Package 'masked.data'

January 20, 2022

Title Masked Data

Version 0.0.0.9000

Description Set of functions for working with masked data, estimating series systems from masked data, and making predictions from masked data.

License GPL (>= 3)

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.1.2

Suggests knitr,
rmarkdown,
testthat (>= 3.0.0)

Config/testthat/edition 3

Imports dplyr,
extraDistr,
jsonlite,
matlib,
numDeriv,
readr,
stats,
tibble,
mvtnorm

Depends R (>= 2.10)

LazyData true

VignetteBuilder knitr

R topics documented:

`confint.md_estimate` *Method to obtain the confidence intervals of the parameter values of a masked data estimator,* `md_estimate`.

Description

Method to obtain the confidence intervals of the parameter values of a masked data estimator, `md_estimate`.

Usage

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

Arguments

`object`	The `md_estimate` object to compute the confidence intervals for
`parm`	Unused
`level`	Confidence level, defaults to 0.95 (alpha=.05)

`exp_series_data_1` *Masked data for a series system with exponentially distributed nodes*

Description

Masked data containing the system lifetime and other attributes of 1000 series system with parameter value `theta=c(3,4,5)` and candidate model m0.

Usage

```
exp_series_data_1
```

Format

A data frame with 1000 rows and 9 variables:

s Real observable variable, system lifetime

k Integer latent variable, the failed node

w Integer observable variable, number of candidates

t.1 Real latent variable, lifetime of node 1

t.2 Real latent variable, lifetime of node 2

t.3 Real latent variable, lifetime of node 3

c.1 Boolean observable variable, TRUE indicates node 1 is in candidate set

c.2 Boolean observable variable, TRUE indicates node 2 is in candidate set

c.3 Boolean observable variable, TRUE indicates node 3 is in candidate set

Details

Each candidate is of size `w=2`.

Source

https://github.com/queelius/masked.data/blob/master/data-raw/exp_series_data_1_gen.R

`exp_series_data_2` *Masked data for a 5-out-of-5 (series) system with exponentially distributed nodes.*

Description

Masked data containing the system lifetime and other attributes of 1000 series parameterized by `theta=c(3,4,5,6,7)` and candidate model m0.

Usage

```
exp_series_data_2
```

Format

A data frame with 100000 rows and 13 variables:

s Real observable variable, system lifetime

k Integer latent variable, the failed node

w Integer observable variable, number of candidates

t.1 Real latent variable, lifetime of node 1

t.2 Real latent variable, lifetime of node 2

t.3 Real latent variable, lifetime of node 3

t.4 Real latent variable, lifetime of node 4

t.5 Real latent variable, lifetime of node 5

c.1 Boolean observable variable, TRUE indicates node 1 is in candidate set

c.2 Boolean observable variable, TRUE indicates node 2 is in candidate set

c.3 Boolean observable variable, TRUE indicates node 3 is in candidate set

c.4 Boolean observable variable, TRUE indicates node 1 is in candidate set

c.5 Boolean observable variable, TRUE indicates node 2 is in candidate set

Details

Candidate set sizes are randomly drawn from `{2,3,4}`.

Source

https://github.com/queelius/masked.data/blob/master/data-raw/exp_series_data_2_gen.R

`exp_series_data_3` *Masked data for a 10-out-of-10 (series system) with exponentially distributed nodes.*

Description

Masked data containing a sample of 100000 system lifetimes and other attributes where the system is parameterized by `theta=c(3,5,4,6,7,2,8,9,10,11)` and candidate model is m0.

Usage

```
exp_series_data_3
```

Format

A data frame with 100000 rows and 23 variables:

s Real observable variable, system lifetime

k Integer latent variable, the failed node

w Integer observable variable, number of candidates

t.1-t.10 Real latent variable, lifetimes of the 10 nodes

c.1-c.10 Boolean observable variable, c.j TRUE indicates nodes j is in candidate set

Details

Candidate set sizes are randomly drawn from 2,3,4,5,6,7,8,9.

Source

https://github.com/queelius/masked.data/blob/master/data-raw/exp_series_data_3_gen.R

`exp_series_data_4` *Masked data for a 3-out-of-3 (series system) with exponentially distributed nodes.*

Description

Masked data containing a sample of 10000 system lifetimes and other attributes where the system is parameterized by `theta=c(1,1,1)` and candidate model m0.

