapt_freq	Frequency of proposal covariance matrix adaptation (default 50).
verbose	Logical; if TRUE, displays progress.
...	Additional arguments.

Value

An S3 object of class "frailty_bayes" containing:

chain	Matrix of posterior draws for all parameters.
summary	Data frame of posterior summaries, credible intervals, ESS, MCSE, and Geweke diagnostics.
diagnostics	List containing DIC, EAIC, EBIC, CPO, and LPML model comparison criteria.
log_lik_matrix	Matrix of individual log-likelihoods across iterations.
accept_rate	MCMC acceptance rate.
data	Processed data list.
call	Matched call.
baseline	Baseline distribution name.
frailty	Frailty distribution name.
type	Model type.

References

- Hanagal, D. D. and Dabade, A. D. (2013). Compound negative binomial shared frailty models for bivariate survival data. *Statistics and Probability Letters*, 83(11): 2507-2515. doi:[10.1016/j.spl.2013.07.015](https://doi.org/10.1016/j.spl.2013.07.015)
- Tyagi, S., Pandey, A., Agiwal, V. and Chesneau, C. (2021). Weighted Lindley multiplicative regression frailty models under random censored data. *Computational and Applied Mathematics*, 40: 265. doi:[10.1007/s40314021016665](https://doi.org/10.1007/s40314021016665)
- Kiprotich, G., Tyagi, S. and Ramos, P. L. (2025). Bayesian survival modeling with mixtures of inverse Gaussian frailties. *Journal of Applied Statistics*. doi:[10.1080/02664763.2025.2560510](https://doi.org/10.1080/02664763.2025.2560510)

Examples

```
data(kidney_frailty)
fit <- frailty_bayes(
  Surv(time1, time2, status1, status2) ~ age + sex,
  data = kidney_frailty,
  baseline = "weibull",
  frailty = "gamma",
  type = "multiplicative_bivariate",
  n_iter = 500,
  burn_in = 100
)
print(fit)
summary(fit)
```

frailty_compare	<i>Model Comparison Table for Frailty Models</i>
-----------------	--

Description

Creates a unified comparison table ranking multiple fitted frailty models by DIC, EAIC, EBIC, LPML, AIC, and BIC.

Usage

```
frailty_compare(..., names = NULL)
```

Arguments

...	One or more fitted objects of class "frailty_bayes" or "frailty_mle".
names	Optional character vector of model names.

Value

A data frame summarizing model comparison criteria and ranks.

Examples

```
data(kidney_frailty)
m1 <- frailty_bayes(Surv(time1, time2, status1, status2) ~ age,
  data = kidney_frailty, frailty = "gamma", n_iter = 200, burn_in = 50)
m2 <- frailty_bayes(Surv(time1, time2, status1, status2) ~ age,
  data = kidney_frailty, frailty = "invgauss", n_iter = 200, burn_in = 50)
frailty_compare(m1, m2, names = c("Gamma", "InvGauss"))
```

frailty_fit

Unified Formula Interface for Frailty Models

Description

Fits frailty survival regression models using either Bayesian MCMC or Maximum Likelihood Estimation.

Usage

```
frailty_fit(formula, data, engine = c("bayes", "mle"), ...)
```

Arguments

formula	Formula expression for bivariate or univariate survival data.
data	Data frame.
engine	Estimation engine: "bayes" (default) or "mle".
...	Arguments passed to frailty_bayes or frailty_mle .

Value

A fitted model object of class "frailty_bayes" or "frailty_mle".

Examples

```
data(kidney_frailty)
fit <- frailty_fit(
  Surv(time1, time2, status1, status2) ~ age + sex,
  data = kidney_frailty,
  engine = "mle",
  baseline = "weibull",
  frailty = "gamma"
)
print(fit)
```

frailty_mle

Maximum Likelihood Estimation of Shared Frailty Models

Description

Estimates shared, additive, multiplicative, and merged frailty models using maximum likelihood estimation with analytical/numerical Hessian.

Usage

```
frailty_mle(
  formula,
  data,
  baseline = c("weibull", "exponential", "genexponential", "exppower", "lfr",
    "genloglogistic", "genweibull", "genrayleigh", "weightedexp", "extendedweibull"),
  frailty = c("gamma", "invgauss", "posstable", "compoundpoisson", "compoundnb",
    "genlindley", "weightedlindley", "mixture_invgauss"),
  type = c("multiplicative_bivariate", "additive_bivariate", "multiplicative_univariate",
    "additive_univariate", "merged_univariate"),
  init = NULL,
  method = "BFGS",
  hessian = TRUE,
  ...
)
```

Arguments

formula	Formula expression for bivariate or univariate survival data.
data	Data frame containing survival data and covariates.
baseline	Baseline distribution name.
frailty	Frailty distribution name.
type	Model formulation type.
init	Initial parameter vector (optional).
method	Optimization method passed to <code>stats::optim</code> (default "BFGS").
hessian	Logical; if TRUE, computes the Hessian matrix.
...	Additional arguments.

Value

An S3 object of class "frailty_mle".

Examples

```
data(kidney_frailty)
fit_mle <- frailty_mle(
  Surv(time1, time2, status1, status2) ~ age + sex,
  data = kidney_frailty,
  baseline = "weibull",
  frailty = "gamma",
  type = "multiplicative_bivariate"
)
print(fit_mle)
summary(fit_mle)
```

generate_powerset	<i>Generate Power Set of Components</i>
-------------------	---

Description

Generates the power set $P(\Omega)$ of components $\Omega = \{1, \dots, M\}$, representing all 2^M possible combinations of malfunctioned components.

Usage

```
generate_powerset(M)
```

Arguments

M	An integer greater than or equal to 1 specifying the number of components.
---	--

Details

Missing values are not allowed for M. If M is not a positive integer, an error is thrown.

Value

A list of 2^M integer vectors, where each vector represents a subset of malfunctioned component indices (from `integer(0)` for no failures to `1:M` for all failures).

References

Ling, M.-H., Balakrishnan, N., Yu, C., and So, H. Y. (2021). Inference for One-Shot Devices with Dependent k-Out-of-M Structured Components under Gamma Frailty. *Mathematics*, 9(23), 3032. [doi:10.3390/math9233032](https://doi.org/10.3390/math9233032).

Examples

```
generate_powerset(2)
generate_powerset(3)
```

hpd_interval	<i>Highest Posterior Density (HPD) Credible Interval</i>
--------------	--

Description

Computes the highest posterior density credible interval for a sample of MCMC draws.

Usage

```
hpd_interval(samples, prob = 0.95)
```

Arguments

`samples` Numeric vector of MCMC posterior draws.

`prob` Numeric scalar in $(0, 1)$ specifying the coverage probability. Default is 0.95.

Details

Finds the shortest interval containing the specified probability mass `prob` from the sorted posterior draws. Missing values (NA or NaN) in `samples` are automatically omitted.

Value

A named numeric vector of length 2 containing:

`lower` The lower bound of the highest posterior density interval.

`upper` The upper bound of the highest posterior density interval.

Examples

```
set.seed(123)
draws <- stats::rnorm(1000, mean = 5, sd = 1)
hpd_interval(draws, prob = 0.95)
```

insulin_gdm

Time to Insulin Initiation in Gestational Diabetes Mellitus Dataset

Description

A real-world dataset comprising retrospective times to insulin initiation for 390 pregnant women diagnosed with gestational diabetes mellitus (GDM) before 24 weeks of pregnancy, followed at the Hospital das Clinicas, University of Sao Paulo, Brazil.

Usage

```
insulin_gdm
```

Format

A data frame with 390 rows and 8 variables:

`time` Time in weeks from the first obstetric visit to insulin initiation. Women initiating insulin within the first 7 days have time recorded as 0.

`status` Event indicator: 1 = insulin initiated during pregnancy, 0 = right-censored (cured/insulin-free throughout pregnancy).

`family_diabetes` Family history of diabetes: 0 = No (n = 143), 1 = Yes (n = 247).

`prior_gdm` Prior history of gestational diabetes: 0 = Primigravida (n = 94), 1 = No prior GDM (n = 249), 2 = Yes (n = 47).

smoking Smoking history: 0 = Non-smoker (n = 356), 1 = Smoker (n = 34).
 prior_macrosomia Prior fetal macrosomia: 0 = No (n = 356), 1 = Yes (n = 34).
 hypertension Prior chronic hypertension: 0 = No (n = 282), 1 = Yes (n = 108).
 bmi_cat Pre-pregnancy body mass index category: 0 = Normal (BMI <= 25, n = 84), 1 = Overweight (25 < BMI <= 30, n = 120), 2 = Obesity (BMI > 30, n = 186).

Source

Assuncao, D., Ramos, P. L., Tomazella, V., Leao, J., Osses, L., and Rodrigues, A. (2026). Zero-Inflated Promotion Cure-Rate Models With Frailty and Flexible Activation Schemes. *Biometrical Journal*, 68:e70162. doi:10.1002/bimj.70162

Calsavara, V. F., Rodrigues, A. S., Rocha, R., Louzada, F., et al. (2019). Zero-Adjusted Defective Regression Models for Modeling Lifetime Data. *Journal of Applied Statistics*, 46(13), 2434-2459.

Examples

```
data(insulin_gdm)
head(insulin_gdm)
table(insulin_gdm$family_diabetes)
table(insulin_gdm$prior_macrosomia)

# Fit final selected model from paper with Random activation scheme
fit_random <- ziprom_fit(
  formula_p0 = ~ prior_macrosomia,
  formula_p1 = ~ smoking + family_diabetes,
  time = insulin_gdm$time,
  status = insulin_gdm$status,
  data = insulin_gdm,
  activation = "random",
  frailty = FALSE
)
summary(fit_random)
```

kendall_tau_rhr

Kendall's Tau Measure of Dependence

Description

Computes Kendall's rank correlation coefficient τ for the shared frailty model under reversed hazard rate via numerical integration:

$$\tau = 4 \int_0^{\infty} s L_W''(s) L_W(s) ds - 1$$

Usage

```
kendall_tau_rhr(  
  frailty = c("WL", "GL", "GIG", "Gamma", "IG", "Lindley", "None"),  
  theta,  
  mu = NULL,  
  theta2 = NULL  
)
```

Arguments

- frailty Character string indicating frailty distribution.
- theta Numeric parameter for frailty distribution.
- mu Optional secondary parameter for "GL".
- theta2 Optional second parameter for "GIG".

Value

Numeric scalar representing Kendall’s tau in [0, 1].

Examples

```
kendall_tau_rhr("WL", theta = 3.4)  
kendall_tau_rhr("Gamma", theta = 2.0)
```

kidney_bivariate	<i>Kidney Infection Recurrence Bivariate Survival Data</i>
------------------	--

Description

Recurrence times of infection at the catheter insertion point for 38 kidney disease patients using portable dialysis machines, as studied in McGilchrist and Aisbett (1991), Pandey et al. (2022), and Tyagi et al. (2024).

Usage

```
kidney_bivariate
```

Format

- A data frame with 38 rows (one per patient) and 10 variables:
- id** Patient identification number (1 to 38).
 - t1** First infection recurrence time in days.
 - delta1** Censoring indicator for first recurrence (1 = infection occurred, 0 = censored).
 - t2** Second infection recurrence time in days.
 - delta2** Censoring indicator for second recurrence (1 = infection occurred, 0 = censored).

age Patient age in years.
sex Patient sex (0 = male, 1 = female).
gn Indicator for Glomerulonephritis kidney disease (1 = yes, 0 = no).
an Indicator for Acute Nephritis kidney disease (1 = yes, 0 = no).
pkd Indicator for Polycystic Kidney Disease (1 = yes, 0 = no).

Source

McGilchrist, C. A., and Aisbett, C. W. (1991). Regression with frailty in survival analysis. *Biometrics*, 47(2), 461-466. doi:10.2307/2532138

Pandey, A., Hanagal, D. D., and Tyagi, S. (2022). Generalised Lindley shared additive frailty regression model for bivariate survival data. *Statistics in Transition new series*, 23(4), 161-176. doi:10.2478/stattrans20220048

Tyagi, S., Pandey, A., Hanagal, D. D., and Chesneau, C. (2024). A study on comparisons of additive regression frailty models to counter heterogeneity: Bayesian strategies and case study. *Communications in Statistics - Simulation and Computation*. doi:10.1080/03610918.2024.2391874

Examples

```
data(kidney_bivariate)
head(kidney_bivariate)
summary(kidney_bivariate)
```

kidney_frailty	<i>Kidney Infection Recurrence Data</i>
----------------	---

Description

Recurrence times to kidney catheter infection for 38 kidney patients (McGilchrist and Aisbett, 1991).

Usage

```
kidney_frailty
```

Format

A data frame with 38 rows and 8 variables:

id Patient identification number (1 to 38).
time1 Time to first catheter infection (days).
time2 Time to second catheter infection (days).
status1 Censoring indicator for first recurrence (1 = infection, 0 = censored).
status2 Censoring indicator for second recurrence (1 = infection, 0 = censored).
age Age of patient in years.
sex Gender of patient (1 = male, 0 = female).
disease Disease type (GN, AN, PKD, Other).

Source

McGilchrist, C. A. and Aisbett, C. W. (1991). Regression with frailty in survival analysis. *Biometrics*, 47: 461-466.

References

Hanagal, D. D. and Dabade, A. D. (2013). A comparative study of shared frailty models for kidney infection data with generalized exponential baseline distribution. *Journal of Data Science*, 11(1): 109-142.

kidney_infection	<i>Kidney Infection Bivariate Survival Dataset</i>
------------------	--

Description

Recurrence times to catheter infection for 38 kidney dialysis patients from the classic study by McGilchrist and Aisbett (1991). Each patient has two recorded recurrence times (first and second insertion of catheter).

Usage

```
kidney_infection
```

Format

A data.frame with 38 rows (one per patient) and 9 variables:

- id** Patient identification number (1 to 38).
- t1** Time to first catheter infection or censoring (in days).
- d1** Censoring indicator for first event (1 = infection, 0 = censored).
- t2** Time to second catheter infection or censoring (in days).
- d2** Censoring indicator for second event (1 = infection, 0 = censored).
- age** Patient age in years.
- sex** Patient sex (1 = male, 2 = female).
- GN** Indicator for Glomerulonephritis disease type (1 = present, 0 = absent).
- AN** Indicator for Acute Nephritis disease type (1 = present, 0 = absent).
- PKD** Indicator for Polycystic Kidney Disease (1 = present, 0 = absent).

