Package ‘maskedcauses’

May 8, 2026

Title Likelihood Models for Systems with Masked Component Cause of Failure

Version 0.9.3

Description Maximum likelihood estimation for series systems where the component cause of failure is masked. Implements analytical log-likelihood, score, and Hessian functions for exponential, homogeneous Weibull, and heterogeneous Weibull component lifetimes under masked cause conditions (C1, C2, C3). Supports exact, right-censored, left-censored, and interval-censored observations via composable observation functors. Provides random data generation, model fitting, and Fisher information for asymptotic inference. See Lin, Loh, and Bai (1993) <doi:10.1109/24.257799> and Craiu and Reiser (2006) <doi:10.1111/j.1541-0420.2005.00498.x>.

License GPL (>= 3)

Language en-US

Encoding UTF-8

RoxygenNote 7.3.3

Suggests knitr, rmarkdown, testthat (>= 3.0.0), algebraic.mle, algebraic.dist

Config/testthat/edition 3

Imports stats, likelihood.model, generics, numDeriv

Depends R (>= 4.1.0)

VignetteBuilder knitr

URL `https://queelius.github.io/maskedcauses/`

BugReports `https://github.com/queelius/maskedcauses/issues`

NeedsCompilation no

Author Alexander Towell [aut, cre] (ORCID: <`https://orcid.org/0000-0001-6443-9897`>)

Maintainer Alexander Towell <`lex@metafunctor.com`>

Repository CRAN

Date/Publication 2026-03-17 19:00:03 UTC

Contents

`assumptions.exp_series_md_c1_c2_c3`

Assumptions for `exp_series_md_c1_c2_c3` *model.*

Description

Returns the assumptions made by this likelihood model.

Usage

```
## S3 method for class 'exp_series_md_c1_c2_c3'
assumptions(model, ...)
```

Arguments

`model`	the likelihood model object
`...`	additional arguments (ignored)

Value

character vector of assumptions

Examples

```
assumptions(exp_series_md_c1_c2_c3())
```

`assumptions.wei_series_homogeneous_md_c1_c2_c3`

Assumptions for `wei_series_homogeneous_md_c1_c2_c3` *model.*

Description

Returns the assumptions made by this likelihood model.

Usage

```
## S3 method for class 'wei_series_homogeneous_md_c1_c2_c3'
assumptions(model, ...)
```

Arguments

`model`	the likelihood model object
`...`	additional arguments (ignored)

Value

character vector of assumptions

Examples

```
assumptions(wei_series_homogeneous_md_c1_c2_c3())
```

`assumptions.wei_series_md_c1_c2_c3` *Assumptions for* `wei_series_md_c1_c2_c3` *model.*

Description

Returns the assumptions made by this likelihood model.

Usage

```
## S3 method for class 'wei_series_md_c1_c2_c3'
assumptions(model, ...)
```

Arguments

`model`	the likelihood model object
`...`	additional arguments (ignored)

Value

character vector of assumptions

Examples

```
assumptions(wei_series_md_c1_c2_c3())
```

`cause_probability` *Marginal cause-of-failure probability*

Description

Returns a closure computing P(K=j | theta) for all components j, marginalized over the system failure time T. By Theorem 5 of the foundational paper, this equals E_T[P(K=j | T, theta)].

Usage

```
cause_probability(model, ...)

## S3 method for class 'series_md'
cause_probability(model, ...)
```

Arguments

`model`	a likelihood model object
`...`	additional arguments passed to the returned closure

Details

The default method uses Monte Carlo integration via `rdata()`.

Value

a function with signature `function(par, ...)` returning an m-vector where element j gives P(K=j | theta)

Methods (by class)

- `cause_probability(series_md)`: Default for series_md models via Monte Carlo integration (Theorem 5)

Examples

```
model <- exp_series_md_c1_c2_c3()
cp <- cause_probability(model)
cp(par = c(0.5, 0.3, 0.2))
```

`component_hazard` *Component hazard function*

Description

Returns a closure computing the hazard function h_j(t; theta) for the j-th component. The returned function takes the full parameter vector `par` and extracts the relevant component parameters internally.

Usage

```
component_hazard(model, j, ...)
```

Arguments

`model`	a likelihood model object
`j`	component index (integer from 1 to m)
`...`	additional arguments passed to the returned closure (e.g., covariates for proportional hazards extensions)

Value

a function with signature `function(t, par, ...)` computing h_j(t)

Examples

```
model <- exp_series_md_c1_c2_c3()
h1 <- component_hazard(model, j = 1)
h1(t = 1.0, par = c(0.5, 0.3, 0.2))
```

`conditional_cause_probability`

Conditional cause-of-failure probability

Description

Returns a closure computing P(K=j | T=t, theta) for all components j, conditional on a specific failure time t. By Theorem 6 of the foundational paper, this equals h_j(t; theta) / sum_l h_l(t; theta).

Usage

```
conditional_cause_probability(model, ...)

## S3 method for class 'series_md'
conditional_cause_probability(model, ...)
```

Arguments

`model`	a likelihood model object
`...`	additional arguments passed to the returned closure

Value

a function with signature `function(t, par, ...)` returning an n x m matrix where n = length(t) and column j gives P(K=j | T=t, theta)

Methods (by class)

- `conditional_cause_probability(series_md)`: Default for series_md models via component hazard ratios (Theorem 6)

Examples

```
model <- exp_series_md_c1_c2_c3()
ccp <- conditional_cause_probability(model)
ccp(t = c(1, 2), par = c(0.5, 0.3, 0.2))
```

`cum_haz` *Cumulative hazard function for a component hazard function*

Description

Creates a cumulative hazard function from a hazard function by integrating.