Usage

```
exp_series_data_4
```

Format

A data frame with 10000 rows and 9 variables:

s Real observable variable, system lifetime

k Integer latent variable, the failed node

w Integer observable variable, number of candidates

t.1 Real latent variable, lifetime of node 1

t.2 Real latent variable, lifetime of node 2

t.3 Real latent variable, lifetime of node 3

c.1 Boolean observable variable, c.1 TRUE indicates nodes j is in candidate set

c.2 Boolean observable variable, c.2 TRUE indicates nodes j is in candidate set

c.3 Boolean observable variable, c.3 TRUE indicates nodes j is in candidate set

Details

Candidate set sizes are `w=2`.

Source

https://github.com/queelius/masked.data/blob/master/data-raw/exp_series_data_4_gen.R

`fisher_info` *Generic method for obtaining the fisher information matrix of an estimator, if supported.*

Description

Generic method for obtaining the fisher information matrix of an estimator, if supported.

Usage

```
fisher_info(x, ...)
```

Arguments

`x`	The object to obtain the fisher information of

`fisher_info.md_estimate` *Method to obtain the fisher information matrix of an* `md_estimate`.

Description

Method to obtain the fisher information matrix of an `md_estimate`.

Usage

```
## S3 method for class 'md_estimate'
fisher_info(x, ...)
```

Arguments

`x`	The `md_estimate` object to obtain the fisher information of

`hazard` *Generic method for obtaining the hazard function of a random variable.*

Description

Generic method for obtaining the hazard function of a random variable.

Usage

```
hazard(x, ...)
```

Arguments

`x`	The object to obtain the hazard function of

`hazard.exp_dist` *Method to obtain the hazard function of an* `exp_dist` *object.*

Description

Method to obtain the hazard function of an `exp_dist` object.

Usage

```
## S3 method for class 'exp_dist'
hazard(x, ...)
```

Arguments

`x`	The `exp_dist` object to obtain the hazard function of

`lomax_series_data_1` *Masked data for a series system with lomax distributed nodes and candidate sets that model m0*

Description

Masked data containing the system lifetime and other attributes of 10000 series system with parameters `lambda=c(3,4,5)` and `kappa=c(2,3,4)`. Every candidate set (of model m0) has `w=2` candidate nodes.

Usage

```
lomax_series_data_1
```

Format

A data frame with 10000 rows and 9 variables:

s Real observable variable, system lifetime

k Integer latent variable, the failed node

w Integer observable variable, number of candidates

t.1 Real latent variable, lifetime of node 1

t.2 Real latent variable, lifetime of node 2

t.3 Real latent variable, lifetime of node 3

c.1 Boolean observable variable, TRUE indicates node 1 is in candidate set

c.2 Boolean observable variable, TRUE indicates node 2 is in candidate set

c.3 Boolean observable variable, TRUE indicates node 3 is in candidate set

Source

https://github.com/queelius/masked.data/blob/master/data-raw/lomax_series_data_1_gen.R

`lomax_series_data_2` *Masked data for a series system with lomax distributed nodes and candidate sets that model m0*

Description

Masked data containing the system lifetime and other attributes of 2000 series system with parameters `lambda=c(1,1.5,.75)` and `kappa=c(2,1.5,2.5)`. Every candidate set (of model m0) has `w=2` candidate nodes.

Usage

```
lomax_series_data_2
```

Format

A data frame with 2000 rows and 9 variables:

s Real observable variable, system lifetime

k Integer latent variable, the failed node

w Integer observable variable, number of candidates

t.1 Real latent variable, lifetime of node 1

t.2 Real latent variable, lifetime of node 2

t.3 Real latent variable, lifetime of node 3

c.1 Boolean observable variable, TRUE indicates node 1 is in candidate set

c.2 Boolean observable variable, TRUE indicates node 2 is in candidate set

c.3 Boolean observable variable, TRUE indicates node 3 is in candidate set

Source

https://github.com/queelius/masked.data/blob/master/data-raw/lomax_series_data_2_gen.R

`make_exp_dist` *Construct exponential distribution object.*

Description

Construct exponential distribution object.

Usage

```
make_exp_dist(rate)
```

Arguments

`rate`	failure rate

`make_exp_series` *Construct exponential series object.*

Description

Construct exponential series object.

Usage

```
make_exp_series(rate)
```

Arguments

`rate`	failure rates

make_normal *Construct (multivariate or univariate) normal distribution object.*

Description

Construct (multivariate or univariate) normal distribution object.