Source

McGilchrist, C. A. and Aisbett, C. W. (1991). Regression with frailty in survival analysis. *Biometrics*, 47(2), 461-466. <doi:10.2307/2532138>

References

Hanagal, D. D., Pandey, A., and Ganguly, A. (2017). Correlated gamma frailty models for bivariate survival data. *Communications in Statistics - Simulation and Computation*, 46(5), 3627-3644. <doi:10.1080/03610918.2015.1085559>

Hanagal, D. D. and Pandey, A. (2018). Correlated inverse Gaussian frailty models for bivariate survival data. *Communications in Statistics - Theory and Methods*, 49(14), 3455-3475. <doi:10.1080/03610926.2018.1549256>

Examples

```
data(kidney_infection)
head(kidney_infection)
```

laplace_frailty	<i>Laplace Transform of Frailty Distributions</i>
-----------------	---

Description

Computes the Laplace transform $L_W(s) = E[e^{-sW}]$ for various frailty distributions.

Usage

```
laplace_frailty(
  s,
  frailty = c("WL", "GL", "GIG", "Gamma", "IG", "Lindley", "None"),
  theta,
  mu = NULL,
  theta2 = NULL
)
```

Arguments

s	Numeric vector of non-negative arguments.
frailty	Character string indicating the frailty distribution. One of "WL", "GL", "GIG", "Gamma", "IG", "Lindley", "None".
theta	Numeric parameter for frailty distribution.
mu	Optional secondary parameter for "GL".
theta2	Optional second parameter for "GIG".

Value

Numeric vector of Laplace transform values.

Examples

```
laplace_frailty(c(0.1, 0.5, 1.0), frailty = "WL", theta = 3.4)
```

`logLik.frailty_bayes` *Model Extraction Methods*

Description

Model Extraction Methods

Usage

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

Arguments

<code>object</code>	Fitted frailty model object.
<code>...</code>	Additional arguments.

Value

The extracted model quantity.

`logLik.oneshot_kofm` *Extract Model Log-Likelihood*

Description

Extracts the maximized log-likelihood value from a fitted "oneshot_kofm" object.

Usage

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

Arguments

<code>object</code>	A fitted model object of class "oneshot_kofm".
<code>...</code>	Additional arguments.

Value

An object of class "logLik" with degrees of freedom attribute.

loglik_add_frailty *Bivariate Survival Log-Likelihood for Additive Shared Frailty Models*

Description

Evaluates the exact log-likelihood function under random right censoring for additive shared frailty regression models with Generalized Lindley, Weighted Lindley, Gamma, or Inverse Gaussian frailty, and Generalized Weibull or Generalized Log-Logistic Type-II baseline distributions.

Usage

```
loglik_add_frailty(
  params,
  t1,
  delta1,
  t2,
  delta2,
  X,
  frailty_type = c("genlindley", "wlindley", "gamma", "invgaussian"),
  baseline_dist = c("gw", "gll2")
)
```

Arguments

params	Numeric vector of all model parameters: <code>c(zeta1, delta1, xi1, zeta2, delta2, xi2, frailty_params, beta)</code> .
t1	Numeric vector of first survival times (positive).
delta1	Numeric vector of first censoring indicators (1 = event, 0 = censored).
t2	Numeric vector of second survival times (positive).
delta2	Numeric vector of second censoring indicators (1 = event, 0 = censored).
X	Covariate design matrix with <code>nrow(X) == length(t1)</code> .
frailty_type	Character string specifying the frailty distribution: "genlindley" (Generalized Lindley), "wlindley" (Weighted Lindley), "gamma" (Gamma), or "invgaussian" (Inverse Gaussian).
baseline_dist	Character string specifying baseline distribution: "gw" (Generalized Weibull) or "gll2" (Generalized Log-Logistic Type-II).

Value

A scalar numeric value representing the total log-likelihood.

References

Pandey, A., Hanagal, D. D., and Tyagi, S. (2022). Generalised Lindley shared additive frailty regression model for bivariate survival data. *Statistics in Transition new series*, 23(4), 161-176. doi:10.2478/stattrans20220048

Tyagi, S., Pandey, A., Hanagal, D. D., and Chesneau, C. (2024). A study on comparisons of additive regression frailty models to counter heterogeneity: Bayesian strategies and case study. *Communications in Statistics - Simulation and Computation*. doi:10.1080/03610918.2024.2391874

Examples

```
data(kidney_bivariate)
X <- as.matrix(kidney_bivariate[, c("age", "sex", "gn", "an", "pkd")])
pars <- c(zeta1 = 2.0, delta1 = 0.05, xi1 = 0.7,
          zeta2 = 2.7, delta2 = 0.06, xi2 = 0.7,
          theta = 2.3, mu = 1.4,
          beta1 = -0.1, beta2 = -9.0, beta3 = 2.4, beta4 = 1.6, beta5 = -50.0)
loglik_add_frailty(pars, kidney_bivariate$t1, kidney_bivariate$delta1,
                  kidney_bivariate$t2, kidney_bivariate$delta2,
                  X, frailty_type = "genlindley", baseline_dist = "gw")
```

loglik_biv_rhr

Bivariate Left-Censored Log-Likelihood under Reversed Hazard Rate

Description

Evaluates the log-likelihood function for bivariate survival data with left censoring and shared frailty based on reversed hazard rates.

Usage

```
loglik_biv_rhr(
  params,
  t1,
  t2,
  d1,
  d2,
  X = NULL,
  frailty = c("WL", "GL", "GIG", "Gamma", "IG", "Lindley", "None"),
  baseline = c("GE", "EG", "GIR", "MIW", "GLL-I", "GLL-II", "GIE", "Weibull",
               "Exponential", "Log-Logistic"),
  separate_baseline = TRUE
)
```

Arguments

params	Numeric vector of all model parameters (frailty parameters, baseline parameters for T1 and T2, and regression coefficients).
t1	Numeric vector of first failure/censoring times.
t2	Numeric vector of second failure/censoring times.
d1	Numeric or integer vector of censoring indicators for first component (1 = observed, 0 = left-censored).
d2	Numeric or integer vector of censoring indicators for second component (1 = observed, 0 = left-censored).
X	Numeric matrix of covariates.
frailty	Character string specifying frailty distribution.
baseline	Character string specifying baseline distribution.
separate_baseline	Logical indicating whether baseline parameters for T1 and T2 are estimated separately. Default is TRUE.

Value

Numeric scalar representing total log-likelihood.

Examples

```
data(austrian_twin)
params <- c(zeta1 = 12.4, delta1 = 0.11, zeta2 = 11.8, delta2 = 0.11,
            theta = 3.4, beta1 = 0.007, beta2 = -0.11, beta3 = 0.13)
X <- as.matrix(austrian_twin[, c("age", "k2", "k3")])
loglik_biv_rhr(params, austrian_twin$t1, austrian_twin$t2,
               austrian_twin$d1, austrian_twin$d2,
               X = X, frailty = "WL", baseline = "GE")
```

loglik_oneshot

Log-Likelihood Function for One-Shot Devices

Description

Computes the log-likelihood of one-shot device testing data under the shared gamma frailty model with multiple failure modes.

Usage

```
loglik_oneshot(
  par,
  count_mat,
  X_cov,
  tau,
```

```

    M,
    baseline = c("weibull", "exponential"),
    frailty = TRUE,
    powerset = NULL
  )

```

Arguments

<code>par</code>	Numeric vector of model parameters.
<code>count_mat</code>	An $I \times 2^M$ matrix of observed device counts.
<code>X_cov</code>	An $I \times p$ matrix of covariates.
<code>tau</code>	A positive numeric vector of inspection times.
<code>M</code>	An integer specifying the number of components ($M \geq 1$).
<code>baseline</code>	A character string: "exponential" (default) or "weibull".
<code>frailty</code>	A logical value: TRUE (default) for frailty model, FALSE for independence.
<code>powerset</code>	An optional list of power set subsets.

Details

Missing values are not permitted in `count_mat`, `X_cov`, `tau`, or `par`.

Value

A numeric scalar representing the total log-likelihood.

References

Ling, M.-H., Balakrishnan, N., Yu, C., and So, H. Y. (2021). Inference for One-Shot Devices with Dependent k-Out-of-M Structured Components under Gamma Frailty. *Mathematics*, 9(23), 3032. [doi:10.3390/math9233032](https://doi.org/10.3390/math9233032).

Examples

```

data(classH)
X_cov <- cbind(1, classH$stress)
tau <- classH$time
count_mat <- as.matrix(classH[, c("n_none", "n_T", "n_G", "n_both")])
par <- c(-4.99, -0.005, -16.44, 0.018, 0.5)
loglik_oneshot(par, count_mat, X_cov, tau, M = 2, baseline = "exponential", frailty = TRUE)

```

lpml	<i>Log Pseudo Marginal Likelihood (LPML)</i>
------	--

Description

Log Pseudo Marginal Likelihood (LPML)

Usage

```
lpml(object)
```

Arguments

object An object of class "frailty_bayes".

Value

Numeric LPML scalar value.

mice_tumor	<i>ED01 Mice Tumor Toxicological Data</i>
------------	---

Description

Toxicological study data from the ED01 experiment on 671 mice across 6 test groups (Table 12 of Ling et al., 2021), originally discussed by Lindsey and Ryan (1993). Animals were exposed to 2-acetylaminofluorene (2-AAF) at control (0 ppm) or high dose (150 ppm) and inspected at 12, 18, or 33 months for $M = 2$ endpoints: Bladder Tumor onset (T) and Death (D).

Usage

```
mice_tumor
```

Format

A data frame with 6 rows and 7 variables:

group Test group index ($i = 1, \dots, 6$).

dose Carcinogen 2-AAF dose level in ppm ($s_i = 0, 150$).

time Sacrifice / inspection time in months ($\tau_i = 12, 18, 33$).

n_none Number of sacrificed mice without tumor ($X = \emptyset$).

n_T Number of sacrificed mice with bladder tumor ($X = \{T\}$).

n_D Number of dead mice without bladder tumor ($X = \{D\}$).

n_both Number of dead mice with bladder tumor ($X = \{D, T\}$).

Source

Ling, M.-H., Balakrishnan, N., Yu, C., and So, H. Y. (2021). Inference for One-Shot Devices with Dependent k -Out-of- M Structured Components under Gamma Frailty. *Mathematics*, 9(23), 3032. doi:10.3390/math9233032.

Lindsey, J. C. and Ryan, L. M. (1993). A three-state multiplicative model for rodent tumorigenicity experiments. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 42(2), 283–300. doi:10.2307/2986234.

Examples

```
data(mice_tumor)
head(mice_tumor)
```

mttf_k_out_of_m	<i>Mean Time to Failure (MTTF) for k-out-of-M Systems</i>
-----------------	---

Description

Computes the Mean Time to Failure (MTTF) of k -out-of- M systems under shared gamma frailty.

Usage

```
mttf_k_out_of_m(object, ...)
```

Arguments

object	A fitted model object of class "oneshot_kofm" or "oneshot_frailty".
...	Additional arguments passed to methods.

Value

A data frame containing MTTF values across system configurations.

mttf_k_out_of_m.oneshot_frailty	<i>Mean Time to Failure (MTTF) for k-out-of-M Systems</i>
---------------------------------	---

Description

Calculates the Mean Time to Failure (MTTF) of k -out-of- M structured one-shot devices by numerical integration of the system reliability function $R_k(t)$.

Usage

```
## S3 method for class 'oneshot_frailty'
mttf_k_out_of_m(object, stress = NULL, k = NULL, upper_limit = 1e+05, ...)
```

Arguments

<code>object</code>	A fitted model object of class "oneshot_frailty".
<code>stress</code>	Optional numeric vector or matrix of stress covariate values.
<code>k</code>	Optional integer specifying system threshold (defaults to all $k = 1, \dots, M$).
<code>upper_limit</code>	Upper limit of integration in time units (default 1e5).
<code>...</code>	Additional arguments passed to other methods.

Details

The MTTF of a k -out-of- M system is given by

$$MTTF_k = \int_0^{\infty} R_k(t) dt$$

evaluated using adaptive quadrature via `stats::integrate`.

Value

A data frame containing:

k System threshold k .

system_type Descriptive string (e.g. "Parallel (k=1)", "Series (k=M)", "k-out-of-M").

mttf Computed Mean Time to Failure.

References

Yu, C., So, H. Y., and Balakrishnan, N. (2025). *Inference for one-shot devices with Weibull component lifetimes under gamma frailty model*. In: Stochastic Modeling and Statistical Methods, Elsevier Inc., Chapter 1, pp. 1–34. doi:10.1016/B9780443316944.000062.

Examples

```
data(classH)
start_par <- c(-3.997, -0.009, -14.412, 0.003, 0.004, -0.001, -5.493e-4, -0.005, 0.082)
fit_w <- oneshot_frailty(
  data = classH,
  time = "time",
  stress = "stress",
  counts = c("n_none", "n_T", "n_G", "n_both"),
  baseline = "weibull",
  frailty = TRUE,
  start = start_par
)
mttf_k_out_of_m(fit_w, stress = 400, upper_limit = 50000)
```

mttf_k_out_of_m.oneshot_kofm

Mean Lifetime and Confidence Intervals for k-Out-of-M Systems

Description

Computes the exact analytical mean lifetime $\mu_{0,k}$ (Mean Time to Failure, MTTF) of k -out-of- M structured one-shot devices under gamma frailty at normal operating stress s_0 , standard errors via the delta method, and both Asymptotic Confidence Intervals (ACI) and Log-Transformed Confidence Intervals (TCI).

Usage

```
## S3 method for class 'oneshot_kofm'
mttf_k_out_of_m(
  object,
  stress,
  k = NULL,
  ci_level = 0.95,
  upper_limit = 1e+06,
  ...
)
```

Arguments

object	A fitted model object of class "oneshot_kofm".
stress	A numeric scalar or vector representing the normal operating condition s_0 .
k	An optional integer vector specifying system structures ($1 \leq k \leq M$). If NULL (default), evaluates all structures $k = 1, \dots, M$.
ci_level	A numeric scalar between 0 and 1 specifying the confidence level (default 0.95).
upper_limit	Upper numerical integration limit for Weibull baseline (default 1e6).
...	Additional arguments passed to other methods.

