

Package ‘algebraic.dist’

December 17, 2025

Title Algebra over Probability Distributions

Version 0.1.0

Description Provides an algebra over probability distributions enabling composition, sampling, and automatic simplification to closed forms. Supports normal, exponential, multivariate normal, and empirical distributions with operations like addition and subtraction that automatically simplify when mathematical identities apply (e.g., the sum of independent normal distributions is normal). Uses S3 classes for distributions and R6 for support objects.

License GPL (>= 3)

Encoding UTF-8

RoxygenNote 7.3.3

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

Config/testthat/edition 3

Imports stats, mvtnorm, R6

Depends R (>= 3.5.0)

VignetteBuilder knitr

URL <https://github.com/queelius/algebraic.dist>,
<https://queelius.github.io/algebraic.dist/>

BugReports <https://github.com/queelius/algebraic.dist/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 2025-12-17 10:50:02 UTC

Contents

<code>+.dist</code>	4
<code>-.dist</code>	5
<code>cdf</code>	5
<code>cdf.empirical_dist</code>	6
<code>cdf.exponential</code>	6
<code>cdf.mvn</code>	7
<code>cdf.normal</code>	7
<code>conditional</code>	8
<code>conditional.dist</code>	8
<code>conditional.empirical_dist</code>	9
<code>density.empirical_dist</code>	9
<code>density.exponential</code>	10
<code>density.mvn</code>	10
<code>density.normal</code>	11
<code>dim.empirical_dist</code>	11
<code>dim.exponential</code>	12
<code>dim.finite_set</code>	12
<code>dim.interval</code>	13
<code>dim.mvn</code>	13
<code>dim.normal</code>	14
<code>edist</code>	14
<code>empirical_dist</code>	15
<code>expectation</code>	15
<code>expectation.dist</code>	16
<code>expectation.empirical_dist</code>	16
<code>expectation.univariate_dist</code>	17
<code>expectation_data</code>	18
<code>exponential</code>	19
<code>finite_set</code>	19
<code>has</code>	20
<code>has.finite_set</code>	21
<code>has.interval</code>	21
<code>hazard</code>	22
<code>hazard.exponential</code>	22
<code>infimum</code>	23
<code>infimum.finite_set</code>	23
<code>infimum.interval</code>	23
<code>interval</code>	24
<code>inv_cdf</code>	25
<code>inv_cdf.exponential</code>	26
<code>inv_cdf.normal</code>	26
<code>is_dist</code>	27
<code>is_edist</code>	27
<code>is_empirical_dist</code>	28
<code>is_exponential</code>	28
<code>is_mvn</code>	28

is_normal	29
marginal	29
marginal.empirical_dist	30
marginal.mvn	30
mean.edist	31
mean.empirical_dist	31
mean.exponential	32
mean.mvn	32
mean.normal	33
mean.univariate_dist	33
mvn	34
nobs.empirical_dist	34
normal	35
nparams	35
nparams.empirical_dist	36
obs	36
obs.empirical_dist	37
params	37
params.edist	38
params.empirical_dist	38
params.exponential	39
params.mvn	39
params.normal	40
print.edist	40
print.empirical_dist	41
print.exponential	41
print.interval	42
print.mvn	42
print.normal	43
print.summary_dist	43
rmap	44
rmap.dist	44
rmap.empirical_dist	45
rmap.mvn	45
sampler	46
sampler.default	46
sampler.edist	47
sampler.empirical_dist	47
sampler.exponential	48
sampler.mvn	48
sampler.normal	49
sample_mvn_region	49
simplify	50
simplify.dist	50
simplify.edist	51
summary.dist	51
summary_dist	52
sup	53

sup.empirical_dist	53
sup.exponential	54
sup.mvn	54
sup.normal	55
supremum	55
supremum.finite_set	56
supremum.interval	56
surv	56
surv.exponential	57
vcov.default	57
vcov.edist	58
vcov.empirical_dist	58
vcov.exponential	59
vcov.mvn	59
vcov.normal	60
vcov.univariate_dist	60
Index	61

<i>+.dist</i>	<i>Method for adding dist objects.</i>
---------------	--

Description

Creates an expression distribution and automatically simplifies to closed form when possible (e.g., normal + normal = normal).