Usage

```
make_normal(mu, sigma = diag(length(mu)))
```

Arguments

mu	mean
sigma	variance-covariance matrix

masked.data *masked.data: A package for estimating parameters from masked data*

Description

The masked.data package provides a general framework for working with masked data and designing functions to solve for the parameter's of latent node lifetime distributions in a series system.

data structures

md_estimate tbl_md

tbl_md tools

md_write_csv

MLE point estimators

md_mle_exp_series_m0 md_mle_exp_series_m1 md_mle_exp_series_m0 point.md_estimate

MLE interval estimators and covariance

vcov.md_estimate info.md_estimate confint.md_estimate

`md_candidates_as_matrix` *Convert the columns corresponding to the candidate matrix to a matrix object.*

Description

Convert the columns corresponding to the candidate matrix to a matrix object.

Usage

```
md_candidates_as_matrix(md)
```

Arguments

`md`	masked data

Value

Candidate sets represented as a Boolean matrix

`md_candidates_to_strings` *Candidate matrix to stringified vector of integers*

Description

Candidate matrix to stringified vector of integers

Usage

```
md_candidates_to_strings(md)
```

Arguments

`md`	masked data

md_candidate_m0 *Candidate model m0*

Description

Decorates masked data object `md` with candidate sets according to candidate model m0.

Usage

```
md_candidate_m0(md, m)
```

Arguments

`md`	masked data, data frame object with column `k` for failed component and column `w` for corresponding candidate set size.
`m`	number of nodes in the series system

Details

Specifically, the candidate sets are generated according to the alpha-masked model, where `C[i,]` contains `k[i]` and `w[i-1]` nodes randomly selected without replacement from `{1,...,m}-{k[i]}`.

Value

masked data with candidate sets that model m0

md_candidate_m1 *Candidate model m1*

Description

Decorates masked data object `md` with candidate sets according to candidate model m1.

Usage

```
md_candidate_m1(md, m)
```

Arguments

`md`	masked data, a data frame object with column 'k' for failed component, column 'w' for corresponding candidate set size, and column 'alpha' for corresponding alpha probabilities
`m`	Integer, number of nodes in the series system

Details

Specifically, the candidate sets are generated according to the alpha-masked model, where with probability `alpha[i]`, `C[i,]` contains `k[i]` and `w[i-1]` nodes randomly selected without replacement from `{1,...,m} -{ k[i]}` and with probability `1-alpha[i]`, `C[i,]` contains `w[i]` nodes randomly selected without replacement from `{1,...,m} -{ k[i] }`.

Value

alpha-masked data with candidate sets that model m1

`md_exp_series` *Generates masked data for a series system with exponentially distributed nodes and candidate sets according to candidate_model.*

Description

Generates masked data for a series system with exponentially distributed nodes and candidate sets according to candidate_model.

Usage

```
md_exp_series(n, theta, w, candidate_model = md_candidate_m0, metadata = T)
```

Arguments

`n`	Integer. The sample size (each row is an observation).
`theta`	Numeric vector. The jth component has a failure rate `theta[j]`.
`w`	Integer vector. For the ith observation, generate `w[j]` candidates.
`candidate_model`	Function that accepts masked data as an argument. The candidate model, defaults to `md_candidate_m0`. If set to `NULL`, then do not generate a candidate set. `md_mle_exp_series` will treat such masked data as a sample that includes every node as candidates.
`metadata`	Boolean. If `TRUE` writes meta-data for series system to attributes of masked data.

Value

masked data, a data frame of n observations, `(s,k,t1,...,tm,c1,...,cm)` where `k`, `t`, and `c` are covariates (or predictors) of `s,k,t1,...,tm`.

Examples

```
md_exp_series(n=10,theta=c(1,2,3),w=rep(2,10))
```

`md_exp_series_node_failure_m0`

Constructs a pdf object for the conditional node failure in an exponential series system according to candidate model m0, `f(k|c,s) = h_k(s)/h(s) I(k in c)`.

Description

This simplifies to `f(k|c) = theta[k] / sum(theta[j],j in c)` for the exponential series system.