Details

For Exponential baseline distributions under gamma frailty, the exact analytical mean lifetime at stress level s_0 is given by Equation (4) of Ling et al. (2021):

$$\mu_{0,k} = \frac{1}{1-\beta} \sum_{n=k}^M \left(\sum_{d=0}^{n-k} (-1)^d \binom{n}{d} \right) \sum_{X \in \mathcal{X}_n} \left(\sum_{m \in X} \lambda_{0,m} \right)^{-1}$$

The standard error is computed via the delta method:

$$se(\mu) = \sqrt{P_\mu V(\hat{\theta}) P_\mu^T}$$

where $P_\mu = \nabla_\theta \mu_{0,k}$ is the analytical gradient vector (Section 5, Ling et al., 2021).

The $100(1 - \delta)\%$ Log-Transformed Confidence Interval (TCI) is:

$$\left(\hat{\mu}_{0,k} \exp \left(-\frac{z_{1-\delta/2} se(\mu)}{\hat{\mu}_{0,k}} \right), \hat{\mu}_{0,k} \exp \left(\frac{z_{1-\delta/2} se(\mu)}{\hat{\mu}_{0,k}} \right) \right)$$

ensuring strictly positive lower bounds.

Missing values are not permitted.

Value

A data frame containing:

k The system structure threshold (k -out-of- M).

structure Character description ("Parallel (k=1)", "Series (k=M)", or "k-out-of-M").

stress Operating stress level s_0 .

mttf Estimated mean lifetime $\mu_{0,k}$.

se Standard error of mean lifetime $se(\mu)$ from the delta method.

aci_lower Lower bound of the Asymptotic Confidence Interval (ACI).

aci_upper Upper bound of the Asymptotic Confidence Interval (ACI).

tci_lower Lower bound of the Log-Transformed Confidence Interval (TCI).

tci_upper Upper bound of the Log-Transformed Confidence Interval (TCI).

References

Ling, M.-H., Balakrishnan, N., Yu, C., and So, H. Y. (2021). Inference for One-Shot Devices with Dependent k-Out-of-M Structured Components under Gamma Frailty. *Mathematics*, 9(23), 3032. [doi:10.3390/math9233032](https://doi.org/10.3390/math9233032).

Balakrishnan, N. and Ling, M. H. (2014). Gamma lifetimes and one-shot device testing analysis. *Reliability Engineering & System Safety*, 126, 54–64. [doi:10.1016/j.res.2014.01.009](https://doi.org/10.1016/j.res.2014.01.009).

Examples

```
data(classH)
fit <- oneshot_kofm(
  data = classH,
  time = "time",
  stress = "stress",
  counts = c("n_none", "n_T", "n_G", "n_both"),
  baseline = "exponential"
)
# Mean lifetimes at normal operating condition s0 = 356 F
mttf_k_out_of_m(fit, stress = 356)
```

oneshot_frailty

*Fit Shared Gamma Frailty Model for One-Shot Device Testing Data***Description**

Fits the Weibull or Exponential shared gamma frailty model to one-shot device testing data with multiple failure modes under constant-stress accelerated life tests, computing maximum likelihood parameter estimates, standard errors, asymptotic confidence intervals, and model selection criteria.

Usage

```
oneshot_frailty(
  formula = NULL,
  data,
  time = "time",
  stress = "stress",
  counts = NULL,
  components = NULL,
  baseline = c("weibull", "exponential"),
  frailty = TRUE,
  method = c("mle", "em"),
  start = NULL,
  ci_level = 0.95,
  control = list(),
  ...
)
```

Arguments

formula	Optional formula of the form $\text{cbind}(n_none, n_T, \dots) \sim \text{stress}$ or $\text{cbind}(n_none, n_T, \dots) \sim \text{stress} + \text{time}$.
data	A data frame containing the test group observations, covariates, inspection times, and malfunction subset counts.
time	A character string specifying the column name in data containing inspection times τ_i .
stress	An optional character vector of column names in data containing stress covariates.
counts	An optional character vector of 2^M column names in data containing device counts for each subset in the power set $P(\Omega)$.
components	An optional character vector of M component / failure mode names (e.g., $\text{c}(\text{"Turn"}, \text{"Ground"})$). If omitted, defaults to $\text{c}(\text{"Comp1"}, \text{"Comp2"}, \dots)$.
baseline	A character string specifying the baseline lifetime distribution: "weibull" (default) or "exponential".
frailty	A logical value: TRUE (default) to fit the shared gamma frailty model ($\beta > 0$), or FALSE to fit the independence model ($\beta = 0$).

<code>method</code>	A character string specifying the estimation algorithm: "mle" (default, direct Quasi-Newton) or "em" (Expectation-Maximization algorithm).
<code>start</code>	An optional numeric vector of starting values for the parameters.
<code>ci_level</code>	A numeric scalar between 0 and 1 specifying the asymptotic confidence level (default 0.95).
<code>control</code>	An optional list of control parameters passed to the optimizer (e.g. <code>maxit</code> , <code>tol</code>).
<code>...</code>	Additional arguments.

Details

The shared gamma frailty model links the component baseline hazard functions $h_{0,i,m}(t) = \alpha_{i,m} \lambda_{i,m} (\lambda_{i,m} t)^{\alpha_{i,m}-1}$ with stress covariates via log-linear links $\lambda_{i,m} = \exp(a_m^T x_i)$ and $\alpha_{i,m} = \exp(b_m^T x_i)$.

Missing values are checked upon input; observations with missing values in required variables are rejected with an informative error message.

Value

An S3 object of class "oneshot_frailty" containing:

coefficients A named numeric vector of maximum likelihood estimates.

vcov The asymptotic variance-covariance matrix of the estimators.

se A named numeric vector of standard errors.

confint A matrix with lower (LCI) and upper (UCI) asymptotic confidence bounds.

loglik The maximized log-likelihood value.

n_params The total number of estimated parameters.

aic Akaike Information Criterion $(-2 \log L + 2k)$.

bic Bayesian Information Criterion $(-2 \log L + k \ln N)$.

aicc Corrected AIC for small sample sizes.

hqic Hannan-Quinn Information Criterion.

sample_size Total number of devices tested across all groups.

fitted_probs An $I \times 2^M$ matrix of model-estimated cell probabilities.

fitted_counts An $I \times 2^M$ matrix of expected device counts.

residuals An $I \times 2^M$ matrix of standardized Pearson residuals.

kendall_tau A list with the estimated Kendall's tau association ($\tau = \beta/(\beta + 2)$), standard error, and confidence interval (if `frailty = TRUE`).

M The number of components.

components Character vector of component names.

baseline Baseline distribution used ("weibull" or "exponential").

frailty Logical flag indicating whether gamma frailty was modeled.

method Estimation method used ("mle" or "em").

data The processed input data frame.

call The matched function call.

References

Yu, C., So, H. Y., and Balakrishnan, N. (2025). *Inference for one-shot devices with Weibull component lifetimes under gamma frailty model*. In: *Stochastic Modeling and Statistical Methods*, Elsevier Inc., Chapter 1, pp. 1–34. doi:10.1016/B9780443316944.000062.

Examples

```
data(classH)
# Fit shared gamma frailty model with Weibull baseline
fit_w <- oneshot_frailty(
  data = classH,
  time = "time",
  stress = "stress",
  counts = c("n_none", "n_T", "n_G", "n_both"),
  components = c("Turn", "Ground"),
  baseline = "weibull",
  frailty = TRUE
)
print(fit_w)
summary(fit_w)
```

oneshot_kofm

Fit Shared Gamma Frailty Model for k-Out-of-M One-Shot Devices

Description

Fits the shared gamma frailty model with Exponential or Weibull component lifetimes to one-shot device testing data with multiple failure modes under constant-stress accelerated life tests. Computes maximum likelihood parameter estimates, standard errors via the missing information principle, asymptotic confidence intervals, and information criteria.

Usage

```
oneshot_kofm(
  formula = NULL,
  data,
  time = "time",
  stress = "stress",
  counts = NULL,
  components = NULL,
  baseline = c("exponential", "weibull"),
  frailty = TRUE,
  method = c("em", "mle"),
  start = NULL,
  ci_level = 0.95,
  control = list(),
  ...
)
```

Arguments

formula	Optional formula of the form $\text{cbind}(n_none, n_T, \dots) \sim \text{stress}$ or $\text{cbind}(n_none, n_T, \dots) \sim \text{stress} + \text{time}$.
data	A data frame containing the test group observations, covariates, inspection times, and malfunction subset counts.
time	A character string specifying the column name in data containing inspection times τ_i .
stress	An optional character vector of column names in data containing stress covariates.
counts	An optional character vector of 2^M column names in data containing device counts for each subset in the power set $P(\Omega)$.
components	An optional character vector of M component / failure mode names (e.g., $c(\text{"Turn"}, \text{"Ground"})$). If omitted, defaults to $c(\text{"Comp1"}, \text{"Comp2"}, \dots)$.
baseline	A character string specifying the baseline lifetime distribution: "exponential" (default) or "weibull".
frailty	A logical value: TRUE (default) to fit the shared gamma frailty model ($\beta > 0$), or FALSE to fit the independence model ($\beta = 0$).
method	A character string specifying the estimation algorithm: "em" (default, Expectation-Maximization) or "mle" (direct Quasi-Newton optimization).
start	An optional numeric vector of starting values for the parameters.
ci_level	A numeric scalar between 0 and 1 specifying the asymptotic confidence level (default 0.95).
control	An optional list of control parameters passed to the optimizer (e.g. <code>maxit</code> , <code>tol</code>).
...	Additional arguments.

Details

The shared gamma frailty model links the component hazard rates $\lambda_{i,m} = \exp(a_m^T x_i)$ with stress covariates.

Missing values are checked upon input; observations with missing values in required variables are rejected with an informative error message.

Value

An S3 object of class "oneshot_kofm" containing:

coefficients A named numeric vector of maximum likelihood estimates.

vcov The asymptotic variance-covariance matrix of the estimators.

se A named numeric vector of standard errors.

confint A matrix with lower (LCI) and upper (UCI) asymptotic confidence bounds.

loglik The maximized log-likelihood value.

n_params The total number of estimated parameters.

aic Akaike Information Criterion ($-2 \log L + 2k$).

bic Bayesian Information Criterion ($-2 \log L + k \ln N$).

aicc Corrected AIC for small sample sizes.

hqic Hannan-Quinn Information Criterion.

sample_size Total number of devices tested across all groups.

fitted_probs An $I \times 2^M$ matrix of model-estimated cell probabilities.

fitted_counts An $I \times 2^M$ matrix of expected device counts.

residuals An $I \times 2^M$ matrix of standardized Pearson residuals.

kendall_tau A list with the estimated Kendall's tau association ($\tau = \beta/(\beta + 2)$), standard error, and confidence interval (if `frailty = TRUE`).

M The number of components.

components Character vector of component names.

baseline Baseline distribution used ("exponential" or "weibull").

frailty Logical flag indicating whether gamma frailty was modeled.

method Estimation method used ("em" or "mle").

data The processed input data frame.

call The matched function call.

References

Ling, M.-H., Balakrishnan, N., Yu, C., and So, H. Y. (2021). Inference for One-Shot Devices with Dependent k-Out-of-M Structured Components under Gamma Frailty. *Mathematics*, 9(23), 3032. [doi:10.3390/math9233032](https://doi.org/10.3390/math9233032).

Examples

```
data(classH)
# Fit shared gamma frailty model with Exponential baseline
fit_exp <- oneshot_kofm(
  data = classH,
  time = "time",
  stress = "stress",
  counts = c("n_none", "n_T", "n_G", "n_both"),
  components = c("Turn", "Ground"),
  baseline = "exponential",
  frailty = TRUE,
  method = "em"
)
print(fit_exp)
summary(fit_exp)
```

p_rhr_baseline	<i>Baseline Cumulative Distribution Function under Reversed Hazard Rate</i>
----------------	---

Description

Computes the cumulative distribution function (CDF) for baseline distributions under reversed hazard rate models.

Usage

```
p_rhr_baseline(  
  t,  
  dist = c("GE", "EG", "GIR", "MIW", "GLL-I", "GLL-II", "GIE", "Weibull", "Exponential",  
           "Log-Logistic"),  
  params  
)
```

Arguments

- t Numeric vector of positive failure times.
- dist Character string specifying baseline distribution. One of "GE", "EG", "GIR", "MIW", "GLL-I", "GLL-II", "GIE", "Weibull", "Exponential", "Log-Logistic".
- params Numeric named vector of baseline parameters.

Value

Numeric vector of cumulative probabilities in (0, 1).

Examples

```
p_rhr_baseline(c(10, 20, 30), dist = "GE", params = c(zeta = 12.4, delta = 0.11))
```

plot.add_frailty	<i>Diagnostic and Visualization Plots for Additive Shared Frailty Models</i>
------------------	--

Description

Generates MCMC trace plots, autocorrelation (ACF) plots, posterior density plots, baseline hazard and survival curves, and goodness-of-fit diagnostic plots.

Usage

```
## S3 method for class 'add_frailty'
plot(
  x,
  type = c("trace", "acf", "density", "survival", "hazard", "gof"),
  params = NULL,
  max_time = 100,
  ...
)
```

Arguments

<code>x</code>	An object of class "add_frailty" or "add_frailty_bayes".
<code>type</code>	Character string specifying plot type: "trace", "acf", "density", "survival", "hazard", or "gof".
<code>params</code>	Optional character vector of parameter names to plot. If NULL, plots the first 4 parameters.
<code>max_time</code>	Maximum time horizon for survival/hazard plots (default 100).
<code>...</code>	Additional graphical parameters passed to plotting functions.

Value

Invisible NULL. Produces base R graphics.

