Package 'maskedhaz'

April 21, 2026

Title Masked-Cause Likelihood Models for Series Systems with Arbitrary Hazard Components

Version 0.1.0

Description Likelihood-based inference for series systems with masked component cause of failure, using arbitrary dynamic failure rate component distributions. Computes log-likelihood, score, Hessian, and maximum likelihood estimates for masked data satisfying conditions C1, C2, C3 under general component hazard functions. Implements the 'series_md' protocol defined in the 'maskedcauses' package.

License GPL (>= 3)

URL https://github.com/queelius/maskedhaz, https://queelius.github.io/maskedhaz/

BugReports https://github.com/queelius/maskedhaz/issues

Encoding UTF-8

Language en-US

RoxygenNote 7.3.3

Depends R (>= 3.5.0)

Imports serieshaz, flexhaz, algebraic.dist, likelihood.model, maskedcauses, generics, numDeriv, stats

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

VignetteBuilder knitr

Config/testthat/edition 3

NeedsCompilation no

Author Alexander Towell [aut, cre]

Maintainer Alexander Towell <lex@metafunctor.com>

Repository CRAN

Date/Publication 2026-04-21 17:52:08 UTC

Contents

`assumptions.dfr_series_md` *Assumptions for masked-cause DFR series systems*

Description

Assumptions for masked-cause DFR series systems

Usage

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

Arguments

`model`	A `dfr_series_md` object.
`...`	Additional arguments (unused).

Value

Character vector of model assumptions.

Examples

```
model <- dfr_series_md(components = list(
  dfr_exponential(0.1), dfr_exponential(0.2)
))
assumptions(model)
```

cause_probability.dfr_series_md

Marginal cause-of-failure probability for DFR series systems

Description

Method for `cause_probability` that returns a closure computing $P(K = j \mid \theta)$ for each component, marginalized over the system failure time T via Monte Carlo integration. By Theorem 5, this equals $E_T[P(K = j \mid T, \theta)]$.

Usage

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

Arguments

`model`	A `dfr_series_md` object.
`...`	Additional arguments passed to the returned closure.

Value

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

Examples

```
model <- dfr_series_md(components = list(
  dfr_exponential(0.1), dfr_exponential(0.2), dfr_exponential(0.3)
))
cp_fn <- cause_probability(model)
set.seed(1)
cp_fn(par = c(0.1, 0.2, 0.3), n_mc = 2000)
```

component_hazard.dfr_series_md

Component hazard for a masked-cause DFR series system

Description

Component hazard for a masked-cause DFR series system

Usage

```
## S3 method for class 'dfr_series_md'
component_hazard(x, j, ...)
```

Arguments

x	A dfr_series_md object.
j	Component index.
...	Additional arguments passed to the closure.

Value

A closure computing component j's hazard.

Examples

```
model <- dfr_series_md(components = list(
  dfr_exponential(0.1), dfr_exponential(0.2)
))
h1 <- component_hazard(model, 1)
h1(t = 5, par = 0.1)   # 0.1 (constant exponential hazard)
```

---

conditional_cause_probability.dfr_series_md

Conditional cause-of-failure probability for DFR series systems

---

Description

Method for conditional_cause_probability that returns a closure computing $P(K = j \mid T = t, \theta)$ for each component. By Theorem 6 of the foundational paper, this equals $h_j(t;\theta) / \sum_l h_l(t;\theta)$.

Usage

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

Arguments

model	A dfr_series_md object.
...	Additional arguments passed to the returned closure.

Value

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

Examples

```
model <- dfr_series_md(components = list(
  dfr_exponential(0.1), dfr_exponential(0.2), dfr_exponential(0.3)
))
ccp_fn <- conditional_cause_probability(model)
ccp_fn(t = c(1, 5, 10), par = c(0.1, 0.2, 0.3))
```

`dfr_series_md` *Masked-Cause Likelihood Model for DFR Series Systems*

Description

Constructs a likelihood model for series systems with masked component cause of failure, where components are arbitrary `dfr_dist` distributions. Supports exact, right-censored, left-censored, and interval-censored observations with candidate sets satisfying C1-C2-C3.