Usage

```
cum_haz(haz)
```

Arguments

`haz`	hazard function

Value

A function that computes the cumulative hazard at time t

Examples

```
# Exponential hazard h(t) = lambda
haz <- function(t, ...) rep(0.5, length(t))
H <- cum_haz(haz)
H(2)  # Should be 1.0 (0.5 * 2)
```

`dexp_series` *pdf for exponential series.*

Description

pdf for exponential series.

Usage

```
dexp_series(t, rates, log = FALSE)
```

Arguments

`t`	series system lifetime
`rates`	rate parameters for exponential component lifetimes
`log`	return the log of the pdf

Value

Density values for the exponential series distribution.

Examples

```
rates <- c(0.5, 0.3, 0.2)
dexp_series(1.0, rates)
dexp_series(c(0.5, 1.0, 2.0), rates, log = TRUE)
```

`exp_series_md_c1_c2_c3` *Constructs a likelihood model for* `exp_series_md_c1_c2_c3`.

Description

Likelihood model for exponential series systems with masked component cause of failure with candidate sets that satisfy conditions C1, C2, and C3, described below.

Usage

```
exp_series_md_c1_c2_c3(
  rates = NULL,
  lifetime = "t",
  lifetime_upper = "t_upper",
  omega = "omega",
  candset = "x"
)
```

Arguments

`rates`	rate parameters for exponential component lifetimes (optional, used as initial values for MLE if provided)
`lifetime`	column name for system lifetime, defaults to `"t"`
`lifetime_upper`	column name for interval upper bound, defaults to `"t_upper"`. Only used for interval-censored observations.
`omega`	column name for observation type, defaults to `"omega"`. Must contain character values: `"exact"` (failure at t), `"right"` (right-censored at t), `"left"` (left-censored: failed before t), or `"interval"` (failed in (t, t_upper)).
`candset`	column prefix for candidate set indicators, defaults to `"x"`

Details

This model satisfies the concept of a `likelihood_model` in the `likelihood.model` package by providing the following methods:

(1) `loglik.exp_series_md_c1_c2_c3` (2) `score.exp_series_md_c1_c2_c3` (3) `hess_loglik.exp_series_md_c1_c2_c`

These are useful for doing maximum likelihood estimation, hypothesis testing (e.g., likelihood ratio test), estimation of asymptotic sampling distribution given data from the DGP according to the specified model, etc.

It is designed to work well with the `likelihood_model` R package. In particular, it is intended to be used with the `likelihood_contr_model` object, which is a `likelihood_model` object that allows likelihood contributions to be added for whatever data model you have in mind.

In this likelihood model, masked component data approximately satisfies the following conditions:

C1: `Pr{K[i] in C[i]) = 1` C2: `Pr{C[i]=c[i] | K[i]=j, T[i]=t[i]) = Pr(C[i]=c[i] | K[i]=j', T[i]=t[i])` for any `j, j' in c[i]`. C3: masking probabilities are independent of `theta`

As a special case, this model also includes exact component cause of failure data where the candidate set is a singleton.

Value

likelihood model object with class `c("exp_series_md_c1_c2_c3", "series_md", "likelihood_model")`

Examples

```
model <- exp_series_md_c1_c2_c3()
# Generate data and evaluate log-likelihood
set.seed(123)
gen <- rdata(model)
df <- gen(theta = c(0.5, 0.3, 0.2), n = 50, tau = 10, p = 0.3)
ll <- loglik(model)
ll(df, par = c(0.5, 0.3, 0.2))
```

`hazard_exp_series` *Hazard function for exponential series.*

Description

Hazard function for exponential series.

Usage

```
hazard_exp_series(t, rates, log.p = FALSE)
```

Arguments

`t`	Vector of series system lifetimes.
`rates`	Vector of rate parameters for exponential component lifetimes.
`log.p`	Logical; if TRUE, return the log of the hazard function.

Value

The hazard function evaluated at the specified lifetimes.

Examples

```
rates <- c(0.5, 0.3, 0.2)
hazard_exp_series(1.0, rates)
hazard_exp_series(c(0.5, 1.0), rates, log.p = TRUE)
```

`hess_loglik.exp_series_md_c1_c2_c3` *Hessian of log-likelihood method for* `exp_series_md_c1_c2_c3` *model.*

Description

Returns the Hessian (second derivative matrix) of the log-likelihood for an exponential series system with respect to parameter `theta` for masked data with candidate sets that satisfy conditions C1, C2, and C3.

Usage

```
## S3 method for class 'exp_series_md_c1_c2_c3'
hess_loglik(model, ...)
```

Arguments

`model`	the likelihood model object
`...`	additional arguments (passed to returned function)

Details

All four observation types have closed-form Hessian contributions.

Value

hessian function that takes the following arguments:

- `df`: masked data frame
- `par`: rate parameters of exponential component lifetime distributions
- `lifetime`: system lifetime column name (default from model)
- `lifetime_upper`: interval upper bound column name (default from model)
- `omega`: observation type column name (default from model)
- `candset`: prefix of Boolean matrix encoding candidate sets

Examples

```
model <- exp_series_md_c1_c2_c3()
set.seed(1)
df <- rdata(model)(theta = c(0.5, 0.3, 0.2), n = 30, tau = 10, p = 0.3)
H <- hess_loglik(model)
H(df, par = c(0.5, 0.3, 0.2))
```

`hess_loglik.wei_series_homogeneous_md_c1_c2_c3` *Hessian of log-likelihood method for* `wei_series_homogeneous_md_c1_c2_c3`.