Usage

```
md_exp_series_node_failure_m0(theta)
```

Arguments

`theta`	parameter value of `exp_series`

`md_exp_series_system_failure_interval_m0`

Constructs the shortest interval for the system lifetime given a candidate set under model m0 with a probability `p` *that the interval contains the system failure.*

Description

Constructs the shortest interval for the system lifetime given a candidate set under model m0 with a probability p that the interval contains the system failure.

Usage

```
md_exp_series_system_failure_interval_m0(theta, p)
```

Arguments

`theta`	parameter value of `exp_series`
`p`	probability that system failure time is in the computed interval

`md_fisher_scoring` *Fisher scoring algorithm.*

Description

Fisher scoring algorithm.

Usage

```
md_fisher_scoring(theta0, info, score, eps = 1e-05, max_iterations = 10000L)
```

Arguments

`theta0`	initial guess of theta with p components
`info`	information matrix function of type $R^p -> R^{p \times q}$
`score`	score function of type $R^p -> R^p$
`eps`	stopping condition
`max_iterations`	maximum number of iterations

Algorithm

The algorithm is straightforward. Details here.

`md_info_exp_series_m0` *Information matrix (observed) for rate parameter with respect to masked data of a series system with exponentially distributed lifetimes and candidate model m0.*

Description

Information matrix (observed) for rate parameter with respect to masked data of a series system with exponentially distributed lifetimes and candidate model m0.

Usage

```
md_info_exp_series_m0(md)
```

Arguments

`md`	masked data for candidate model m0

Value

observed information matrix of type R^m -> R^(m x m)

md_info_lomax_series_m0 *Observed information matrix of the rate parameter of the series system with exponentially distributed component lifetimes given masked data with candidate sets according to model m0.*

Description

Observed information matrix of the rate parameter of the series system with exponentially distributed component lifetimes given masked data with candidate sets according to model m0.

Usage

```
md_info_lomax_series_m0(md)
```

Arguments

md	masked data

Value

observed info

md_is_masked_data *Test whether* x *is masked data*

Description

An object is considered to be masked data if it is a type of data frame (e.g., tibble) and it has at least two columns for candidate sets named `c.1` and `c.2`.

Usage

```
md_is_masked_data(x)
```

Arguments

x	object to test

`md_kloglike_exp_series_m0`
Kernel log-likelihood for masked data m0 for exponential series system using sufficient statistics.

Description

The log of the kernel of the likelihood function for masked data for a series system with exponentially distributed lifetimes and candidate sets that model m0 using sufficient statistics.

Usage

```
md_kloglike_exp_series_m0(md)
```

Arguments

`md`	masked data

`md_kloglike_lomax_series_m0_ref`
Kernel log-likelihood for masked data m0 for lomax series system.

Description

The log of the kernel of the likelihood function for masked data for a series system with lomax distributed lifetimes and candidate sets that model m0.

Usage

```
md_kloglike_lomax_series_m0_ref(md)
```

Arguments

`md`	masked data for candidate model m0

Details

This is the unoptimized version, which serves as a ground-truth for testing a more efficient implementation.

`md_lomax_series` *Generates masked data for a series system with lomax distributed nodes and candidate sets according to candidate_model.*

Description

Generates masked data for a series system with lomax distributed nodes and candidate sets according to candidate_model.

Usage

```
md_lomax_series(
  n,
  lambda,
  kappa,
  w,
  candidate_model = md_candidate_m0,
  metadata = T
)
```

Arguments

`n`	Integer. The sample size (each row is an observation).
`lambda`	Numeric vector.
`kappa`	Numeric vector. The jth node is parameterized by theta_j := (lambda_j,kappa_j).
`w`	Integer vector. For the ith observation, generate w_j candidates.
`candidate_model`	Function that accepts masked data as an argument. The candidate model, defaults to md_candidate_m0. If set to NULL, then do not generate a candidate set. md_mle_exp_series will treat such masked data as a sample that includes every node as candidates.
`metadata`	Boolean. If TRUE writes meta-data for series system to attributes of masked data (tbl_md).

Value

masked data, a data frame of n observations, (s,k,t1,...,tm,c1,...,cm) where k, t, and c are covariates (or predictors) of s,k,t1,...,tm.

Examples

```
md_lomax_series(n=10,lambda=c(1,2,3),kappa=c(4,5,6),w=rep(2,10))
```

`md_mle_exp_series_m0` *Maximum likelihood estimator of the parameters of a series system with nodes that have exponentially distributed lifetimes given a sample of masked data according to candidate model m0.*

Description

Maximum likelihood estimator of the parameters of a series system with nodes that have exponentially distributed lifetimes given a sample of masked data according to candidate model m0.