Usage

```
dfr_series_md(
  series = NULL,
  components = NULL,
  par = NULL,
  n_par = NULL,
  lifetime = "t",
  lifetime_upper = "t_upper",
  omega = "omega",
  candset = "x"
)
```

Arguments

`series`	A `dfr_dist_series` object. Ignored if `components` is provided.
`components`	A list of `dfr_dist` objects. If provided, a `dfr_dist_series` is built from these.
`par`	Optional concatenated parameter vector.
`n_par`	Optional integer vector of parameter counts per component.
`lifetime`	Column name for system lifetime (default `"t"`).
`lifetime_upper`	Column name for interval upper bound (default `"t_upper"`).
`omega`	Column name for observation type (default `"omega"`).
`candset`	Column prefix for candidate set indicators (default `"x"`).

Details

The model computes the masked-cause log-likelihood for series systems where the system lifetime is the minimum of independent component lifetimes, and the causing component is partially observed through candidate sets.

Observation types (stored in the `omega` column):

`"exact"` Failed at time t, cause masked among candidates

`"right"` Right-censored: survived past time t

`"left"` Left-censored: failed before time t

`"interval"` Failed in interval (t, t_upper)

Masking conditions:

C1 Failed component is in candidate set with probability 1

C2 Uniform probability for candidate sets given component cause

C3 Masking probabilities independent of system parameters

Value

An object of class `c("dfr_series_md", "series_md", "likelihood_model")`.

See Also

`is_dfr_series_md` for the type predicate, `dfr_dist_series` for the series distribution, `loglik` for the likelihood interface

Examples

```
library(flexhaz)
library(serieshaz)

# From components
model <- dfr_series_md(components = list(
    dfr_exponential(0.1),
    dfr_exponential(0.2),
    dfr_exponential(0.3)
))

# From pre-built series
sys <- dfr_dist_series(list(
    dfr_weibull(shape = 2, scale = 100),
    dfr_exponential(0.05)
))
model2 <- dfr_series_md(series = sys)
```

`fit.dfr_series_md` *MLE fitting for masked-cause DFR series systems*

Description

Returns a solver function that finds the maximum likelihood estimates for component parameters given masked series system data.

Usage

```
## S3 method for class 'dfr_series_md'
fit(object, ...)
```

Arguments

object	A dfr_series_md object.
...	Additional arguments (currently unused).

Details

Uses optim to maximize the log-likelihood. The score function (gradient) is computed from the same loglik closure via grad, and the Hessian at the MLE via hessian. One-parameter problems auto-upgrade from Nelder-Mead to BFGS with a warning, because Nelder-Mead is unreliable in one dimension.

Value

A solver function with signature `function(df, par, method = "Nelder-Mead", ..., control = list())` that returns a fisher_mle object.

Examples

```
model <- dfr_series_md(components = list(
  dfr_exponential(0.1), dfr_exponential(0.2)
))
set.seed(1)
df <- rdata(model)(theta = c(0.1, 0.2), n = 200, tau = 10, p = 0)
solver <- fit(model)
result <- solver(df, par = c(0.15, 0.15))
coef(result)
```

---

hess_loglik.dfr_series_md

Hessian of log-likelihood for masked-cause DFR series systems

---

Description

Returns a Hessian function computed via numerical differentiation of the log-likelihood using hessian.

Usage

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

Arguments

model	A dfr_series_md object.
...	Additional arguments (currently unused).

Value

A function with signature `function(df, par, ...)` returning the Hessian matrix.

Examples

```
model <- dfr_series_md(components = list(
  dfr_exponential(0.1), dfr_exponential(0.2)
))
set.seed(1)
df <- rdata(model)(theta = c(0.1, 0.2), n = 50, tau = 10, p = 0.3)
H_fn <- hess_loglik(model)
H_fn(df, par = c(0.1, 0.2))
```

---

is_dfr_series_md *Test whether an object is a dfr_series_md*

Description

Test whether an object is a dfr_series_md

Usage

```
is_dfr_series_md(x)
```

Arguments

`x`	Object to test.

Value

Logical scalar.

Examples

```
model <- dfr_series_md(components = list(
  dfr_exponential(0.1), dfr_exponential(0.2)
))
is_dfr_series_md(model)    # TRUE
is_dfr_series_md(42)       # FALSE
```

`loglik.dfr_series_md` *Log-likelihood for masked-cause DFR series systems*

Description

Returns a log-likelihood function for a series system with masked component cause of failure. Supports four observation types: exact failures, right-censored, left-censored, and interval-censored.