Description

Returns the Hessian (second derivative matrix) of the log-likelihood for a Weibull series system with homogeneous shape. Computed numerically via the Jacobian of the score.

Usage

```
## S3 method for class 'wei_series_homogeneous_md_c1_c2_c3'
hess_loglik(model, ...)
```

Arguments

`model`	the likelihood model object
`...`	additional arguments (passed to returned function)

Value

hessian function that takes the following arguments:

- `df`: masked data frame
- `par`: parameter vector (shape, scale1, scale2, ...)
- `lifetime`: system lifetime column name (default from model)
- `lifetime_upper`: interval upper bound column name (default from model)
- `omega`: observation type column name (default from model)
- `candset`: prefix of Boolean matrix encoding candidate sets

Examples

```
model <- wei_series_homogeneous_md_c1_c2_c3()
set.seed(1)
theta <- c(1.2, 1000, 900, 850)
df <- rdata(model)(theta = theta, n = 30, tau = 500, p = 0.3)
H <- hess_loglik(model)
H(df, par = theta)
```

`hess_loglik.wei_series_md_c1_c2_c3` *Hessian of log-likelihood method for* `wei_series_md_c1_c2_c3` *model.*

Description

Returns the Hessian (second derivative matrix) of the log-likelihood for a Weibull series system. Computed numerically via the Jacobian of the score.

Usage

```
## S3 method for class 'wei_series_md_c1_c2_c3'
hess_loglik(model, ...)
```

Arguments

`model`	the likelihood model object
`...`	additional arguments (passed to returned function)

Value

hessian function that takes the following arguments:

- `df`: masked data frame
- `par`: parameter vector (shape1, scale1, shape2, scale2, ...)
- `lifetime`: system lifetime column name (default from model)
- `lifetime_upper`: interval upper bound column name (default from model)
- `omega`: observation type column name (default from model)
- `candset`: prefix of Boolean matrix encoding candidate sets

Examples

```
model <- wei_series_md_c1_c2_c3()
set.seed(1)
theta <- c(1.2, 1000, 0.8, 900)
df <- rdata(model)(theta = theta, n = 30, tau = 500, p = 0.3)
H <- hess_loglik(model)
H(df, par = theta)
```

`loglik.exp_series_md_c1_c2_c3` *Log-likelihood method for* `exp_series_md_c1_c2_c3` *model.*

Description

Returns a log-likelihood function for an exponential series system with respect to rate parameters for masked data with candidate sets that satisfy conditions C1, C2, and C3.

Usage

```
## S3 method for class 'exp_series_md_c1_c2_c3'
loglik(model, ...)
```

Arguments

`model`	the likelihood model object
`...`	additional arguments (passed to returned function)

Details

Supports four observation types: exact failures, right-censored, left-censored, and interval-censored. All have closed-form likelihood contributions for the exponential model.

Value

log-likelihood function that takes the following arguments:

- `df`: masked data frame
- `par`: rate parameters of exponential component lifetime distributions
- `lifetime`: system lifetime column name (default from model)
- `lifetime_upper`: interval upper bound column name (default from model)
- `omega`: observation type column name (default from model)
- `candset`: prefix of Boolean matrix encoding candidate sets

Examples

```
model <- exp_series_md_c1_c2_c3()
set.seed(1)
df <- rdata(model)(theta = c(0.5, 0.3, 0.2), n = 30, tau = 10, p = 0.3)
ll <- loglik(model)
ll(df, par = c(0.5, 0.3, 0.2))
```

`loglik.wei_series_homogeneous_md_c1_c2_c3` *Log-likelihood method for* `wei_series_homogeneous_md_c1_c2_c3` *model.*

Description

Returns a log-likelihood function for a Weibull series system with homogeneous shape parameter. The parameter vector is (k, scale_1, ..., scale_m) for masked data with candidate sets that satisfy conditions C1, C2, and C3.

Usage

```
## S3 method for class 'wei_series_homogeneous_md_c1_c2_c3'
loglik(model, ...)
```

Arguments

`model`	the likelihood model object
`...`	additional arguments (passed to returned function)

Details

Supports four observation types. Left-censored and interval-censored have closed-form likelihood contributions because all shapes are equal.

Value

log-likelihood function that takes the following arguments:

- `df`: masked data frame
- `par`: parameter vector (shape, scale1, scale2, ...)
- `lifetime`: system lifetime column name (default from model)
- `lifetime_upper`: interval upper bound column name (default from model)
- `omega`: observation type column name (default from model)
- `candset`: prefix of Boolean matrix encoding candidate sets

Examples

```
model <- wei_series_homogeneous_md_c1_c2_c3()
set.seed(1)
# theta: (shape, scale1, scale2, scale3)
theta <- c(1.2, 1000, 900, 850)
df <- rdata(model)(theta = theta, n = 30, tau = 500, p = 0.3)
ll <- loglik(model)
ll(df, par = theta)
```

`loglik.wei_series_md_c1_c2_c3` *Log-likelihood method for* `wei_series_md_c1_c2_c3` *model.*

Description

Returns a log-likelihood function for a Weibull series system with respect to parameter vector (shape_1, scale_1, ..., shape_m, scale_m) for masked data with candidate sets that satisfy conditions C1, C2, and C3.