References

Pandey, A., Hanagal, D. D., and Tyagi, S. (2022). Generalised Lindley shared additive frailty regression model for bivariate survival data. *Statistics in Transition new series*, 23(4), 161-176. [doi:10.2478/stattrans20220048](https://doi.org/10.2478/stattrans20220048)

Tyagi, S., Pandey, A., Hanagal, D. D., and Chesneau, C. (2024). A study on comparisons of additive regression frailty models to counter heterogeneity: Bayesian strategies and case study. *Communications in Statistics - Simulation and Computation*. [doi:10.1080/03610918.2024.2391874](https://doi.org/10.1080/03610918.2024.2391874)

Examples

```
data(kidney_bivariate)
fit_mle <- fit_add_frailty_mle(
  data = kidney_bivariate,
  frailty_type = "wlindley", baseline_dist = "gw"
)
plot(fit_mle, type = "survival")
plot(fit_mle, type = "hazard")
```

plot.frailty_bayes	<i>S3 Plot Method for Bayesian Frailty Models</i>
--------------------	---

Description

S3 Plot Method for Bayesian Frailty Models

Usage

```
## S3 method for class 'frailty_bayes'
plot(x, which = c("trace", "density", "autocorr", "survival", "hazard"), ...)
```

Arguments

x	An object of class "frailty_bayes".
which	Character vector specifying plot types: "trace", "density", "autocorr", "survival", or "hazard".
...	Additional graphical parameters.

Value

Invisibly returns NULL.

Produces trace plots, posterior density plots, autocorrelation plots, estimated survival curves, or estimated hazard rate functions.

plot.oneshot_frailty	<i>Diagnostic and Reliability Plots for One-Shot Device Testing</i>
----------------------	---

Description

Provides diagnostic visualization and reliability plotting for fitted "oneshot_frailty" model objects.

Usage

```
## S3 method for class 'oneshot_frailty'
plot(
  x,
  which = c("all", "reliability", "survival", "residuals", "fitted"),
  stress = NULL,
  t_max = NULL,
  ask = prod(graphics::par("mfcol")) < 2 && grDevices::dev.interactive(),
  ...
)
```

Arguments

<code>x</code>	A fitted object of class "oneshot_frailty".
<code>which</code>	A character string or integer specifying the plot type: "all" (default), "reliability" (1), "survival" (2), "residuals" (3), or "fitted" (4).
<code>stress</code>	Optional numeric vector or value of stress covariate for reliability evaluation.
<code>t_max</code>	Optional maximum time for reliability curves (defaults to $1.5x \max(\tau)$).
<code>ask</code>	Logical: TRUE to prompt before new plot pages in interactive mode.
<code>...</code>	Additional graphical parameters passed to plot.

Details

The available plot types are:

"reliability" System reliability curves $R_k(t)$ for $k = 1$ (parallel), $k = M$ (series), and intermediate k -out-of- M structures.

"survival" Marginal component survival curves $S_m(t)$.

"residuals" Standardized Pearson residuals vs. test group index.

"fitted" Observed vs. model-fitted device counts for each malfunction subset.

Value

Invisible NULL. Produces base R graphical plots.

References

Yu, C., So, H. Y., and Balakrishnan, N. (2025). *Inference for one-shot devices with Weibull component lifetimes under gamma frailty model*. In: Stochastic Modeling and Statistical Methods, Elsevier Inc., Chapter 1, pp. 1–34. doi:[10.1016/B9780443316944.000062](https://doi.org/10.1016/B9780443316944.000062).

Examples

```
data(classH)
start_par <- c(-3.997, -0.009, -14.412, 0.003, 0.004, -0.001, -5.493e-4, -0.005, 0.082)
fit_w <- oneshot_frailty(
  data = classH,
  time = "time",
  stress = "stress",
  counts = c("n_none", "n_T", "n_G", "n_both"),
  baseline = "weibull",
  frailty = TRUE,
  start = start_par
)
plot(fit_w, which = "reliability", stress = 400)
```

plot.oneshot_kofm *Diagnostic and Visual Plots for One-Shot Frailty Models*

Description

Generates publication-ready diagnostic plots for fitted "oneshot_kofm" models, including k -out-of- M system reliability curves, component marginal survival curves, observed vs. fitted cell proportions, and standardized Pearson residuals.

Usage

```
## S3 method for class 'oneshot_kofm'
plot(
  x,
  which = c("all", "reliability", "marginal", "fit", "residuals"),
  stress = NULL,
  t_max = NULL,
  col = NULL,
  lty = NULL,
  lwd = 2,
  ...
)
```

Arguments

x	A fitted model object of class "oneshot_kofm".
which	A character string or integer specifying the plot type: "all" (default), "reliability" (or 1), "marginal" (or 2), "fit" (or 3), or "residuals" (or 4).
stress	A numeric scalar specifying the operating stress condition for reliability curves. If NULL (default), uses the minimum stress level in the dataset.
t_max	A positive numeric scalar specifying the maximum mission time for curves. If NULL (default), uses $1.5 \times \max(\tau_i)$.
col	Optional vector of colors for curves.
lty	Optional vector of line types.
lwd	Line width (default 2).
...	Additional graphical parameters passed to plotting functions.

Value

No return value, called for plotting side effects.

plot.rhr_bayes_fit *Plot Diagnostics for Bayesian Shared RHR Frailty Model*

Description

Produces graphical diagnostics for fitted Bayesian shared frailty models, including trace plots, posterior density plots with HPD intervals, autocorrelation decay, Kolmogorov-Smirnov goodness-of-fit plots, conditional predictive ordinate (CPO) plots, and reversed hazard rate (RHR) curves.

Usage

```
## S3 method for class 'rhr_bayes_fit'
plot(
  x,
  type = c("trace", "density", "autocorr", "ks", "cpo", "rhr"),
  params = NULL,
  ask = FALSE,
  ...
)
```

Arguments

x	An object of class "rhr_bayes_fit".
type	Character vector or string specifying plot type. One or more of: "trace", "density", "autocorr", "ks", "cpo", "rhr". Default is c("trace", "density").
params	Optional character vector of parameter names to plot.
ask	Logical; if TRUE, user is prompted before each new page of plots. Default is FALSE.
...	Additional arguments passed to base plotting functions.

Value

Invisible NULL.

Examples

```
data(austrian_twin)
fit <- fit_rhr_frailty_bayes(cbind(t1, t2, d1, d2) ~ age, data = austrian_twin[1:30, ],
                           frailty = "WL", baseline = "GE", n_iter = 40, burn_in = 10)
plot(fit, type = "trace", params = "theta")
```

plot.rhr_mle_fit	<i>Plot Diagnostics for Maximum Likelihood Shared RHR Frailty Model</i>
------------------	---

Description

Produces goodness-of-fit and reversed hazard curves for an MLE shared frailty model.

Usage

```
## S3 method for class 'rhr_mle_fit'
plot(x, type = c("ks", "rhr"), ...)
```

Arguments

x	An object of class "rhr_mle_fit".
type	Character string indicating plot type: "ks" or "rhr".
...	Additional arguments passed to plotting functions.

Value

Invisible NULL.

Examples

```
data(austrian_twin)
fit <- fit_rhr_frailty_mle(cbind(t1, t2, d1, d2) ~ age, data = austrian_twin[1:30, ],
                          frailty = "WL", baseline = "GE")
plot(fit, type = "ks")
```

plot.ziprom_fit	<i>Plot Diagnostics and Survival Curves for ziprom_fit Objects</i>
-----------------	--

Description

Provides rich visual diagnostics including fitted population and susceptible survival curves, multi-level significance forecast plots (90%, 95%, 99%), normal QQ-plots of randomized quantile residuals with simulated envelopes, residual index plots, hazard/density plots, and activation scheme comparisons.

Usage

```
## S3 method for class 'ziprom_fit'
plot(
  x,
  which = c("survival", "forecast", "qq", "residuals", "hazard", "density",
    "activation_compare"),
  newdata = NULL,
  times = NULL,
  level = c(0.9, 0.95, 0.99),
  col = NULL,
  lty = NULL,
  lwd = 2,
  main = NULL,
  xlab = NULL,
  ylab = NULL,
  ask = FALSE,
  ...
)

plot_survival_fit(x, ...)

plot_forecast(x, ...)

plot_quantile_residuals(x, ...)
```

Arguments

<code>x</code>	An object of class "ziprom_fit".
<code>which</code>	Character string specifying the plot type: • "survival" (default): Fitted population survival curve $S_p(t)$ with Kaplan-Meier overlay. • "forecast": Forecast survival curve with 90%, 95%, and 99% confidence bands. • "qq": Normal QQ-plot of randomized quantile residuals with 95% confidence envelope. • "residuals": Residuals versus observation index or fitted values. • "hazard": Baseline hazard rate curve. • "density": Population density curve. • "activation_compare": Comparative survival curves for Minimum, Maximum, and Random schemes.
<code>newdata</code>	Optional data frame containing covariate values for plotting.
<code>times</code>	Optional numeric vector of time points for plotting curves.
<code>level</code>	Numeric vector of confidence levels for forecast bands. Default is <code>c(0.90, 0.95, 0.99)</code> .
<code>col</code>	Vector of colors used for curves and confidence bands.

lty	Vector of line types.
lwd	Line width (default lwd = 2).
main	Custom title for the plot.
xlab	Custom x-axis label.
ylab	Custom y-axis label.
ask	Logical; if TRUE, the user is asked before each plot. Default is FALSE.
...	Additional arguments passed to base graphics functions.

Value

Invisible NULL. The function is called for its graphical side effects.

References

Assuncao, D., Ramos, P. L., Tomazella, V., Leao, J., Osses, L., and Rodrigues, A. (2026). Zero-Inflated Promotion Cure-Rate Models With Frailty and Flexible Activation Schemes. *Biometrical Journal*, 68:e70162. doi:[10.1002/bimj.70162](https://doi.org/10.1002/bimj.70162)

Examples

```
set.seed(42)
sim_dat <- sim_ziprom(n = 60, activation = "random")
fit <- ziprom_fit(survival::Surv(time, status) ~ x, data = sim_dat, activation = "random")
plot(fit, which = "survival")
plot(fit, which = "forecast")
plot(fit, which = "qq")
```

plot_activation_compare

Compare Survival Curves Across Activation Schemes

Description

Fits and compares the fitted population survival curves under Minimum, Maximum, and Random activation schemes for a given dataset.

Usage

```
plot_activation_compare(object, times = NULL, col = NULL, lwd = 2, ...)
```

Arguments

object	An object of class "ziprom_fit".
times	Optional numeric vector of evaluation times.
col	Colors for the three activation schemes.
lwd	Line width (default is 2).
...	Additional graphical parameters.

Value

Invisible list of fitted models for each activation scheme.

Examples

```
set.seed(42)
sim_dat <- sim_ziprom(n = 60, activation = "random")
fit <- ziprom_fit(survival::Surv(time, status) ~ x, data = sim_dat, activation = "random")
plot_activation_compare(fit)
```

predict.frailty_bayes *Predict Method for Frailty Models*

Description

Predicts survival probabilities, cumulative hazards, or frailty estimates.

Usage

```
## S3 method for class 'frailty_bayes'
predict(
  object,
  newdata = NULL,
  times = NULL,
  type = c("survival", "cumhaz", "hazard"),
  ...
)
```

Arguments

object	Fitted model object.
newdata	New data frame.
times	Time points for evaluation.
type	Prediction type: "survival", "cumhaz", or "hazard".
...	Additional arguments.

Value

A numeric vector or matrix of predictions.

predict.oneshot_kofm	<i>Predict System Reliability or Hazard Rates</i>
----------------------	---

Description

Generates predictions of k -out-of- M system reliability or component failure rates for new stress levels and time intervals from a fitted "oneshot_kofm" object.

Usage

```
## S3 method for class 'oneshot_kofm'
predict(
  object,
  newdata = NULL,
  type = c("reliability", "mttf", "moments"),
  k = NULL,
  ...
)
```

Arguments

- object A fitted model object of class "oneshot_kofm".
- newdata An optional data frame with new stress or time values.
- type A character string: "reliability" (default), "mttf", or "moments".
- k An optional integer vector specifying system structures.
- ... Additional arguments passed to underlying prediction functions.

Value

A data frame of predictions.

predict.ziprom_fit	<i>Prediction and Multi-Level Significance Forecasting for ziprom_fit Models</i>
--------------------	--

Description

Generates predicted survival probabilities, susceptible survival, cumulative hazard, hazard rates, probability density, zero/cure proportions, and future forecasts with 90%, 95%, and 99% confidence / significance intervals.

Usage

```
## S3 method for class 'ziprom_fit'
predict(
  object,
  newdata = NULL,
  type = c("survival", "susceptible_survival", "hazard", "cumhazard", "density", "p0",
    "p1", "forecast", "quantile"),
  times = NULL,
  quantiles = c(0.25, 0.5, 0.75),
  level = c(0.9, 0.95, 0.99),
  ...
)
```

Arguments

object	An object of class "ziprom_fit".
newdata	Optional data frame containing covariate values for prediction. If omitted, predictions are made at the sample mean of the covariates.
type	Character string indicating the type of prediction: • "survival" (default): Population survival probability $S_p(t)$. • "susceptible_survival": Proper survival for susceptible subjects $S_p^*(t)$. • "hazard": Baseline or population hazard rate. • "cumhazard": Cumulative baseline hazard $H_0(t)$. • "density": Population density $f_p(t)$. • "p0": Zero-failure probability p_0. • "p1": Cure fraction p_1. • "forecast": Detailed forecast table across time points with multi-level intervals. • "quantile": Survival quantiles (times) for susceptible individuals.
times	Numeric vector of future or evaluation time points. If NULL, a 100-point regular grid from 0 to $\max(t)$ is constructed.
quantiles	Numeric vector of quantile levels (in (0, 1)) for type = "quantile". Default is $c(0.25, 0.50, 0.75)$.
level	Numeric vector of confidence levels for interval estimation. Default is $c(0.90, 0.95, 0.99)$.
...	Additional arguments.

Value

Depending on type:

- For "survival", "susceptible_survival", "hazard", "cumhazard", "density": A numeric matrix (or vector) of predictions with rows corresponding to observations in newdata and columns to times.
- For "p0" or "p1": A numeric vector of fitted probabilities.

- For "quantile": A numeric matrix of estimated quantile survival times.
- For "forecast": A comprehensive data frame containing time, survival, se, lower_90, upper_90, lower_95, upper_95, lower_99, and upper_99.