Usage

```
md_mle_exp_series_m0(md, theta0 = NULL, eps = 1e-05, max_iterations = 10000L)
```

Arguments

`md`	masked data
`theta0`	initial guess for MLE
`eps`	stopping condition
`max_iterations`	stop if iterations reaches max_iterations.

Value

MLE estimate

`md_node_times_as_matrix` *Convert the columns corresponding to the node times matrix to a matrix object.*

Description

Convert the columns corresponding to the node times matrix to a matrix object.

Usage

```
md_node_times_as_matrix(md)
```

Arguments

`md`	masked data

Value

Node times represented as a real matrix

`md_num_nodes` *Retrieve the number of nodes implicitly defined by the masked data input 'md'.*

Description

Retrieve the number of nodes implicitly defined by the masked data input 'md'.

Usage

```
md_num_nodes(md)
```

Arguments

`md`	masked data

Value

number of nodes in the series system

`md_read_json` *Read masked data from a JSON file. If the JSON file has a 'dataset' field, then each member of this field is assumed to refer to a CSV file to read a masked data sample from.*

Description

Any metadata in the JSON file is inserted into the attributes of the masked data samples.

Usage

```
md_read_json(filename)
```

Arguments

`filename`	filename for csv

Value

list of masked data objects

`md_score_exp_series_m0` *score function of masked data for a series system with exponentially distributed lifetimes.*

Description

score function of masked data for a series system with exponentially distributed lifetimes.

Usage

```
md_score_exp_series_m0(md)
```

Arguments

`md`	masked data for candidate model m0

Value

score function of type R^m -> R

`md_series_data` *Generates masked data for a series system with the given node failure times* `t`*, candidate set model* `candidate_model`*, and candidate set sizes* `w`*.*

Description

Generates masked data for a series system with the given node failure times `t`, candidate set model `candidate_model`, and candidate set sizes `w`.

Usage

```
md_series_data(t, w, candidate_model = md_candidate_m0)
```

Arguments

`t`	matrix of node failure times
`w`	Integer vector. For the ith observation, generate w_j candidates.
`candidate_model`	Function that accepts masked data as an argument. The candidate model, defaults to md_candidate_m0. If set to NULL, then do not generate a candidate set. md_mle_exp_series will treat such masked data as a sample that includes every node as candidates.

Value

masked data, a data frame of n observations, (s,k,t1,...,tm,c1,...,cm) where k, t, and c are covariates (or predictors) of s,k,t1,...,tm.

`md_series_node_failure_decorator_m0`

Decorate masked data (tbl_md) with node failure probabilities.

Description

Under model m0, we do not know which node caused the failure, (note: if |C|=1, under m0 we know precisely which node failed), but if we have an estimate (or know) theta, then we may construct f(k|s,c) and compute the node failure probabilities in a masked data object `md`.

Usage

```
md_series_node_failure_decorator_m0(md, fk)
```

Arguments

`md`	masked data
`fk`	pdf f(k|s,c)

Details

We decorate masked data `md` with an estimate of the probabilities, `f(k|s,c)` for `k=1,...,k=m` and return the result.

`md_series_system_failure_decorator_m0`

Decorate masked data (tbl_md) with node failure probabilities.

Description

Under model m0, we do not know which node caused the failure, (note: if |C|=1, under m0 we know precisely which node failed), but if we have an estimate (or know) theta, then we may construct f(k|s,c) and compute the node failure probabilities in a masked data object `md`.

Usage

```
md_series_system_failure_decorator_m0(md, q)
```

Arguments

`md`	masked data
`q`	interval computer for s|c

Details

We decorate masked data `md` with an estimate of the probabilities, `f(k|s,c)` for `k=1,...,k=m` and return the result.

`md_write_csv` *Write masked data data frame (tibble) object to a CSV (comma separated file), optionally writing associated meta-data to a JSON file. In particular, meta-data in this case is defined as the attributes of the data frame object.*

Description

Write masked data data frame (tibble) object to a CSV (comma separated file), optionally writing associated meta-data to a JSON file. In particular, meta-data in this case is defined as the attributes of the data frame object.