Usage

```
## S3 method for class 'wei_series_md_c1_c2_c3'
loglik(model, ...)
```

Arguments

`model`	the likelihood model object
`...`	additional arguments (passed to returned function)

Details

Supports four observation types. Left-censored and interval-censored observations require numerical integration (via stats::integrate) because heterogeneous shapes prevent a closed-form solution.

Value

log-likelihood function that takes the following arguments:

- `df`: masked data frame
- `par`: parameter vector (shape1, scale1, shape2, scale2, ...)
- `lifetime`: system lifetime column name (default from model)
- `lifetime_upper`: interval upper bound column name (default from model)
- `omega`: observation type column name (default from model)
- `candset`: prefix of Boolean matrix encoding candidate sets

Examples

```
model <- wei_series_md_c1_c2_c3()
set.seed(1)
theta <- c(1.2, 1000, 0.8, 900)
df <- rdata(model)(theta = theta, n = 30, tau = 500, p = 0.3)
ll <- loglik(model)
ll(df, par = theta)
```

`md_bernoulli_cand_c1_c2_c3` *Bernoulli candidate set model for systems with unobserved components.*

Description

Bernoulli candidate set model is a particular type of *uninformed* model. Note that we do not generate candidate sets with this function. See `md_cand_sampler` for that.

Usage

```
md_bernoulli_cand_c1_c2_c3(
  df,
  p,
  prob = "q",
  comp = "t",
  right_censoring_indicator = "delta"
)
```

Arguments

`df`	masked data.
`p`	a vector of probabilities (`p[j]` is the probability that the jth system will include a non-failed component in its candidate set, assuming the jth system is not right-censored).
`prob`	column prefix for component probabilities, defaults to `q`, e.g., `q1, q2, q3`.
`comp`	column prefix for component lifetimes, defaults to `t`, e.g., `t1, t2, t3`.
`right_censoring_indicator`	right-censoring indicator column name. if TRUE, then the system lifetime is right-censored, otherwise it is observed. If NULL, then no right-censoring is assumed. Defaults to `delta`.

Details

This model satisfies conditions C1, C2, and C3. The failed component will be in the corresponding candidate set with probability 1, and the remaining components will be in the candidate set with probability `p` (the same probability for each component). `p` may be different for each system, but it is assumed to be the same for each component within a system, so `p` can be a vector such that the length of `p` is the number of systems in the data set (with recycling if necessary).

Value

Data frame with added candidate set probability columns (e.g., `q1, q2, ..., qm`).

Examples

```
# Generate component lifetimes and system data
mat <- matrix(rexp(9, rate = rep(c(0.5, 0.3, 0.2), each = 3)),
              nrow = 3, ncol = 3)
df <- md_encode_matrix(mat, "t")
df$t <- apply(mat, 1, min)
df$delta <- TRUE
md_bernoulli_cand_c1_c2_c3(df, p = 0.5)
```

`md_boolean_matrix_to_charsets` *Convert Boolean candidate set columns to character set notation*

Description

Replaces Boolean matrix columns (e.g., `x1, x2, x3`) with a single character column showing set notation like `{1, 3}`.

Usage

```
md_boolean_matrix_to_charsets(df, setvar = "x", cname = NULL, drop_set = FALSE)
```

Arguments

`df`	data frame containing Boolean matrix columns
`setvar`	column prefix for the Boolean matrix (default `"x"`)
`cname`	name for the new character column (default: same as `setvar`)
`drop_set`	if TRUE, remove the original Boolean columns (default FALSE)

Value

data frame with character set column added

Examples

```
df <- data.frame(x1 = c(TRUE, FALSE, TRUE),
                 x2 = c(TRUE, TRUE, FALSE),
                 x3 = c(FALSE, TRUE, TRUE))
md_boolean_matrix_to_charsets(df)
```

`md_cand_sampler` *Sample candidate sets for systems with unobserved components.*

Description

Candidate set generator. Requires columns for component probabilities e.g., `q1,...,qm` where `qj` is the probability that the jth component will be in the corresponding candidate set generated for that observation in the `md` table.

Usage

```
md_cand_sampler(df, prob = "q", candset = "x")
```

Arguments

`df`	(masked) data frame
`prob`	column prefix for component probabilities, defaults to `q`, e.g., `q1, q2, q3`.
`candset`	column prefix for candidate sets (as Boolean matrix), defaults to `x`, e.g., `x1, x2, x3`.

Value

Data frame with added Boolean candidate set columns (e.g., `x1, x2, ..., xm`).

Examples

```
# Generate component lifetimes
set.seed(42)
mat <- matrix(rexp(9, rate = rep(c(0.5, 0.3, 0.2), each = 3)),
              nrow = 3, ncol = 3)
df <- md_encode_matrix(mat, "t")
df$t <- apply(mat, 1, min)
df$delta <- TRUE
df <- md_bernoulli_cand_c1_c2_c3(df, p = 0.5)
md_cand_sampler(df)
```

`md_encode_matrix` *Encode a matrix as a data frame with prefixed column names*

Description

Converts a matrix to a data frame with columns named `var1, var2, ...`.