References

Assuncao, D., Ramos, P. L., Tomazella, V., Leao, J., Osses, L., and Rodrigues, A. (2026). Zero-Inflated Promotion Cure-Rate Models With Frailty and Flexible Activation Schemes. *Biometrical Journal*, 68:e70162. doi:[10.1002/bimj.70162](https://doi.org/10.1002/bimj.70162)

Examples

```
set.seed(42)
sim_dat <- sim_ziprom(n = 60, activation = "random")
fit <- ziprom_fit(survival::Surv(time, status) ~ x, data = sim_dat, activation = "random")
fc <- predict(fit, type = "forecast", times = seq(0, 5, length.out = 10))
print(head(fc))
```

print.frailty_bayes *Print Methods for Frailty Models*

Description

Print Methods for Frailty Models

Usage

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

Arguments

x	An object of class "frailty_bayes" or "frailty_mle".
...	Additional arguments.

Value

The evaluated object x, invisibly.

print.oneshot_kofm	<i>Print Method for One-Shot Frailty Model Fits</i>
--------------------	---

Description

Prints a concise summary of the fitted "oneshot_kofm" model object.

Usage

```
## S3 method for class 'oneshot_kofm'
print(x, digits = 4, ...)
```

Arguments

x	A fitted model object of class "oneshot_kofm".
digits	An integer specifying the number of significant digits (default 4).
...	Additional arguments.

Value

No return value, called for printing side effects.

print.summary.oneshot_kofm	<i>Print Summary for One-Shot Frailty Model Fits</i>
----------------------------	--

Description

Prints the summary table for class "summary.oneshot_kofm".

Usage

```
## S3 method for class 'summary.oneshot_kofm'
print(x, digits = 4, ...)
```

Arguments

x	An object of class "summary.oneshot_kofm".
digits	An integer specifying the number of significant digits (default 4).
...	Additional arguments.

Value

No return value, called for printing side effects.

print.ziprom_fit	<i>Methods for ziprom_fit Objects</i>
------------------	---------------------------------------

Description

Standard S3 methods for inspecting, summarizing, and extracting components from fitted zero-inflated promotion cure-rate models.

Usage

```
## S3 method for class 'ziprom_fit'
print(x, digits = 4, ...)

## S3 method for class 'ziprom_fit'
summary(object, level = 0.95, ...)

## S3 method for class 'summary.ziprom_fit'
print(x, digits = 4, ...)

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

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

## S3 method for class 'ziprom_fit'
confint(object, parm, level = 0.95, ...)

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

## S3 method for class 'ziprom_fit'
AIC(object, ..., k = 2)

## S3 method for class 'ziprom_fit'
BIC(object, ...)

## S3 method for class 'ziprom_fit'
residuals(
  object,
  type = c("quantile", "cox-snell", "deviance", "response"),
  ...
)
```

Arguments

x An object of class "ziprom_fit" or "summary.ziprom_fit".

digits	Integer indicating the number of decimal digits to print. Default is 4.
...	Additional arguments passed to or from other methods.
object	An object of class "ziprom_fit".
level	Confidence level for intervals (e.g. 0.90, 0.95, 0.99). Default is 0.95.
parm	Optional vector of parameter names or indices for confint.
k	Penalty parameter for AIC (default k = 2).
type	Character string indicating residual type: "quantile" (default), "cox-snell", "deviance", or "response".

Value

- `print.ziprom_fit`: Invisible copy of the model object.
- `summary.ziprom_fit`: A summary object of class "summary.ziprom_fit".
- `coef.ziprom_fit`: Named vector of estimated model coefficients.
- `vcov.ziprom_fit`: Asymptotic variance-covariance matrix.
- `confint.ziprom_fit`: Matrix of confidence intervals.
- `logLik.ziprom_fit`: Log-likelihood value with attributes `df` and `nobs`.
- `AIC.ziprom_fit`: Akaike Information Criterion.
- `BIC.ziprom_fit`: Bayesian Information Criterion.
- `residuals.ziprom_fit`: Vector of residuals.

q_rhr_baseline

Baseline Quantile Function under Reversed Hazard Rate

Description

Computes quantiles for baseline distributions used in reversed hazard rate models.

Usage

```
q_rhr_baseline(
  p,
  dist = c("GE", "EG", "GIR", "MIW", "GLL-I", "GLL-II", "GIE", "Weibull", "Exponential",
    "Log-Logistic"),
  params
)
```

Arguments

p	Numeric vector of probabilities in (0, 1).
dist	Character string specifying baseline distribution.
params	Numeric named vector of baseline parameters.

Value

Numeric vector of quantiles.

Examples

```
q_rhr_baseline(c(0.1, 0.5, 0.9), dist = "GE", params = c(zeta = 12.4, delta = 0.11))
```

quantile_proper

Quantile Function for Susceptible Group

Description

Computes the quantile survival time t such that $S_p^*(t) = 1 - v$ for uniform probabilities $v \in (0, 1)$.

Usage

```
quantile_proper(
  v,
  theta,
  alpha,
  shape,
  scale,
  activation = c("random", "minimum", "maximum"),
  baseline = c("weibull", "gompertz", "exponential")
)
```

Arguments

<code>v</code>	Numeric vector of probabilities in (0, 1).
<code>theta</code>	Numeric vector or scalar representing the latent Poisson parameter (> 0).
<code>alpha</code>	Numeric scalar representing Gamma frailty variance (≥ 0).
<code>shape</code>	Numeric vector or scalar representing baseline shape parameter(s) (> 0).
<code>scale</code>	Numeric vector or scalar representing baseline scale parameter(s) (> 0).
<code>activation</code>	Character string indicating the activation scheme: "random" (default), "minimum", or "maximum".
<code>baseline</code>	Character string indicating the baseline hazard distribution: "weibull" (default), "gompertz", or "exponential".

Value

Numeric vector of quantile survival times for susceptible individuals.

reliability_k_out_of_m

Reliability Evaluation for k-out-of-M Systems

Description

Evaluates the exact system reliability function $R_k(t)$ of k -out-of- M structured one-shot devices under shared gamma frailty at specified operating times and stress levels.

Usage

```
reliability_k_out_of_m(object, ...)
```

Arguments

object	A fitted model object of class "oneshot_kofm" or "oneshot_frailty".
...	Additional arguments passed to methods.

Value

A data frame containing mission times, stress levels, system thresholds k , and evaluated system reliabilities.

reliability_k_out_of_m.oneshot_frailty

System Reliability for k-out-of-M Systems under Gamma Frailty

Description

Computes the system reliability $R_k(t)$ at specified mission times t for k -out-of- M structured one-shot devices under shared gamma frailty and Weibull or Exponential baseline lifetime distributions.

Usage

```
## S3 method for class 'oneshot_frailty'
reliability_k_out_of_m(object, t, stress = NULL, k = NULL, ...)
```

Arguments

object	A fitted model object of class "oneshot_frailty".
t	A non-negative numeric vector of mission time points at which reliability is evaluated.
stress	An optional numeric vector or matrix of stress covariate values.
k	An integer specifying the system threshold (the system functions if at least k of its M components function).
...	Additional arguments passed to other methods.

Details

For a k -out-of- M system, the system operates if at least k components survive. Using the inclusion-exclusion principle and the power set $P(\Omega)$, the unconditional system reliability is

$$R_k(t) = \sum_{X: |X| \leq M-k} P(X, t)$$

where $|X|$ is the number of malfunctioned components.

Special cases include:

- $k = 1$: Parallel system (functions if at least 1 component functions).
- $k = M$: Series system (functions only if all M components function).

Missing values are not permitted in t or stress.

Value

A data frame containing:

time Evaluation time point t .

k System threshold k .

reliability Computed system reliability $R_k(t)$.

unreliability Failure probability $F_k(t) = 1 - R_k(t)$.

References

Yu, C., So, H. Y., and Balakrishnan, N. (2025). *Inference for one-shot devices with Weibull component lifetimes under gamma frailty model*. In: Stochastic Modeling and Statistical Methods, Elsevier Inc., Chapter 1, pp. 1–34. doi:10.1016/B9780443316944.000062.

Ling, M. H., Balakrishnan, N., Yu, C., and So, H. Y. (2021). *Inference for one-shot devices with dependent k -out-of- M structured components under gamma frailty*. Mathematics, 9(23), 3032. doi:10.3390/math9233032.

Examples

```
data(classH)
start_par <- c(-3.997, -0.009, -14.412, 0.003, 0.004, -0.001, -5.493e-4, -0.005, 0.082)
fit_w <- oneshot_frailty(
  data = classH,
  time = "time",
  stress = "stress",
  counts = c("n_none", "n_T", "n_G", "n_both"),
  baseline = "weibull",
  frailty = TRUE,
  start = start_par
)
# Reliability of 1-out-of-2 (parallel) and 2-out-of-2 (series) systems at 400 deg F
reliability_k_out_of_m(fit_w, t = c(1000, 2000, 5000), stress = 400, k = 1)
reliability_k_out_of_m(fit_w, t = c(1000, 2000, 5000), stress = 400, k = 2)
```

reliability_k_out_of_m.oneshot_kofm

Reliability Evaluation for k-Out-of-M Systems

Description

Evaluates the exact system reliability function $R_k(t)$ of k -out-of- M structured one-shot devices under shared gamma frailty at specified operating times and stress levels.

Usage

```
## S3 method for class 'oneshot_kofm'
reliability_k_out_of_m(object, t, stress, k = NULL, ...)
```

Arguments

object	A fitted model object of class "oneshot_kofm".
t	A positive numeric vector of mission times t .
stress	A numeric vector or matrix specifying stress conditions.
k	An optional integer vector specifying system structures ($1 \leq k \leq M$). If NULL (default), evaluates all structures $k = 1, \dots, M$.
...	Additional arguments passed to other methods.

Details

For a k -out-of- M system, at least k components must function for the device to work. Under gamma frailty, the system reliability is given by:

$$R_k(t) = \sum_{n=k}^M \left(\sum_{d=0}^{n-k} (-1)^d \binom{n}{d} \right) \sum_{X \in \mathcal{X}_n} g(X, t)$$

where $\mathcal{X}_n$ is the set of all subsets of Ω of size n , and $g(X, t) = (1 + \beta \sum_{m \in X} (\lambda_m t)^{\alpha_m})^{-1/\beta}$.

Missing values are not permitted.

Value

A data frame containing:

k The system structure threshold (k -out-of- M).

structure Character description ("Parallel (k=1)", "Series (k=M)", or "k-out-of-M").

time Mission time t .

stress Operating stress level.

reliability Model-estimated system reliability $R_k(t)$.

References

Ling, M.-H., Balakrishnan, N., Yu, C., and So, H. Y. (2021). Inference for One-Shot Devices with Dependent k-Out-of-M Structured Components under Gamma Frailty. *Mathematics*, 9(23), 3032. [doi:10.3390/math9233032](https://doi.org/10.3390/math9233032).

Examples

```
data(classH)
fit <- oneshot_kofm(
  data = classH,
  time = "time",
  stress = "stress",
  counts = c("n_none", "n_T", "n_G", "n_both"),
  baseline = "exponential"
)
rel <- reliability_k_out_of_m(fit, t = c(1000, 2000, 5000), stress = 356)
head(rel)
```

residuals.frailty_bayes

Residuals for Frailty Models

Description

Computes Cox-Snell, Martingale, or deviance residuals.

Usage

```
## S3 method for class 'frailty_bayes'
residuals(object, type = c("cox-snell", "martingale", "deviance"), ...)
```

Arguments

object	Fitted frailty model object.
type	Type of residual: "cox-snell", "martingale", or "deviance".
...	Additional arguments.

Value

A numeric vector or matrix of residuals.

residuals.oneshot_kofm

Extract Pearson Residuals

Description

Extracts standardized Pearson residuals from a fitted "oneshot_kofm" object.

Usage

```
## S3 method for class 'oneshot_kofm'
residuals(object, type = c("pearson"), ...)
```

Arguments

object	A fitted model object of class "oneshot_kofm".
type	A character string: "pearson" (default).
...	Additional arguments.

Value

A numeric matrix of standardized Pearson residuals.

retinopathy_frailty

Diabetic Retinopathy Study Paired Eye Data

Description

Bivariate survival times to vision loss in 197 diabetic patients with paired eyes.

Usage

```
retinopathy_frailty
```

Format

A data frame with 197 rows and 9 variables:

id Patient identifier.
time1 Time to visual acuity loss for eye 1.
time2 Time to visual acuity loss for eye 2.
status1 Status indicator for eye 1 (1 = blindness, 0 = censored).
status2 Status indicator for eye 2 (1 = blindness, 0 = censored).
laser Laser treatment type (1 = Xenon, 0 = Argon).
age Age at diabetes diagnosis in years.
risk Risk group indicator.
trt Treatment status (0 = control eye, 1 = treated eye).

Source

Huster, W. J., Brookmeyer, R. and Self, S. G. (1989). Modelling paired survival data with covariates. *Biometrics*, 45: 145-156.

rhr_baseline	<i>Baseline Reversed Hazard Rate Function</i>
--------------	---

Description

Computes the reversed hazard rate (RHR), defined as $r_0(t) = f_0(t)/F_0(t)$.

Usage

```
rhr_baseline(  
  t,  
  dist = c("GE", "EG", "GIR", "MIW", "GLL-I", "GLL-II", "GIE", "Weibull", "Exponential",  
           "Log-Logistic"),  
  params  
)
```

Arguments

- t Numeric vector of positive failure times.
- dist Character string specifying baseline distribution.
- params Numeric named vector of baseline parameters.

Value

Numeric vector of reversed hazard rate values.

Examples

```
rhr_baseline(c(10, 20, 30), dist = "GE", params = c(zeta = 12.4, delta = 0.11))
```

rzicr_promotion	<i>Random Sample Generator for Zero-Inflated Promotion Models</i>
-----------------	---

Description

Alias and wrapper around [sim_ziprom](#) for generating synthetic survival data.

Usage

```
rzicr_promotion(n = 250, ...)
```

Arguments

- n Sample size (integer > 0). Default is 250.
- ... Additional arguments passed to [sim_ziprom](#).

Value

A data frame of simulated observations.