Usage

```
## S3 method for class 'dist'  
x + y
```

Arguments

- x The first dist object to add
- y The second dist object to add

Value

A simplified distribution or edist if no closed form exists

-.dist

*Method for negation or subtraction of dist objects.***Description**

Unary: returns negated distribution (e.g., $-N(\mu, \text{var}) = N(-\mu, \text{var})$) Binary: creates expression distribution and simplifies to closed form when possible (e.g., normal - normal = normal).

Usage

```
## S3 method for class 'dist'
x - y
```

Arguments

x The first dist object
y The second dist object (optional for unary negation)

Value

A simplified distribution or edist if no closed form exists

cdf

*Generic method for obtaining the cdf of an object.***Description**

Generic method for obtaining the cdf of an object.

Usage

```
cdf(x, ...)
```

Arguments

x The object to obtain the cdf of.
... Additional arguments to pass.

cdf.empirical_dist	<i>Method for obtaining the cdf of empirical_dist object x.</i>
--------------------	---

Description

If x is a multivariate empirical distribution, this function will throw an error. It's only defined for univariate empirical distributions.

Usage

```
## S3 method for class 'empirical_dist'
cdf(x, ...)
```

Arguments

x	The empirical distribution object.
\dots	Additional arguments to pass (not used)

Value

A function that takes a numeric vector t and returns the empirical cdf of x evaluated at t .

cdf.exponential	<i>Method to obtain the cdf of an exponential object.</i>
-----------------	---

Description

Method to obtain the cdf of an exponential object.

Usage

```
## S3 method for class 'exponential'
cdf(x, ...)
```

Arguments

x	The object to obtain the pdf of
\dots	Additional arguments (not used)

Value

A function that computes the cdf of the exponential. Accepts as input a vector t at which to compute the cdf, an input rate denoting the failure rate of the exponential distribution, and a logical `log` indicating whether to compute the log of the cdf. By default, `rate` is the failure rate of object x .

cdf.mvn	<i>Method for obtaining the CDF of a mvn object.</i>
---------	--

Description

Method for obtaining the CDF of a mvn object.

Usage

```
## S3 method for class 'mvn'  
cdf(x, ...)
```

Arguments

x	The object to obtain the CDF of
...	Additional arguments to pass (not used)

cdf.normal	<i>Method for obtaining the cdf of an normal object.</i>
------------	--

Description

Method for obtaining the cdf of an normal object.

Usage

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

Arguments

x	The object to obtain the cdf of
...	Additional arguments to pass (not used)

Value

A function that computes the cdf of the normal distribution. It accepts as input a parameter vector `q`, a mean vector `mu`, a variance `var`, and a `log` argument determining whether to compute the log of the cdf. By default, `mu` and `var` are the mean and variance of object `x` and `log` is `FALSE`. Finally, it accepts additional arguments `...` to pass to the `pnorm` function.

conditional	<i>Generic method for obtaining the conditional distribution of a distribution object x given condition P.</i>
-------------	--

Description

Generic method for obtaining the conditional distribution of a distribution object x given condition P.

Usage

```
conditional(x, P, ...)
```

Arguments

x	The empirical distribution object.
P	The predicate function to condition x on
...	additional arguments to pass into P

conditional.dist	<i>Method for obtaining the condition distribution, $x \mid P(x)$, of dist object x.</i>
------------------	---

Description

We just sample from x and place the sample in empirical_dist and then condition on it.

Usage

```
## S3 method for class 'dist'
conditional(x, P, n = 10000L, ...)
```

Arguments

x	The distribution object.
P	The predicate function to condition the distribution on
n	The number of samples to generate for the MC estimate of the conditional distribution $x \mid P$. Defaults to 10000.
...	additional arguments to pass into P.

conditional.empirical_dist

Method for obtaining the condition distribution, $x \mid P(x)$, of empirical_dist object x.

Description

In other words, we condition the data on the predicate function. In order to do so, we simply remove all rows from the data that do not satisfy the predicate P. For instance, if we have a 2-dimensional distribution, and we want to condition on the first dimension being greater than the second dimension, we would do the following:

Usage

```
## S3 method for class 'empirical_dist'
conditional(x, P, ...)
```

Arguments

x	The empirical distribution object.
P	The predicate function to condition the data on.
...	additional arguments to pass into P.