Usage

```
md_encode_matrix(mat, var)
```

Arguments

`mat`	matrix to encode
`var`	character prefix for the column names

Value

a data frame with named columns

Examples

```
mat <- matrix(1:6, nrow = 2, ncol = 3)
md_encode_matrix(mat, "x")
```

`md_series_lifetime_right_censoring` *Masked data generation for series system lifetime data*

Description

Generates right-censored system failure times and right-censoring indicators for a series system with the given data frame of component lifetimes.

Usage

```
md_series_lifetime_right_censoring(
  df,
  tau = Inf,
  comp = "t",
  lifetime = "t",
  right_censoring_indicator = "delta"
)
```

Arguments

`df`	a data frame with the indicated component lifetimes
`tau`	vector of right-censoring times, defaults to `Inf` (no right censoring)
`comp`	component lifetime prefix variable name, defaults to `t`, e.g., `t1, t2, t3`.
`lifetime`	system lifetime variable name, defaults to `t`
`right_censoring_indicator`	right-censoring indicator variable, defaults to `delta`

Value

(masked) data frame with masked data as described in the paper

Examples

```
mat <- matrix(rexp(9, rate = 0.5), nrow = 3, ncol = 3)
df <- md_encode_matrix(mat, "t")
md_series_lifetime_right_censoring(df, tau = 5)
```

`mean.exp_series` *Mean function for exponential series.*

Description

Computes the expected value of a series system with exponentially distributed component lifetimes. For a series system with component rates $\lambda_1, \ldots, \lambda_m$, the system lifetime is exponential with rate $\sum \lambda_j$, so $E[T] = 1/\sum \lambda_j$.

Usage

```
## S3 method for class 'exp_series'
mean(x, ...)
```

Arguments

`x`	An object of class `exp_series` (a vector of rate parameters).
`...`	Additional arguments (ignored, for S3 generic compatibility).

Value

The mean of the exponential series distribution (1/sum of rates).

Examples

```
rates <- structure(c(0.5, 0.3, 0.2), class = "exp_series")
mean(rates)
```

`ncomponents` *Number of components in a series system model*

Description

Returns the number of components m in the series system. If the model was constructed without parameter values, returns NULL.

Usage

```
ncomponents(model, ...)
```

Arguments

`model`	a likelihood model object
`...`	additional arguments (currently ignored)

Value

integer number of components, or NULL if not determinable

Examples

```
model <- exp_series_md_c1_c2_c3(rates = c(0.5, 0.3, 0.2))
ncomponents(model)
```

`observe_left_censor` *Left-censoring observation scheme (single inspection)*

Description

Creates an observation functor for a single-inspection design. If the system has already failed by inspection time `tau`, we know it failed before `tau` but not exactly when (left-censored). If it is still running, we know it survived past `tau` (right-censored).

Usage

```
observe_left_censor(tau)
```

Arguments

`tau`	inspection time (positive numeric)

Value

A function with signature `function(t_true)` returning a list with components:

t inspection time `tau`

omega "left" if failed before `tau`, "right" otherwise

t_upper NA (not used for this scheme)

Examples

```
obs <- observe_left_censor(tau = 100)
obs(50)   # left-censored: list(t = 100, omega = "left", t_upper = NA)
obs(150)  # right-censored: list(t = 100, omega = "right", t_upper = NA)
```

`observe_mixture` *Mixture of observation schemes*

Description

Creates an observation functor that randomly selects from a set of observation schemes for each observation. This models heterogeneous monitoring environments where different units are observed differently.

Usage

```
observe_mixture(..., weights = NULL)
```

Arguments

`...`	observation functors (created by `observe_*` functions)
`weights`	mixing probabilities (numeric vector). If NULL, uniform weights are used. Weights are normalized to sum to 1.

Value

A function with signature `function(t_true)` returning a list from one of the constituent schemes, selected randomly according to `weights`.

Examples

```
obs <- observe_mixture(
  observe_right_censor(tau = 100),
  observe_left_censor(tau = 50),
  weights = c(0.7, 0.3)
)
set.seed(42)
obs(30)  # randomly selects one of the two schemes
```

`observe_periodic` *Periodic inspection observation scheme*

Description

Creates an observation functor for periodic inspections at intervals of `delta`. Failures are bracketed between the last inspection before failure and the first inspection after failure (interval-censored). Systems surviving past `tau` are right-censored.

Usage

```
observe_periodic(delta, tau = Inf)
```

Arguments

`delta`	inspection interval (positive numeric)
`tau`	study end time (positive numeric or Inf for no right-censoring)

Value

A function with signature `function(t_true)` returning a list with components:

t lower bound of interval (or `tau` if right-censored)

omega "interval" or "right"

t_upper upper bound of interval (NA if right-censored)

Examples

```
obs <- observe_periodic(delta = 10, tau = 100)
obs(25)   # interval: list(t = 20, omega = "interval", t_upper = 30)
obs(150)  # right-censored: list(t = 100, omega = "right", t_upper = NA)
```

`observe_right_censor` *Right-censoring observation scheme*

Description

Creates an observation functor that applies right-censoring at time `tau`. Systems that fail before `tau` are observed exactly; systems surviving past `tau` are right-censored.

Usage

```
observe_right_censor(tau)
```

Arguments

`tau`	censoring time (positive numeric)

Value

A function with signature `function(t_true)` returning a list with components:

t observed time

omega "exact" or "right"

t_upper NA (not used for this scheme)

Examples

```
obs <- observe_right_censor(tau = 100)
obs(50)   # exact: list(t = 50, omega = "exact", t_upper = NA)
obs(150)  # right-censored: list(t = 100, omega = "right", t_upper = NA)
```

`pexp_series` *Cumulative distribution function for exponential series.*

Description

Cumulative distribution function for exponential series.