Examples

```
set.seed(42)
d <- rzicr_promotion(n = 50, activation = "minimum")
head(d)
```

sim_add_frailty	<i>Simulate Bivariate Survival Data under Additive Shared Frailty Models</i>
-----------------	--

Description

Generates synthetic bivariate survival data with random right censoring according to the additive shared frailty model specifications described in Pandey et al. (2022) and Tyagi et al. (2024).

Usage

```
sim_add_frailty(
  n = 50,
  beta = 0.15,
  X = NULL,
  frailty_type = c("genlindley", "wlindley", "gamma", "invgaussian"),
  baseline_dist = c("gw", "gll2"),
  frailty_params = NULL,
  baseline_params1 = NULL,
  baseline_params2 = NULL,
  censor_type = c("exponential", "gamma"),
  censor_param = 0.1,
  seed = NULL
)
```

Arguments

- n Number of bivariate observations (subjects/clusters) to generate.
- beta Numeric vector of regression coefficients (e.g. `c(0.15)`).
- X Optional covariate matrix of dimensions `n x length(beta)`. If `NULL`, a single binary covariate is generated from `rbinom(n, size = 1, prob = 0.6)`.

frailty_type	Character string specifying the frailty distribution: "genlindley" (Generalized Lindley), "wlindley" (Weighted Lindley), "gamma" (Gamma), or "invgaussian" (Inverse Gaussian).
baseline_dist	Character string specifying the baseline distribution: "gw" (Generalized Weibull) or "gll2" (Generalized Log-Logistic Type-II).
frailty_params	Named or unnamed numeric vector of frailty parameters: c(theta, mu) for "genlindley", c(s_param) for "wlindley", or c(theta) for "gamma"/"invgaussian".
baseline_params1	Numeric vector of baseline parameters c(zeta1, delta1, xi1) for component 1.
baseline_params2	Numeric vector of baseline parameters c(zeta2, delta2, xi2) for component 2.
censor_type	Character string specifying censoring distribution: "exponential" or "gamma".
censor_param	Numeric parameter for censoring distribution (rate for exponential; rate for gamma).
seed	Optional integer random seed for reproducibility.

Value

A data frame containing:

- id** Subject identifier (1 to n).
- t1** Observed survival time for component 1.
- delta1** Censoring indicator for component 1 (1 = event, 0 = censored).
- t2** Observed survival time for component 2.
- delta2** Censoring indicator for component 2 (1 = event, 0 = censored).
- w** Generated true frailty values.
- covariates** Covariate columns.

References

- Pandey, A., Hanagal, D. D., and Tyagi, S. (2022). Generalised Lindley shared additive frailty regression model for bivariate survival data. *Statistics in Transition new series*, 23(4), 161-176. [doi:10.2478/stattrans20220048](https://doi.org/10.2478/stattrans20220048)
- Tyagi, S., Pandey, A., Hanagal, D. D., and Chesneau, C. (2024). A study on comparisons of additive regression frailty models to counter heterogeneity: Bayesian strategies and case study. *Communications in Statistics - Simulation and Computation*. [doi:10.1080/03610918.2024.2391874](https://doi.org/10.1080/03610918.2024.2391874)

Examples

```
# Simulate 50 observations from Generalized Lindley + GW additive frailty model
sim_data <- sim_add_frailty(n = 50, beta = 0.15, frailty_type = "genlindley",
                           baseline_dist = "gw", frailty_params = c(theta = 2.8, mu = 1.5),
                           baseline_params1 = c(zeta = 22.5, delta = 0.013, xi = 0.35),
                           baseline_params2 = c(zeta = 22.5, delta = 0.013, xi = 0.33),
                           seed = 42)

head(sim_data)
```

sim_biv_rhr_frailty	<i>Simulate Bivariate Left-Censored Survival Data under Shared RHR Frailty</i>
---------------------	--

Description

Generates synthetic bivariate left-censored survival data with covariates and shared frailty based on reversed hazard rates.

Usage

```
sim_biv_rhr_frailty(
  n = 100,
  frailty = c("WL", "GL", "GIG", "Gamma", "IG", "Lindley", "None"),
  baseline = c("GE", "EG", "GIR", "MIW", "GLL-I", "GLL-II", "GIE", "Weibull",
    "Exponential", "Log-Logistic"),
  theta = 3.4,
  mu = NULL,
  theta2 = NULL,
  params1 = c(zeta = 12.4, delta = 0.11),
  params2 = c(zeta = 11.8, delta = 0.11),
  beta = c(0.0075, -0.11, 0.13),
  covars = NULL,
  censor_time1 = 11,
  censor_time2 = 11,
  seed = NULL
)
```

Arguments

n	Sample size (number of pairs). Default is 100.
frailty	Character string specifying frailty distribution. Default is "WL".
baseline	Character string specifying baseline distribution. Default is "GE".
theta	Numeric parameter for frailty distribution. Default is 3.4.
mu	Optional secondary parameter for "GL".
theta2	Optional second parameter for "GIG".
params1	Numeric named vector of baseline parameters for first component. Default is <code>c(zeta = 12.4, delta = 0.11)</code> .
params2	Numeric named vector of baseline parameters for second component. Default is <code>c(zeta = 11.8, delta = 0.11)</code> .
beta	Numeric vector of regression coefficients. Default is <code>c(0.0075, -0.11, 0.13)</code> .
covars	Optional numeric matrix of covariates of dimension $n \times p$. If NULL, covariates are generated automatically.
censor_time1	Numeric scalar or vector of left-censoring thresholds for T1. Default is 11.
censor_time2	Numeric scalar or vector of left-censoring thresholds for T2. Default is 11.
seed	Optional integer seed for reproducibility. Default is NULL.

Details

Simulates the latent frailty variable W_j from the specified frailty distribution, then generates conditionally independent lifetimes T_{1j}, T_{2j} via probability integral transforms on conditional baseline reversed hazard distributions:

$$F(t \mid W_j, X_j) = [F_0(t)]^{W_j \exp(X_j^T \beta)}$$

Left-censoring is applied such that if $T_{ij} \leq C_{ij}$, the recorded time is C_{ij} and $\delta_{ij} = 0$; otherwise $\delta_{ij} = 1$. Missing values (NA or NaN) are not generated.

Value

A data frame containing simulated columns:

- t1 Observed or left-censored lifetime for component 1.
- t2 Observed or left-censored lifetime for component 2.
- d1 Censoring indicator for component 1 (1 = observed, 0 = left-censored).
- d2 Censoring indicator for component 2 (1 = observed, 0 = left-censored).
- covariates** Columns corresponding to the generated or supplied covariates.
- frailty_w The unobserved latent frailty value W_j .

Examples

```
sim_data <- sim_biv_rhr_frailty(n = 50, frailty = "WL", baseline = "GE", seed = 123)
head(sim_data)
```

sim_copula

Simulate Dependent Component Lifetimes via Clayton Copula

Description

Implements the conditional sampling algorithm using the Clayton survival copula (Equations 16–17 of Ling et al., 2021) to generate correlated component failure times under Lomax distributions.

Usage

```
sim_copula(n, lambda, beta, seed = NULL)
```

Arguments

- | | |
|--------|---|
| n | An integer specifying the number of devices to generate. |
| lambda | A positive numeric vector of failure rates λ_m for each component ($M = \text{length}(\lambda)$). |
| beta | A positive numeric scalar representing the frailty variance / copula dependence parameter ($\beta > 0$). |
| seed | An optional integer for reproducibility. |

Details

The algorithm generates $u_1, v_2, \dots, v_M \sim U(0, 1)$ and successively computes:

$$u_2 = \left(1 - (1 - v_2)^{-\frac{\beta}{\beta+1}} u_1^{-\beta} \right)^{-1/\beta}$$

$$u_m = \left(\left(v_m^{-\frac{\beta}{(m-1)\beta+1}} - 1 \right) \sum_{j=1}^{m-1} u_j^{-\beta} - (m-2) v_m^{-\frac{\beta}{(m-1)\beta+1}} + (m-1) \right)^{-1/\beta}$$

and component failure times $t_m = \frac{u_m^{-\beta} - 1}{\beta \lambda_m}$.

Missing values are not permitted.

Value

An $n \times M$ numeric matrix of simulated component failure times, where each row represents a device and each column represents a component.

References

Ling, M.-H., Balakrishnan, N., Yu, C., and So, H. Y. (2021). Inference for One-Shot Devices with Dependent k-Out-of-M Structured Components under Gamma Frailty. *Mathematics*, 9(23), 3032. [doi:10.3390/math9233032](https://doi.org/10.3390/math9233032).

Examples

```
lam <- c(0.0086, 0.0067, 0.0052, 0.0024)
sim_times <- sim_copula(n = 100, lambda = lam, beta = 0.3, seed = 123)
head(sim_times)
```

sim_corrFrailty	<i>Simulate Bivariate Survival Data with Correlated Frailty</i>
-----------------	---

Description

Generates synthetic bivariate right-censored survival data from correlated frailty models with user-specified frailty families, baseline distributions, regression coefficients, and censoring rates.

Usage

```
sim_corrFrailty(
  n = 100,
  frailty = c("gamma", "invgaussian", "posstable", "comppoisson", "none"),
  frailty.params = list(sigma = 0.5, rho = 0.5),
  baseline = c("genweibull", "genloglogistic", "weibull", "exponential", "pareto", "lfr",
    "burr"),
  baseline.params = list(lambda = 0.05, gamma = 1.2),
  beta = c(0.5, -0.3),
```

```

    censor.rate = 0.2,
    seed = NULL
  )

```

Arguments

n	Integer, sample size (number of pairs/clusters). Default is 100.
frailty	Character, frailty distribution. One of "gamma", "invgaussian", "posstable", "comppoisson", or "none".
frailty.params	List of frailty parameters. E.g., for "gamma" or "invgaussian": list(sigma = 0.5, rho = 0.5); for "posstable": list(theta = 0.7, rho = 0.5); for "comppoisson": list(eta = 0.5, nu = 1.0, theta = 1.0).
baseline	Character, baseline distribution for T1 and T2. One of "genweibull", "genloglogistic", "weibull", "exponential", "pareto", "lfr", or "burr".
baseline.params	List of baseline parameters. E.g., for "weibull": list(lambda = 0.05, gamma = 1.2); for "genweibull" or "genloglogistic": list(alpha = 1.5, lambda = 0.05, gamma = 1.2).
beta	Numeric vector of regression coefficients. Default is c(0.5, -0.3).
censor.rate	Numeric between 0 and 1, target censoring proportion. Default is 0.2.
seed	Optional integer for reproducibility.

Value

A data.frame containing:

id	Integer cluster / pair identifier (1 to n).
t1	Observed time for first component.
d1	Censoring indicator for first component (1 = event, 0 = censored).
t2	Observed time for second component.
d2	Censoring indicator for second component (1 = event, 0 = censored).
x1, x2, ...	Simulated covariates.

Author(s)

Shikhar Tyagi, Arvind Pandey, Vrijesh Tripathi

References

- Hanagal, D. D., Pandey, A., and Ganguly, A. (2017). Correlated gamma frailty models for bivariate survival data. *Communications in Statistics - Simulation and Computation*, 46(5), 3627-3644. <doi:10.1080/03610918.2015.1085559>
- Hanagal, D. D. and Pandey, A. (2018). Correlated inverse Gaussian frailty models for bivariate survival data. *Communications in Statistics - Theory and Methods*, 49(14), 3455-3475. <doi:10.1080/03610926.2018.1549256>

Examples

```

dat <- sim_corrFrailty(n = 50, frailty = "gamma",
                      frailty.params = list(sigma = 0.5, rho = 0.4),
                      baseline = "weibull",
                      baseline.params = list(lambda = 0.05, gamma = 1.2),
                      beta = c(0.4, -0.3), seed = 123)

head(dat)

```

sim_data4

*Simulated One-Shot Device Test Data with 4 Failure Modes***Description**

A simulated one-shot device testing dataset with $M = 4$ failure modes across 6 test groups (Table 15 of Ling et al., 2021). Evaluated at three stress levels ($s_i = 35, 45, 55$) and two inspection times ($\tau_i = 10, 20$) with $K_i = 100$ devices per group.

Usage

```
sim_data4
```

Format

A data frame with 6 rows and 19 variables:

group Test group index ($i = 1, \dots, 6$).

stress Accelerated stress level ($s_i = 35, 45, 55$).

time Inspection time ($\tau_i = 10, 20$).

n_none Number of devices with no malfunction ($X = \emptyset$).

n_1 Devices with only Component 1 failed.

n_2 Devices with only Component 2 failed.

n_3 Devices with only Component 3 failed.

n_4 Devices with only Component 4 failed.

n_12 Devices with Components 1 and 2 failed.

n_13 Devices with Components 1 and 3 failed.

n_14 Devices with Components 1 and 4 failed.

n_23 Devices with Components 2 and 3 failed.

n_24 Devices with Components 2 and 4 failed.

n_34 Devices with Components 3 and 4 failed.

n_123 Devices with Components 1, 2, and 3 failed.

n_124 Devices with Components 1, 2, and 4 failed.

n_134 Devices with Components 1, 3, and 4 failed.

n_234 Devices with Components 2, 3, and 4 failed.

n_all Devices with all 4 components failed ($X = \Omega$).

Source

Ling, M.-H., Balakrishnan, N., Yu, C., and So, H. Y. (2021). Inference for One-Shot Devices with Dependent k-Out-of-M Structured Components under Gamma Frailty. *Mathematics*, 9(23), 3032. [doi:10.3390/math9233032](https://doi.org/10.3390/math9233032).

Examples

```
data(sim_data4)
head(sim_data4)
```

sim_frailty	<i>Simulate Frailty Survival Data or One-Shot Device Lifetimes</i>
-------------	--

Description

Simulates synthetic survival datasets under shared, additive, or multiplicative frailty models, or simulates correlated component lifetimes for one-shot devices under shared gamma frailty.