Details

```
x_cond <- conditional(x, function(d) d[1] > d[2])
```

This would return a new empirical distribution object with the same dimensionality as x, but with all rows where the first dimension is less than or equal to the second dimension removed.

density.empirical_dist

Method for obtaining the pdf of a empirical_dist object.

Description

Method for obtaining the pdf of a empirical_dist object.

Usage

```
## S3 method for class 'empirical_dist'
density(x, ...)
```

Arguments

x	The object to obtain the pdf of.
...	Additional arguments to pass into the pdf function.

Note

sort tibble lexicographically and do a binary search to find upper and lower bound in $\log(\text{nobs}(x))$ time.

density.exponential	<i>Method to obtain the pdf of an exponential object.</i>
---------------------	---

Description

Method to obtain the pdf of an exponential object.

Usage

```
## S3 method for class 'exponential'
density(x, ...)
```

Arguments

x	The object to obtain the pdf of
...	Additional arguments (not used)

Value

A function that computes the pdf of the exponential distribution at a given point t . Also accepts a rate argument that determines the failure rate of the exponential distribution (defaults to the failure rate of object x) and a log argument that determines whether to compute the log of the pdf.

density.mvn	<i>Function generator for obtaining the pdf of an mvn object (multivariate normal).</i>
-------------	---

Description

Function generator for obtaining the pdf of an mvn object (multivariate normal).

Usage

```
## S3 method for class 'mvn'
density(x, ...)
```

Arguments

x	The mvn (S3) object to obtain the pdf (density) of
...	Additional arguments to pass into the generated function.

Value

A function that computes the pdf of the mvn distribution. It accepts as input: - obs: vector or matrix of quantiles. when x is a matrix, each row is taken to be a quantile and columns correspond to the number of dimensions, p. - mu: a vector denoting the population mean. Defaults to the mean of x (an mvn object) - sigma: a matrix denoting the variance-covariance of observations. Defaults to the variance-covariance of x. - log: logical, determines whether to compute the log of the pdf. Defaults to FALSE. - ...: any additional parameters to pass to dmnorm.

density.normal	<i>Method for obtaining the pdf of an normal object.</i>
----------------	--

Description

Method for obtaining the pdf of an normal object.

Usage

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

Arguments

x	The object to obtain the pdf of
...	Additional arguments to pass (not used)

Value

A function that computes the pdf of the normal distribution. It accepts as input a parameter vector x, a mean vector mu, a variance-covariance matrix var, and a log argument determining whether to compute the log of the pdf. By default, mu and var are the mean and variance of object x.

dim.empirical_dist	<i>Method for obtaining the dimension of a empirical_dist object.</i>
--------------------	---

Description

Method for obtaining the dimension of a empirical_dist object.

Usage

```
## S3 method for class 'empirical_dist'
dim(x)
```

Arguments

x	The object to obtain the dimension of.
---	--

dim.exponential	<i>Method to obtain the dimension of an exponential object.</i>
-----------------	---

Description

Method to obtain the dimension of an exponential object.

Usage

```
## S3 method for class 'exponential'  
dim(x)
```

Arguments

x	The exponential object to obtain the dimension of
---	---

Value

The dimension of the exponential object

dim.finite_set	<i>Return the dimension of the finite set.</i>
----------------	--

Description

Return the dimension of the finite set.

Usage

```
## S3 method for class 'finite_set'  
dim(x)
```

Arguments

x	A finite set.
---	---------------

dim.interval	<i>Return the dimension of the interval.</i>
--------------	--

Description

Return the dimension of the interval.

Usage

```
## S3 method for class 'interval'  
dim(x)
```

Arguments

x	An interval object.
---	---------------------

dim.mvn	<i>Method for obtaining the dimension of an mvn object.</i>
---------	---

Description

Method for obtaining the dimension of an mvn object.

Usage

```
## S3 method for class 'mvn'  
dim(x)
```

Arguments

x	The object to obtain the dimension of
---	---------------------------------------

Value

The dimension of the mvn object

dim.normal	<i>Method for obtaining the dimension of a normal object.</i>
------------	---

Description

Method for obtaining the dimension of a normal object.

Usage

```
## S3 method for class 'normal'
dim(x)
```

Arguments

x The normal object to obtain the dimension of

Value

The dimension of the normal object

edist	<i>Takes an expression e and a list vars and returns a lazy edist (expression distribution object), that is a subclass of dist that can be used in place of a dist object.</i>
-------	--

Description

Takes an expression e and a list vars and returns a lazy edist (expression distribution object), that is a subclass of dist that can be used in place of a dist object.