Usage

```
pexp_series(t, rates, lower.tail = TRUE, log.p = FALSE)
```

Arguments

`t`	Vector of series system lifetimes.
`rates`	Vector of rate parameters for exponential component lifetimes.
`lower.tail`	Logical; if TRUE (default), probabilities are P(X <= x), otherwise, P(X > x).
`log.p`	Logical; if TRUE, return the log of the cdf.

Value

The cumulative probabilities evaluated at the specified lifetimes.

Examples

```
rates <- c(0.5, 0.3, 0.2)
pexp_series(1.0, rates)
pexp_series(c(0.5, 1.0, 2.0), rates)
```

`qcomp` *Quantile function for a component with custom survival function*

Description

Finds the time t such that S(t) = p using root finding. The survival function S(t) is assumed to be monotonically decreasing from S(0) = 1 to S(inf) = 0.

Usage

```
qcomp(
  p,
  surv,
  theta,
  t_lower = .Machine$double.eps,
  t_upper = .Machine$double.xmax^0.5,
  ...
)
```

Arguments

`p`	probability (quantile level), must be in (0, 1)
`surv`	survival function S(t, theta, ...)
`theta`	parameter vector passed to surv
`t_lower`	lower bound for search interval
`t_upper`	upper bound for search interval (sqrt to avoid overflow)
`...`	additional arguments passed to surv

Value

time t such that S(t) = p

Examples

```
# Exponential survival function
surv_exp <- function(t, theta) exp(-theta * t)

# Median lifetime (50th percentile) for rate = 2
qcomp(0.5, surv = surv_exp, theta = 2.0)
```

`qexp_series` *Quantile function for exponential series.*

Description

Quantile function for exponential series.

Usage

```
qexp_series(p, rates, lower.tail = TRUE, log.p = FALSE)
```

Arguments

`p`	Vector of quantiles.
`rates`	Vector of rate parameters for exponential component lifetimes.
`lower.tail`	Logical, if TRUE (default), probabilities are P(X <= x), otherwise, P(X > x).
`log.p`	Logical, if TRUE, vector of probabilities `p` are given as `log(p)`.

Value

Quantiles corresponding to the given probabilities `p`.

Examples

```
rates <- c(0.5, 0.3, 0.2)
qexp_series(0.5, rates)
qexp_series(c(0.25, 0.5, 0.75), rates)
```

`rcomp` *Random generation for a component with custom survival function*

Description

Generates random samples using inverse transform sampling.

Usage

```
rcomp(n, surv, theta)
```

Arguments

`n`	number of samples to generate
`surv`	survival function S(t, theta)
`theta`	parameter vector passed to surv

Value

vector of n random samples

Examples

```
# Exponential survival function
surv_exp <- function(t, theta) exp(-theta * t)

# Generate 10 random samples with rate = 2
set.seed(123)
rcomp(10, surv = surv_exp, theta = 2.0)
```

`rdata.exp_series_md_c1_c2_c3` *Random data generation for* `exp_series_md_c1_c2_c3` *model.*

Description

Returns a function that generates random masked data from the exponential series system DGP at a given parameter value.

Usage

```
## S3 method for class 'exp_series_md_c1_c2_c3'
rdata(model, ...)
```

Arguments

`model`	the likelihood model object
`...`	additional arguments (passed to returned function)

Value

function that takes (theta, n, tau, p, observe, ...) and returns a data frame with columns: t, omega, x1, x2, ..., xm

Examples

```
model <- exp_series_md_c1_c2_c3()
gen <- rdata(model)
set.seed(42)
df <- gen(theta = c(0.5, 0.3, 0.2), n = 10, tau = 5, p = 0.5)
head(df)
```

`rdata.wei_series_homogeneous_md_c1_c2_c3`

Random data generation for `wei_series_homogeneous_md_c1_c2_c3` *model.*

Description

Returns a function that generates random masked data from the homogeneous Weibull series system DGP at a given parameter value.

Usage

```
## S3 method for class 'wei_series_homogeneous_md_c1_c2_c3'
rdata(model, ...)
```

Arguments

`model`	the likelihood model object
`...`	additional arguments (passed to returned function)

Value

function that takes (theta, n, tau, p, observe, ...) and returns a data frame with columns: t, omega, x1, x2, ..., xm

Examples

```
model <- wei_series_homogeneous_md_c1_c2_c3()
gen <- rdata(model)
set.seed(42)
df <- gen(theta = c(1.2, 1000, 900, 850), n = 10, tau = 500, p = 0.5)
head(df)
```

`rdata.wei_series_md_c1_c2_c3` *Random data generation for* `wei_series_md_c1_c2_c3` *model.*

Description

Returns a function that generates random masked data from the Weibull series system DGP at a given parameter value.

Usage

```
## S3 method for class 'wei_series_md_c1_c2_c3'
rdata(model, ...)
```

Arguments

`model`	the likelihood model object
`...`	additional arguments (passed to returned function)

Value

function that takes (theta, n, tau, p, observe, ...) and returns a data frame with columns: t, omega, x1, x2, ..., xm

Examples

```
model <- wei_series_md_c1_c2_c3()
gen <- rdata(model)
set.seed(42)
# theta: (shape1, scale1, shape2, scale2)
df <- gen(theta = c(1.2, 1000, 0.8, 900), n = 10, tau = 500, p = 0.5)
head(df)
```

`rexp_series` *Random number generation for exponential series.*

Description

Generates random variates from an exponential series distribution.