Usage

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

Arguments

`model`	A `dfr_series_md` object.
`...`	Additional arguments (currently unused).

Details

Log-likelihood contributions by observation type:

Exact ($\omega =$ **"exact"**) $\log L_i = \log(\sum_{j \in C_i} h_j(t_i)) - H_{sys}(t_i)$

Right-censored ($\omega =$ **"right"**) $\log L_i = -H_{sys}(t_i)$

Left-censored ($\omega =$ **"left"**) $\log L_i = \log \int_0^{t_i} [\sum_{j \in C_i} h_j(u)] S_{sys}(u)\, du$

Interval-censored ($\omega =$ **"interval"**) $\log L_i = \log \int_{t_i}^{t_{upper,i}} [\sum_{j \in C_i} h_j(u)] S_{sys}(u)\, du$

The exact and right-censored paths use vectorized hazard / cumulative hazard calls. Left and interval censoring require per-row numerical integration via `integrate`.

The returned closure caches validated and decoded masked-data extracted from the data frame across repeated calls with the same `df`, so that the O(n) validation cost is paid only once per `optim`/`numDeriv` sweep. The cache is per-closure, kept in the closure's enclosing environment. This is safe for sequential use; if you share the same closure object across forked workers (e.g. `parallel::mcparallel`), concurrent writes to the cache are possible but only affect performance, not correctness.

Value

A function with signature `function(df, par, ...)` that computes the log-likelihood.

Examples

```
model <- dfr_series_md(components = list(
  dfr_exponential(0.1), dfr_exponential(0.2)
))
set.seed(1)
df <- rdata(model)(theta = c(0.1, 0.2), n = 50, tau = 10, p = 0.3)
ll_fn <- loglik(model)
ll_fn(df, par = c(0.1, 0.2))
```

`ncomponents.dfr_series_md` *Number of components in a masked-cause DFR series system*

Description

Number of components in a masked-cause DFR series system

Usage

```
## S3 method for class 'dfr_series_md'
ncomponents(x, ...)
```

Arguments

`x`	A `dfr_series_md` object.
`...`	Additional arguments (unused).

Value

Integer, the number of components.

Examples

```
model <- dfr_series_md(components = list(
  dfr_exponential(0.1), dfr_exponential(0.2), dfr_exponential(0.3)
))
ncomponents(model)   # 3
```

print.dfr_series_md *Print method for dfr_series_md*

Description

Print method for dfr_series_md

Usage

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

Arguments

x	A dfr_series_md object.
...	Additional arguments (unused).

Value

Invisibly returns x.

Examples

```
model <- dfr_series_md(components = list(
  dfr_weibull(shape = 2, scale = 100),
  dfr_exponential(0.05)
))
print(model)
```

rdata.dfr_series_md *Random data generation for masked-cause DFR series systems*

Description

Returns a function that generates random masked series system data from the model's data-generating process (DGP). Uses sample_components for component lifetimes and applies right-censoring and masking satisfying C1-C2-C3.

Usage

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

Arguments

model	A dfr_series_md object.
...	Additional arguments (currently unused).

Value

A function with signature `function(theta, n, tau = Inf, p = 0, ...)` that returns a data frame with columns for lifetime, observation type, and candidate sets.

Examples

```
model <- dfr_series_md(components = list(
  dfr_exponential(0.1), dfr_exponential(0.2)
))
set.seed(1)
df <- rdata(model)(theta = c(0.1, 0.2), n = 20, tau = 10, p = 0.3)
head(df)
```

---

score.dfr_series_md *Score function for masked-cause DFR series systems*

Description

Returns a score (gradient) function computed via numerical differentiation of the log-likelihood using `grad`.

Usage

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

Arguments

`model`	A `dfr_series_md` object.
`...`	Additional arguments (currently unused).

Value

A function with signature `function(df, par, ...)` returning the gradient vector.

Examples

```
model <- dfr_series_md(components = list(
  dfr_exponential(0.1), dfr_exponential(0.2)
))
set.seed(1)
df <- rdata(model)(theta = c(0.1, 0.2), n = 50, tau = 10, p = 0.3)
s_fn <- score(model)
s_fn(df, par = c(0.1, 0.2))
```

Index