Usage

```
sim_frailty(
  n = 100,
  baseline = "weibull",
  frailty = "gamma",
  type = "multiplicative_bivariate",
  beta = c(0.5, -0.5),
  theta = 1,
  baseline_params = c(0.01, 1.2),
  censor_rate = 0.2,
  lambda = NULL,
  alpha = NULL,
  seed = NULL
)
```

Arguments

n	An integer specifying sample size (number of subjects, clusters, or devices). Default is 100.
baseline	Baseline hazard distribution name: "weibull" (default), "exponential", "exppower", etc.
frailty	Frailty distribution name: "gamma" (default), etc.
type	Model type: "multiplicative_bivariate" (default), "additive_bivariate", "multiplicative_univariate", or "additive_univariate".
beta	Regression coefficients vector or frailty variance parameter (for one-shot devices).

theta	Frailty parameter(s).
baseline_params	Baseline hazard parameter vector.
censor_rate	Proportion of right-censored observations (default 0.2).
lambda	Optional positive numeric vector of component failure rates λ_m (for one-shot device testing simulation). When supplied, one-shot device lifetimes are simulated.
alpha	Optional positive numeric vector of Weibull shape parameters (for one-shot devices).
seed	Optional integer for reproducibility.

Value

If lambda is supplied, an $n \times M$ numeric matrix of simulated one-shot component lifetimes. Otherwise, a data frame containing simulated survival times, status indicators, covariates, and true frailty values. Missing values are not generated.

Examples

```
# Simulate survival data
sim_dat <- sim_frailty(n = 50, baseline = "weibull", frailty = "gamma")
head(sim_dat)

# Simulate one-shot device lifetimes
lam <- c(0.0086, 0.0067, 0.0052, 0.0024)
sim_times <- sim_frailty(n = 50, lambda = lam, beta = 0.3, seed = 123)
head(sim_times)
```

sim_oneshot	<i>Simulate One-Shot Device Testing Datasets</i>
-------------	--

Description

Generates synthetic one-shot device testing data frames across multiple test groups under constant-stress accelerated life-testing plans with dependent components.

Usage

```
sim_oneshot(
  n_groups = 4,
  K = 50,
  inspection_times = c(1000, 2000, 3000, 4000),
  stress_levels = c(25, 35, 45, 55),
  a_params,
  b_params = NULL,
  beta = 0.2,
  baseline = c("exponential", "weibull"),
```

```

    method = c("frailty", "copula"),
    seed = NULL
  )

```

Arguments

n_groups	An integer specifying the number of test groups I .
K	An integer scalar or vector of length I specifying sample sizes per group.
inspection_times	A positive numeric vector of inspection times τ_i .
stress_levels	A numeric vector or matrix of stress levels s_i .
a_params	An $M \times p$ matrix of scale regression coefficients.
b_params	An optional $M \times p$ matrix of shape regression coefficients (for Weibull).
beta	A positive numeric scalar representing the frailty variance ($\beta > 0$).
baseline	A character string: "exponential" (default) or "weibull".
method	A character string: "frailty" (default) or "copula".
seed	An optional integer for reproducibility.

Details

Missing values are not permitted in any arguments.

Value

A data frame containing:

group Test group index i .

stress Stress level s_i .

time Inspection time τ_i .

counts Observed counts for each of the 2^M malfunction combinations.

References

Ling, M.-H., Balakrishnan, N., Yu, C., and So, H. Y. (2021). Inference for One-Shot Devices with Dependent k-Out-of-M Structured Components under Gamma Frailty. *Mathematics*, 9(23), 3032. [doi:10.3390/math9233032](https://doi.org/10.3390/math9233032).

Examples

```

a_par <- matrix(c(-6.0, 0.05, -6.5, 0.06), nrow = 2, byrow = TRUE)
sim_df <- sim_oneshot(
  n_groups = 4,
  K = 50,
  inspection_times = c(10, 20, 10, 20),
  stress_levels = c(35, 35, 45, 45),
  a_params = a_par,
  beta = 0.2,

```

```

    seed = 42
  )
  head(sim_df)

```

sim_oneshot_frailty	<i>Simulate One-Shot Device Lifetimes under Shared Gamma Frailty</i>
---------------------	--

Description

Convenience alias to generate correlated component lifetimes for one-shot devices under gamma frailty.

Usage

```
sim_oneshot_frailty(n, lambda, beta, alpha = NULL, seed = NULL)
```

Arguments

n	An integer specifying the number of devices to generate.
lambda	A positive numeric vector of failure rates λ_m for each component.
beta	A positive numeric scalar representing the frailty variance ($\beta > 0$).
alpha	An optional positive numeric vector of Weibull shape parameters.
seed	An optional integer for reproducibility.

Value

An $n \times M$ numeric matrix of simulated component failure times. Missing values are not permitted.

sim_oneshot_frailty_lifetimes	<i>Simulate Dependent Component Lifetimes via Gamma Frailty</i>
-------------------------------	---

Description

Generates correlated component lifetimes by first generating a latent gamma frailty $\gamma \sim \text{Gamma}(1/\beta, \beta)$ and then simulating conditionally independent Exponential or Weibull component lifetimes.

Usage

```
sim_oneshot_frailty_lifetimes(n, lambda, beta, alpha = NULL, seed = NULL)
```

Arguments

n	An integer specifying the number of devices to generate.
lambda	A positive numeric vector of failure rates λ_m for each component ($M = \text{length}(\lambda)$).
beta	A positive numeric scalar representing the frailty variance ($\beta > 0$).
alpha	An optional positive numeric vector of Weibull shape parameters (default 1 for exponential).
seed	An optional integer for reproducibility.

Details

Conditioned on $\gamma_{i,j} \sim \text{Gamma}(\text{shape} = 1/\beta, \text{scale} = \beta)$, the component lifetimes are independent with:

$$T_{i,j,m} = \frac{1}{\lambda_m} \left(-\frac{\ln(1 - u_m)}{\gamma_{i,j}} \right)^{1/\alpha_m}$$

where $u_m \sim U(0, 1)$.

Missing values are not permitted.

Value

An $n \times M$ numeric matrix of simulated component failure times.

References

Ling, M.-H., Balakrishnan, N., Yu, C., and So, H. Y. (2021). Inference for One-Shot Devices with Dependent k-Out-of-M Structured Components under Gamma Frailty. *Mathematics*, 9(23), 3032. [doi:10.3390/math9233032](https://doi.org/10.3390/math9233032).

Examples

```
lam <- c(0.0086, 0.0067, 0.0052, 0.0024)
sim_times <- sim_frailty(n = 100, lambda = lam, beta = 0.3, seed = 123)
head(sim_times)
```

sim_ziprom

Simulate Data from Zero-Inflated Promotion Cure-Rate Models

Description

Generates synthetic survival datasets under the zero-adjusted promotion cure-rate model with Gamma frailty and flexible activation schemes (Algorithm 1 of Assuncao et al., 2026).

Usage

```
sim_ziprom(
  n = 250,
  beta1 = c(-2, 0.5),
  beta2 = c(-1, 0.75),
  beta3 = c(0.5, 0.5),
  beta4 = c(1.5, 2),
  alpha = 0.25,
  activation = c("random", "minimum", "maximum"),
  baseline = c("weibull", "gompertz", "exponential"),
  censor_rate = 0.2,
  seed = NULL
)
```

Arguments

n	Sample size (integer > 0). Default is 250.
beta1	Numeric vector of coefficients for zero-inflation linear predictor β_1 .
beta2	Numeric vector of coefficients for cure fraction linear predictor β_2 .
beta3	Numeric vector of coefficients for shape linear predictor β_3 .
beta4	Numeric vector of coefficients for scale linear predictor β_4 .
alpha	Numeric scalar for Gamma frailty variance ($\alpha \geq 0$). Default is 0.25.
activation	Character string for activation scheme: "random" (default), "minimum", or "maximum".
baseline	Character string for baseline activation distribution: "weibull" (default), "gompertz", or "exponential".
censor_rate	Nominal right-censoring rate in (0, 1). Default is 0.20.
seed	Optional integer random seed for reproducibility.

Value

A data frame containing:

- time Observed survival times ($\min(s_i, w_i)$).
- status Censoring indicator (1 = event, 0 = right-censored).
- x Simulated binary covariate ($x_i \sim \text{Bernoulli}(0.5)$).

References

Assuncao, D., Ramos, P. L., Tomazella, V., Leao, J., Osses, L., and Rodrigues, A. (2026). Zero-Inflated Promotion Cure-Rate Models With Frailty and Flexible Activation Schemes. *Biometrical Journal*, 68:e70162. doi:[10.1002/bimj.70162](https://doi.org/10.1002/bimj.70162)

Examples

```
set.seed(123)
dat <- sim_ziprom(n = 100, activation = "random")
head(dat)
table(dat$time == 0) # Instantaneous zero failures
table(dat$status)    # Events vs censored
```

summary.frailty_bayes	<i>Summary Method for Bayesian Frailty Models</i>
-----------------------	---

Description

Summary Method for Bayesian Frailty Models

Usage

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

Arguments

object Object of class "frailty_bayes".
... Additional arguments.

Value

An object of class "summary.frailty_bayes".

summary.frailty_mle	<i>Summary Method for MLE Frailty Models</i>
---------------------	--

Description

Summary Method for MLE Frailty Models

Usage

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

Arguments

object Object of class "frailty_mle".
... Additional arguments.

Value

An object of class "summary.frailty_mle".

summary.oneshot_kofm *Summary Method for One-Shot Frailty Model Fits*

Description

Computes and formats comprehensive estimation summaries, standard errors, z -values, p -values, and confidence intervals for "oneshot_kofm" objects.

Usage

```
## S3 method for class 'oneshot_kofm'
summary(object, ci_level = object$ci_level, ...)
```

Arguments

object	A fitted model object of class "oneshot_kofm".
ci_level	A numeric scalar between 0 and 1 specifying the confidence level (default from object).
...	Additional arguments.

Value

An S3 object of class "summary.oneshot_kofm" containing formatted tables and model evaluation statistics.

surv_proper *Proper Conditional Survival Function for Susceptible Group*

Description

Computes $S_p^*(t) = P(T > t \mid M \geq 1)$ under Minimum, Maximum, and Random activation schemes with Gamma frailty.

Usage

```
surv_proper(
  t,
  theta,
  alpha,
  shape,
  scale,
  activation = c("random", "minimum", "maximum"),
  baseline = c("weibull", "gompertz", "exponential")
)
```

Arguments

t	Numeric vector of positive survival times.
theta	Numeric vector or scalar representing the latent Poisson parameter (> 0).
alpha	Numeric scalar representing Gamma frailty variance (≥ 0).
shape	Numeric vector or scalar representing baseline shape parameter(s) (> 0).
scale	Numeric vector or scalar representing baseline scale parameter(s) (> 0).
activation	Character string indicating the activation scheme: "random" (default), "minimum", or "maximum".
baseline	Character string indicating the baseline hazard distribution: "weibull" (default), "gompertz", or "exponential".

Value

Numeric vector of proper conditional survival probabilities.

test_baseline	<i>Likelihood Ratio Test for Baseline Lifetime Distribution</i>
---------------	---

Description

Performs the Likelihood Ratio Test comparing Weibull versus Exponential baseline lifetime distributions.

Usage

```
test_baseline(object, ...)
```

Arguments

object	A fitted Weibull model object of class "oneshot_kofm" or "oneshot_frailty".
...	Additional arguments passed to methods (such as fit_exp for "oneshot_frailty").

Value

An S3 object of class "htest" containing the test statistic, degrees of freedom, and p-value.

Examples

```
data(classH)
fit_w <- oneshot_kofm(
  data = classH,
  time = "time",
  stress = "stress",
  counts = c("n_none", "n_T", "n_G", "n_both"),
  baseline = "weibull",
  frailty = TRUE
)
test_baseline(fit_w)
```

test_independence	<i>Likelihood Ratio Test for Component Independence</i>
-------------------	---

Description

Performs the Likelihood Ratio Test (LRT) to evaluate the null hypothesis of component independence against the shared gamma frailty alternative.

Usage

```
test_independence(object, ...)
```

Arguments

object	A fitted model object of class "oneshot_kofm" or "oneshot_frailty".
...	Additional arguments passed to methods (such as fit_indep for "oneshot_frailty").

Value

An S3 object of class "htest" containing the test statistic, degrees of freedom, and p-value.

Examples

```
data(classH)
fit_w <- oneshot_kofm(
  data = classH,
  time = "time",
  stress = "stress",
  counts = c("n_none", "n_T", "n_G", "n_both"),
  baseline = "exponential",
  frailty = TRUE
)
test_independence(fit_w)
```

var_frailty	<i>Variance of Frailty Distribution</i>
-------------	---

Description

Computes the theoretical variance of the frailty distribution under the identifiability condition $E[W] = 1$.

Usage

```
var_frailty(
  frailty = c("WL", "GL", "GIG", "Gamma", "IG", "Lindley", "None"),
  theta,
  mu = NULL,
  theta2 = NULL
)
```

Arguments

frailty	Character string indicating frailty distribution.
theta	Numeric parameter for frailty distribution.
mu	Optional secondary parameter for "GL".
theta2	Optional second parameter for "GIG".

Value

Numeric scalar representing variance of frailty variable.

Examples

```
var_frailty("WL", theta = 3.4)
var_frailty("Gamma", theta = 2.0)
```

vcov.oneshot_kofm	<i>Extract Variance-Covariance Matrix</i>
-------------------	---

Description

Extracts the asymptotic variance-covariance matrix of parameter estimators from a fitted "oneshot_kofm" object.

Usage

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

Arguments

object	A fitted model object of class "oneshot_kofm".
...	Additional arguments.

Value

A symmetric numeric variance-covariance matrix.

ziprom_core_dens	<i>Population Density Function</i>
------------------	------------------------------------

Description

Computes the population density function $f_p(t)$ for positive survival times $t > 0$.

Usage

```
ziprom_core_dens(
  t,
  p0,
  p1,
  alpha = 0,
  shape = 1,
  scale = 1,
  activation = c("random", "minimum", "maximum"),
  baseline = c("weibull", "gompertz", "exponential")
)
```

Arguments

t	Numeric vector of positive times ($t > 0$).
p0	Numeric vector or scalar of zero-failure probabilities.
p1	Numeric vector or scalar of cure fractions.
alpha	Numeric scalar representing Gamma frailty variance (≥ 0).
shape	Numeric vector or scalar for shape parameter.
scale	Numeric vector or scalar for scale parameter.
activation	Character string for activation scheme ("random", "minimum", or "maximum").
baseline	Character string for baseline hazard distribution ("weibull", "gompertz", or "exponential").