Usage

```
rexp_series(n, rates, keep_latent = FALSE)
```

Arguments

`n`	Integer, number of observations to generate.
`rates`	Vector of rate parameters for exponential component lifetimes.
`keep_latent`	Logical; if TRUE, returns a matrix with system lifetimes in the first column and individual component lifetimes in subsequent columns. If FALSE (default), returns only system lifetimes.

Value

If `keep_latent = FALSE`, a vector of random variates from the exponential series distribution. If `keep_latent = TRUE`, a matrix with system lifetime in the first column and component lifetimes in columns 2 through m+1.

Examples

```
set.seed(123)
rexp_series(5, rates = c(0.5, 0.3, 0.2))
rexp_series(3, rates = c(0.5, 0.3, 0.2), keep_latent = TRUE)
```

`score.exp_series_md_c1_c2_c3`

Score method for `exp_series_md_c1_c2_c3` *model.*

Description

Returns a score (gradient) function for an exponential series system with respect to parameter `theta` for masked component failure with candidate sets that satisfy conditions C1, C2, and C3.

Usage

```
## S3 method for class 'exp_series_md_c1_c2_c3'
score(model, ...)
```

Arguments

`model`	the likelihood model object
`...`	additional arguments (passed to returned function)

Details

All four observation types (exact, right, left, interval) have closed-form score contributions.

Value

score function that takes the following arguments:

- `df`: masked data frame
- `par`: rate parameters of exponential component lifetime distributions
- `lifetime`: system lifetime column name (default from model)
- `lifetime_upper`: interval upper bound column name (default from model)
- `omega`: observation type column name (default from model)
- `candset`: prefix of Boolean matrix encoding candidate sets

Examples

```
model <- exp_series_md_c1_c2_c3()
set.seed(1)
df <- rdata(model)(theta = c(0.5, 0.3, 0.2), n = 30, tau = 10, p = 0.3)
s <- score(model)
s(df, par = c(0.5, 0.3, 0.2))
```

`score.wei_series_homogeneous_md_c1_c2_c3`

Score method for `wei_series_homogeneous_md_c1_c2_c3` *model.*

Description

Returns a score (gradient) function for a Weibull series system with homogeneous shape parameter. The parameter vector is (k, scale_1, ..., scale_m) for masked data with candidate sets that satisfy conditions C1, C2, and C3.

Usage

```
## S3 method for class 'wei_series_homogeneous_md_c1_c2_c3'
score(model, ...)
```

Arguments

`model`	the likelihood model object
`...`	additional arguments (passed to returned function)

Details

Uses a hybrid approach: analytical score for exact and right-censored observations, numerical gradient (via numDeriv) for left-censored and interval-censored observations.

Value

score function that takes the following arguments:

- `df`: masked data frame
- `par`: parameter vector (shape, scale1, scale2, ...)
- `lifetime`: system lifetime column name (default from model)
- `lifetime_upper`: interval upper bound column name (default from model)
- `omega`: observation type column name (default from model)
- `candset`: prefix of Boolean matrix encoding candidate sets

Examples

```
model <- wei_series_homogeneous_md_c1_c2_c3()
set.seed(1)
theta <- c(1.2, 1000, 900, 850)
df <- rdata(model)(theta = theta, n = 30, tau = 500, p = 0.3)
s <- score(model)
s(df, par = theta)
```

`score.wei_series_md_c1_c2_c3`

Score method for `wei_series_md_c1_c2_c3` *model.*

Description

Returns a score (gradient) function for a Weibull series system with respect to parameter vector (shape_1, scale_1, ..., shape_m, scale_m) for masked data with candidate sets that satisfy conditions C1, C2, and C3.

Usage

```
## S3 method for class 'wei_series_md_c1_c2_c3'
score(model, ...)
```

Arguments

`model`	the likelihood model object
`...`	additional arguments (passed to returned function)

Details

Uses a hybrid approach: analytical score for exact and right-censored observations, numerical gradient (via numDeriv) for left-censored and interval-censored observations.

Value

score function that takes the following arguments:

- `df`: masked data frame
- `par`: parameter vector (shape1, scale1, shape2, scale2, ...)
- `lifetime`: system lifetime column name (default from model)
- `lifetime_upper`: interval upper bound column name (default from model)
- `omega`: observation type column name (default from model)
- `candset`: prefix of Boolean matrix encoding candidate sets

Examples

```
model <- wei_series_md_c1_c2_c3()
set.seed(1)
theta <- c(1.2, 1000, 0.8, 900)
df <- rdata(model)(theta = theta, n = 30, tau = 500, p = 0.3)
s <- score(model)
s(df, par = theta)
```

`surv.exp_series` *Survival function for exponential series.*

Description

Survival function for exponential series.

Usage

```
surv.exp_series(t, rates, log.p = FALSE)
```

Arguments

`t`	Vector of series system lifetimes.
`rates`	Vector of rate parameters for exponential component lifetimes.
`log.p`	Logical; if TRUE, return the log of the survival function.

Value

The survival function evaluated at the specified lifetimes.