Usage

```
md_write_csv(md, filename, write.metadata = T)
```

Arguments

`md`	a masked data frame
`filename`	filename for csv
`write.metadata`	write a separate

`num_nodes` *Method for obtaining the number of nodes in an object.*

Description

Method for obtaining the number of nodes in an object.

Usage

```
num_nodes(series)
```

Arguments

`series`	The object to obtain the number of nodes of

params *Generic method for obtaining the parameters of a parametric distribution.*

Description

Generic method for obtaining the parameters of a parametric distribution.

Usage

```
params(x, ...)
```

Arguments

x	The object to obtain the parameters of

params.normal *Method for obtaining the parameters of a* normal *object.*

Description

Method for obtaining the parameters of a normal object.

Usage

```
## S3 method for class 'normal'
params(x, ...)
```

Arguments

x	The object to obtain the parameters of

params.series *Method for obtaining the parameters of a* series *distribution object.*

Description

Method for obtaining the parameters of a series distribution object.

Usage

```
## S3 method for class 'series'
params(x, ...)
```

Arguments

x	The series object to obtain the parameters of

pdf *Generic method for obtaining the pdf function of a random variable.*

Description

Generic method for obtaining the pdf function of a random variable.

Usage

```
pdf(x, ...)
```

Arguments

x	The object to obtain the hazard function of

pdf.exp_dist *Method to obtain the pdf of an* exp_dist *object.*

Description

Note that since exp_series is also exponentially distributed, this works for that too.

Usage

```
## S3 method for class 'exp_dist'
pdf(x, ...)
```

Arguments

x	The object to obtain the pdf of

point *Generic method for obtaining the point estimate of an estimator.*

Description

Generic method for obtaining the point estimate of an estimator.

Usage

```
point(x, ...)
```

Arguments

x	The object to obtain the point estimate of

point.md_estimate *Method to obtain the point estimate of a masked data estimator,* md_estimate.

Description

Method to obtain the point estimate of a masked data estimator, md_estimate.

Usage

```
## S3 method for class 'md_estimate'
point(x, ...)
```

Arguments

x	The md_estimate object to obtain the point estimate of

print.tbl_md *Print method for masked data (tbl_md).*

Description

Print method for masked data (tbl_md).

Usage

```
## S3 method for class 'tbl_md'
print(x, pprint = F, drop_latent = F, ...)
```

Arguments

x	masked data to print
pprint	Boolean, show candidates as a string column
drop_latent	Boolean, drop the latent random variables

sampler *Generic method for sampling from distribution objects.*

Description

Generic method for sampling from distribution objects.

Usage

```
sampler(x, ...)
```

Arguments

x	The object to sample from.

sampler.exp_dist *Method to sample from an* exp_dist *object.*

Description

Method to sample from an exp_dist object.

Usage

```
## S3 method for class 'exp_dist'
sampler(x, ...)
```

Arguments

x	The exp_dist object to sample from.

sampler.md_estimate *Method to obtain the sampler for an* md_estimate *object.*

Description

Method to obtain the sampler for an md_estimate object.

Usage

```
## S3 method for class 'md_estimate'
sampler(x, ...)
```

Arguments

x	The md_estimate object to create a sampling procedure from

sampler.normal *Method for sampling from a* normal *object.*

Description

Method for sampling from a normal object.

Usage

```
## S3 method for class 'normal'
sampler(x, ...)
```

Arguments

x	The object to sample from

vcov.exp_dist *Method for obtaining the variance of a* exp_dist *object.*

Description

Method for obtaining the variance of a exp_dist object.

Usage

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

Arguments

object	The exp_dist object to obtain the variance of

vcov.exp_series — *Method for obtaining the variance-covariance of a* exp_series *object.*

Description

Method for obtaining the variance-covariance of a exp_series object.

Usage

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

Arguments

object	The exp_seriesThe object to obtain the variance of

vcov.md_estimate — *Compute the covariance matrix from the given masked data estimate.*

Description

Sampling distribution of the MLE is a multivariate normal with mean given by the true parameter value and, asymptotically, a covariance given by the inverse of the Fisher information matrix.

Usage

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

Arguments

object	The variance-covariance matrix of the estimator to obtain

`vcov.normal` *Retrieve the variance-covariance matrix (or scalar) of a* `normal` *object.*

Description

Retrieve the variance-covariance matrix (or scalar) of a `normal` object.

Usage

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

Arguments

`object`	The `normal` object to retrieve the variance-covariance matrix from

Index