Value

Numeric vector of population density values for $t > 0$.

Examples

```
t_seq <- seq(0.1, 10, by = 1)
d_vals <- ziprom_core_dens(t_seq, p0 = 0.1, p1 = 0.4, alpha = 0.25,
                           shape = 1.2, scale = 2.5, activation = "random")
print(d_vals)
```

ziprom_core_surv	<i>Population Survival Function</i>
------------------	-------------------------------------

Description

Computes the unconditional population survival function $S_p(t)$.

Usage

```
ziprom_core_surv(
  t,
  p0,
  p1,
  alpha = 0,
  shape = 1,
  scale = 1,
  activation = c("random", "minimum", "maximum"),
  baseline = c("weibull", "gompertz", "exponential")
)
```

Arguments

t	Numeric vector of non-negative times.
p0	Numeric vector or scalar of zero-failure probabilities.
p1	Numeric vector or scalar of cure fractions.
alpha	Numeric scalar representing Gamma frailty variance (≥ 0).
shape	Numeric vector or scalar for shape parameter.
scale	Numeric vector or scalar for scale parameter.
activation	Character string for activation scheme ("random", "minimum", or "maximum").
baseline	Character string for baseline hazard distribution ("weibull", "gompertz", or "exponential").

Value

Numeric vector of population survival probabilities.

Examples

```
t_seq <- seq(0, 10, by = 1)
s_vals <- ziprom_core_surv(t_seq, p0 = 0.1, p1 = 0.4, alpha = 0.25,
                          shape = 1.2, scale = 2.5, activation = "random")
print(s_vals)
```

ziprom_fit

*Fit Zero-Inflated Promotion Cure-Rate Models with Frailty***Description**

Fits generalized zero-inflated promotion cure-rate regression models with Gamma frailty and flexible activation schemes (minimum, maximum, or random) using Maximum Likelihood Estimation (MLE).

Usage

```
ziprom_fit(
  formula = NULL,
  data = NULL,
  formula_p0 = NULL,
  formula_p1 = NULL,
  formula_shape = NULL,
  formula_scale = NULL,
  time = NULL,
  status = NULL,
  activation = c("random", "minimum", "maximum"),
  baseline = c("weibull", "gompertz", "exponential"),
  frailty = TRUE,
  init = NULL,
  control = list()
)
```

Arguments

formula	An object of class Surv or formula , e.g. <code>survival::Surv(time, status) ~ x1 + x2</code> . If specified, the same linear predictor is used for components unless separate formulas are supplied.
data	A data frame containing the variables named in the formulas. Missing values (NA) in the response or covariates are automatically removed prior to model fitting via listwise deletion (na.omit).
formula_p0	Optional one-sided formula for the zero-inflation component p_0 , e.g. $\sim x1$.
formula_p1	Optional one-sided formula for the cure fraction component p_1 , e.g. $\sim x2$.
formula_shape	Optional one-sided formula for the baseline shape parameter α_1 , e.g. ~ 1 .
formula_scale	Optional one-sided formula for the baseline scale parameter α_2 , e.g. ~ 1 .
time	Optional numeric vector of observed survival times (used if <code>formula</code> is not provided).
status	Optional numeric vector of event indicators (1 = event, 0 = censored).
activation	Character string specifying the activation scheme: "random" (default), "minimum", or "maximum".

baseline	Character string specifying the baseline activation-time distribution: "weibull" (default), "gompertz", or "exponential".
frailty	Logical value (TRUE or FALSE) indicating whether Gamma frailty should be included in the model. Default is TRUE. If FALSE, fits the no-frailty model.
init	Optional numeric vector of starting values for the optimization.
control	Optional list of control parameters passed to <code>optim</code> .

Details

Missing values (NA) in the response or covariates are omitted prior to estimation. Observations with survival time equal to zero ($t = 0$) represent instantaneous failures and are naturally accommodated via the zero-inflation probability mass p_0 .

Value

An S3 object of class "ziprom_fit" containing:

`coefficients` Vector of maximum likelihood estimates for all parameters.

`coef_list` List of estimated coefficients partitioned by component (beta1, beta2, beta3, beta4, alpha).

`se` Standard errors of parameter estimates obtained from the inverted observed Hessian.

`z_values` Wald z-test statistics ($\hat{\beta}/SE$).

`p_values` Two-sided asymptotic p-values for Wald tests.

`ci_90` 90% Wald confidence intervals for parameters.

`ci_95` 95% Wald confidence intervals for parameters.

`ci_99` 99% Wald confidence intervals for parameters.

`vcov` Estimated asymptotic variance-covariance matrix.

`hessian` Observed information / Hessian matrix at the optimum.

`loglik` Maximum log-likelihood value.

`npars` Number of free estimated parameters.

`aic` Akaike Information Criterion ($2k - 2\ell$).

`bic` Bayesian Information Criterion ($k \log(n) - 2\ell$).

`hqic` Hannan-Quinn Information Criterion.

`caic` Consistent Akaike Information Criterion.

`p0_mean` Estimated zero-failure proportion at mean covariates.

`p0_mean_se` Delta-method standard error for mean p_0 .

`p1_mean` Estimated cure fraction at mean covariates.

`p1_mean_se` Delta-method standard error for mean p_1 .

`p0_ci_95` 95% confidence interval for mean p_0 .

`p1_ci_95` 95% confidence interval for mean p_1 .

`fitted_p0` Fitted zero-inflation probabilities for each subject.

- fitted_p1 Fitted cure probabilities for each subject.
- residuals Normalized randomized quantile residuals.
- n Total sample size.
- n_zeros Number of instantaneous zero failures ($t = 0$).
- n_events Number of positive observed events ($t > 0, \delta = 1$).
- n_censored Number of right-censored observations ($t > 0, \delta = 0$).
- activation Activation scheme used.
- baseline Baseline hazard distribution used.
- frailty Logical indicating whether frailty was modeled.
- convergence Convergence code returned by optim (0 indicates success).
- data Data frame containing the modeled variables.
- call The matched function call.

References

Assuncao, D., Ramos, P. L., Tomazella, V., Leao, J., Osses, L., and Rodrigues, A. (2026). Zero-Inflated Promotion Cure-Rate Models With Frailty and Flexible Activation Schemes. *Biometrical Journal*, 68:e70162. doi:10.1002/bimj.70162

Examples

```
set.seed(42)
sim_dat <- sim_ziprom(n = 80, beta1 = c(-1.5, 0.4), beta2 = c(-0.8, 0.6),
                    beta3 = c(0.2, 0.1), beta4 = c(1.0, 0.5), alpha = 0.25,
                    activation = "random", baseline = "weibull")
fit <- ziprom_fit(survival::Surv(time, status) ~ x, data = sim_dat,
                 activation = "random", frailty = TRUE)
summary(fit)
```

ziprom_loglik	Log-Likelihood Function for Zero-Inflated Promotion Cure-Rate Models
---------------	--

Description

Computes the exact log-likelihood for the zero-adjusted promotion cure-rate model with Gamma frailty and activation schemes under random right censoring.

Usage

```
ziprom_loglik(
  par,
  time,
  status,
  X1,
  X2,
  X3,
  X4,
  activation = c("random", "minimum", "maximum"),
  baseline = c("weibull", "gompertz", "exponential"),
  frailty = TRUE
)
```

Arguments

par	Numeric vector of parameters: $c(\text{beta1}, \text{beta2}, \text{beta3}, \text{beta4}, \text{log_alpha})$ where beta1 is for zero inflation, beta2 is for cure fraction, beta3 is for baseline shape, beta4 is for baseline scale, and log_alpha is $\log(\text{frailty variance})$ if frailty = TRUE.
time	Numeric vector of non-negative observed survival times.
status	Numeric vector of event indicators (1 = event, 0 = censored).
X1	Design matrix for zero-inflation probability p_0 .
X2	Design matrix for cure probability p_1 .
X3	Design matrix for baseline shape parameter α_1 .
X4	Design matrix for baseline scale parameter α_2 .
activation	Character string: "random", "minimum", or "maximum".
baseline	Character string: "weibull", "gompertz", or "exponential".
frailty	Logical indicating whether Gamma frailty is included (TRUE or FALSE).

Value

Scalar value of the negative log-likelihood (for minimization in `optim`), or the positive log-likelihood if called with appropriate sign.

References

Assuncao, D., Ramos, P. L., Tomazella, V., Leao, J., Osses, L., and Rodrigues, A. (2026). Zero-Inflated Promotion Cure-Rate Models With Frailty and Flexible Activation Schemes. *Biometrical Journal*, 68:e70162. doi:[10.1002/bimj.70162](https://doi.org/10.1002/bimj.70162)

Examples

```
set.seed(123)
dat <- sim_ziprom(n = 50, activation = "random")
X <- matrix(1, nrow = nrow(dat), ncol = 1)
```

```
par_init <- c(-0.5, 1.0, 0.0, 1.0, log(0.25))
nll <- ziprom_loglik(par_init, dat$time, dat$status, X, X, X, X,
                    activation = "random", baseline = "weibull", frailty = TRUE)
print(nll)
```

Index

* datasets

- austrian_twin, 5
- classH, 10
- insulin_gdm, 57
- kidney_bivariate, 59
- kidney_frailty, 60
- kidney_infection, 61
- mice_tumor, 68
- retinopathy_frailty, 99
- sim_data4, 107

AIC.ziprom_fit (print.ziprom_fit), 92

austrian_twin, 5

base_cumhaz, 6

base_haz, 7

bayes_factor, 7

bayes_factor_rhr, 8

BIC.ziprom_fit (print.ziprom_fit), 92

calc_bayes_factor (criteria), 25

calc_loss_estimates, 9

calc_model_criteria (criteria), 25

classH, 10

coef.oneshot_kofm, 11

coef.ziprom_fit (print.ziprom_fit), 92

compare_rhr_models, 11

component_moments, 12

component_moments.oneshot_frailty, 13

component_moments.oneshot_kofm, 14

compute_mcmc_diagnostics, 15

compute_p0_p1, 16

compute_probabilities, 17

cond_prob_rhr, 18

confint.oneshot_kofm, 19

confint.ziprom_fit (print.ziprom_fit), 92

corrFrailty, 20, 22

corrFrailty.control, 21, 22

cpo, 23

crhr_baseline, 24

criteria, 25

cross_ratio_rhr, 26

d2laplace_frailty, 28

d_rhr_baseline, 27

dens_proper, 28

dependence, 29

diagnostics_mcmc, 30

dic, 31

dlaplace_frailty, 32

eaic, 33

ebic, 33

em_oneshot_frailty, 34

em_oneshot_kofm, 35

evaluate_ziprom, 37

fb_sim_frailty, 38

fisher_info_oneshot, 39

fit_add_frailty_bayes, 41

fit_add_frailty_mle, 43

fit_rhr_frailty_bayes, 45

fit_rhr_frailty_mle, 48

fitted.oneshot_kofm, 49

forecast_ziprom, 50

formula, 121

frailty_bayes, 51, 54

frailty_compare, 53

frailty_fit, 54

frailty_mle, 54, 54

frailty_variance (dependence), 29

generate_powerset, 17, 56

hpd_interval, 56

insulin_gdm, 57

kendall_tau (dependence), 29

kendall_tau_rhr, 58

kidney_bivariate, 59
 kidney_frailty, 60
 kidney_infection, 61

 laplace_frailty, 62
 logLik.frailty_bayes, 63
 logLik.oneshot_kofm, 63
 logLik.ziprom_fit (print.ziprom_fit), 92
 loglik_add_frailty, 64
 loglik_biv_rhr, 65
 loglik_oneshot, 66
 lpml, 68

 mice_tumor, 68
 mttf_k_out_of_m, 69
 mttf_k_out_of_m.oneshot_frailty, 69
 mttf_k_out_of_m.oneshot_kofm, 71

 na.omit, 121

 oneshot_frailty, 73
 oneshot_kofm, 75
 optim, 23, 48, 49, 122

 p_rhr_baseline, 78
 plot.add_frailty, 78
 plot.frailty_bayes, 80
 plot.oneshot_frailty, 80
 plot.oneshot_kofm, 82
 plot.rhr_bayes_fit, 83
 plot.rhr_mle_fit, 84
 plot.ziprom_fit, 84
 plot_activation_compare, 86
 plot_forecast (plot.ziprom_fit), 84
 plot_quantile_residuals
 (plot.ziprom_fit), 84
 plot_survival_fit (plot.ziprom_fit), 84
 predict.frailty_bayes, 87
 predict.oneshot_kofm, 88
 predict.ziprom_fit, 50, 88
 print.frailty_bayes, 90
 print.oneshot_kofm, 91
 print.summary.oneshot_kofm, 91
 print.summary.ziprom_fit
 (print.ziprom_fit), 92
 print.ziprom_fit, 92

 q_rhr_baseline, 93
 quantile_proper, 94

 reliability_k_out_of_m, 95
 reliability_k_out_of_m.oneshot_frailty,
 95
 reliability_k_out_of_m.oneshot_kofm,
 97
 residuals.frailty_bayes, 98
 residuals.oneshot_kofm, 99
 residuals.ziprom_fit
 (print.ziprom_fit), 92
 retinopathy_frailty, 99
 rhr_baseline, 100
 rzicr_promotion, 100

 sim_add_frailty, 101
 sim_biv_rhr_frailty, 103
 sim_copula, 104
 sim_corrFrailty, 105
 sim_data4, 107
 sim_frailty, 108
 sim_oneshot, 109
 sim_oneshot_frailty, 111
 sim_oneshot_frailty_lifetimes, 111
 sim_ziprom, 100, 101, 112
 summary.frailty_bayes, 114
 summary.frailty_mle, 114
 summary.oneshot_kofm, 115
 summary.ziprom_fit (print.ziprom_fit),
 92
 Surv, 121
 surv_proper, 115

 test_baseline, 116
 test_independence, 117

 var_frailty, 117
 vcov.oneshot_kofm, 118
 vcov.ziprom_fit (print.ziprom_fit), 92

 ziprom_core_dens, 119
 ziprom_core_surv, 120
 ziprom_fit, 121
 ziprom_loglik, 123
 ziprom_residuals (evaluate_ziprom), 37