Examples

```
rates <- c(0.5, 0.3, 0.2)
surv.exp_series(1.0, rates)
surv.exp_series(c(0.5, 1.0, 2.0), rates)
```

`wei_series_homogeneous_md_c1_c2_c3` *Constructs a likelihood model for* `wei_series_homogeneous_md_c1_c2_c3`.

Description

Likelihood model for Weibull series systems with homogeneous shape parameter and masked component cause of failure with candidate sets that satisfy conditions C1, C2, and C3.

Usage

```
wei_series_homogeneous_md_c1_c2_c3(
  shape = NULL,
  scales = NULL,
  lifetime = "t",
  lifetime_upper = "t_upper",
  omega = "omega",
  candset = "x"
)
```

Arguments

`shape`	common shape parameter for all Weibull component lifetimes
`scales`	scale parameters for Weibull component lifetimes (optional)
`lifetime`	column name for system lifetime, defaults to `"t"`
`lifetime_upper`	column name for interval upper bound, defaults to `"t_upper"`. Only used for interval-censored observations.
`omega`	column name for observation type, defaults to `"omega"`. Must contain character values: `"exact"`, `"right"`, `"left"`, or `"interval"`.
`candset`	column prefix for candidate set indicators, defaults to `"x"`

Details

This is a reduced model where all components share a common shape parameter k, while retaining individual scale parameters. The parameter vector is (k, scale_1, ..., scale_m), giving m+1 parameters instead of 2m.

A key property of this model is that the series system lifetime is itself Weibull distributed with shape k and scale $\lambda_s = (\sum \beta_j^{-k})^{-1/k}$.

This model satisfies the concept of a `likelihood_model` in the `likelihood.model` package by providing the following methods:

(1) `loglik.wei_series_homogeneous_md_c1_c2_c3` (2) `score.wei_series_homogeneous_md_c1_c2_c3` (3) `hess_loglik.wei_series_homogeneous_md_c1_c2_c3`

In this likelihood model, masked component data approximately satisfies:

C1: `Pr{K[i] in C[i]} = 1` C2: `Pr{C[i]=c[i] | K[i]=j, T[i]=t[i]} = Pr{C[i]=c[i] | K[i]=j', T[i]=t[i]}` for any `j, j' in c[i]`. C3: masking probabilities are independent of `theta`

Value

likelihood model object with class `c("wei_series_homogeneous_md_c1_c2_c3", "series_md", "likelihood_model")`

See Also

`wei_series_md_c1_c2_c3()` for the full model with heterogeneous shapes

Examples

```
# Create model and fit to data using generic dispatch
model <- wei_series_homogeneous_md_c1_c2_c3()
# solver <- fit(model)
# theta: (shape, scale1, scale2, ...)
# mle <- solver(data, par = c(1.2, 1000, 900, 850))
```

`wei_series_md_c1_c2_c3` *Constructs a likelihood model for* `wei_series_md_c1_c2_c3`.

Description

Likelihood model for Weibull series systems with masked component cause of failure with candidate sets that satisfy conditions C1, C2, and C3.

Usage

```
wei_series_md_c1_c2_c3(
  shapes = NULL,
  scales = NULL,
  lifetime = "t",
  lifetime_upper = "t_upper",
  omega = "omega",
  candset = "x"
)
```

Arguments

`shapes`	shape parameters for Weibull component lifetimes (optional)
`scales`	scale parameters for Weibull component lifetimes (optional)
`lifetime`	column name for system lifetime, defaults to `"t"`
`lifetime_upper`	column name for interval upper bound, defaults to `"t_upper"`. Only used for interval-censored observations.
`omega`	column name for observation type, defaults to `"omega"`. Must contain character values: `"exact"`, `"right"`, `"left"`, or `"interval"`.
`candset`	column prefix for candidate set indicators, defaults to `"x"`

Details

This model satisfies the concept of a `likelihood_model` in the `likelihood.model` package by providing the following methods:

(1) `loglik.wei_series_md_c1_c2_c3` (2) `score.wei_series_md_c1_c2_c3` (3) `hess_loglik.wei_series_md_c1_c2_c`

The Weibull series system has 2m parameters: (shape_1, scale_1, ..., shape_m, scale_m).

In this likelihood model, masked component data approximately satisfies:

C1: `Pr{K[i] in C[i]} = 1` C2: `Pr{C[i]=c[i] | K[i]=j, T[i]=t[i]} = Pr{C[i]=c[i] | K[i]=j', T[i]=t[i]}` for any `j, j' in c[i]`. C3: masking probabilities are independent of `theta`

Value

likelihood model object with class `c("wei_series_md_c1_c2_c3", "series_md", "likelihood_model")`

Examples

```
# Create model and fit to data using generic dispatch
model <- wei_series_md_c1_c2_c3()
# solver <- fit(model)
# theta: (shape1, scale1, shape2, scale2, ...)
# mle <- solver(data, par = c(1, 1000, 1, 1000, 1, 1000))
```

`wei_series_system_scale` *System scale parameter for homogeneous Weibull series*

Description

For a series system with Weibull components sharing shape k but with individual scales, the system lifetime is itself Weibull with shape k and this computed scale.

Usage

```
wei_series_system_scale(k, scales)
```

Arguments

`k`	common shape parameter
`scales`	vector of component scale parameters

Value

system scale parameter

Examples

```
# 3-component system with common shape 1.2
wei_series_system_scale(k = 1.2, scales = c(1000, 900, 850))
```

Index