Usage

```
edist(e, vars)
```

Arguments

e the expression to evaluate against the arguments.
vars the list of distributions (with variable names) to evaluate the expression e against.

Value

An edist object.

empirical_dist	<i>Construct empirical distribution object.</i>
----------------	---

Description

Construct empirical distribution object.

Usage

```
empirical_dist(data)
```

Arguments

data	data to construct empirical distribution from. if matrix or data frame, each row is a joint observation, if a vector, each element is an observation. whatever data is, it must be convertible to a tibble.
------	---

expectation	<i>Generic method for obtaining the expectation of f with respect to x.</i>
-------------	---

Description

Generic method for obtaining the expectation of f with respect to x.

Usage

```
expectation(x, g, ...)
```

Arguments

x	The distribution object.
g	The function to take the expectation of.
...	Additional arguments to pass into g.

expectation.dist	<i>Expectation of a Function Applied to a dist Object</i>
------------------	---

Description

Expectation operator applied to x of type `dist` with respect to a function g . Optionally, constructs a confidence interval for the expectation estimate using the Central Limit Theorem.

Usage

```
## S3 method for class 'dist'
expectation(x, g = function(t) t, ..., control = list())
```

Arguments

<code>x</code>	A <code>dist</code> object.
<code>g</code>	Characteristic function of interest, defaults to identity.
<code>...</code>	Additional arguments to pass to g .
<code>control</code>	A list of control parameters: <code>compute_stats</code> - Logical, whether to compute CIs for the expectations, defaults to <code>FALSE</code> <code>n</code> - Integer, the number of samples to use for the MC estimate, defaults to 10000 <code>alpha</code> - Real, the significance level for the confidence interval, defaults to 0.05

Value

If `compute_stats` is `FALSE`, then the estimate of the expectation, otherwise a list with the following components: `value` - The estimate of the expectation `ci` - The confidence intervals for each component of the expectation `n` - The number of samples

expectation.empirical_dist	<i>Method for obtaining the expectation of empirical_dist object x under function g.</i>
----------------------------	--

Description

Method for obtaining the expectation of `empirical_dist` object x under function g .

Usage

```
## S3 method for class 'empirical_dist'
expectation(x, g = function(t) t, ..., control = list())
```

Arguments

x	The distribution object.
g	The function to take the expectation of.
...	Additional arguments to pass into function g.
control	a list of control parameters: compute_stats - Whether to compute CIs for the expectations, defaults to FALSE n - The number of samples to use for the MC estimate, defaults to 10000 alpha - The significance level for the confidence interval, defaults to 0.05

Value

If compute_stats is FALSE, then the estimate of the expectation, otherwise a list with the following components: value - The estimate of the expectation ci - The confidence intervals for each component of the expectation n - The number of samples

expectation.univariate_dist

Method for obtaining the expectation of f with respect to a univariate_dist object x.

Description

Assumes the support is a contiguous interval that has operations for retrieving the lower and upper bounds.

Usage

```
## S3 method for class 'univariate_dist'
expectation(x, g, ..., control = list())
```

Arguments

x	The distribution object.
g	The function to take the expectation of.
...	Additional arguments to pass into g.
control	An (optional) list of control parameters for integrate or expectation_data (if x is not continuous)

expectation_data	<i>Function used for computing expectations given data (e.g., from an MC simulation or bootstrap). it expects a matrix, or something that can be coerced to a matrix (e.g., a data frame). it also expects a function g to apply to each row of the data, and returns the expectation of g under the empirical distribution of the data. it also returns a confidence interval for the expectation, and the number of samples used to compute the expectation.</i>
------------------	--

Description

example: `expectation_data(D, function(x) (x-colMeans(D)) %*% t(x-colMeans(D)))` computes the covariance of the data D, except the matrix structure is lost (it's just a vector, which can be coerced back to a matrix if needed).

Usage

```
expectation_data(
  data,
  g = function(x) x,
  ...,
  compute_stats = TRUE,
  alpha = 0.05
)
```

Arguments

data	a matrix of data
g	a function to apply to each row of the data
...	additional arguments to pass to g
compute_stats	whether to compute CIs for the expectations
alpha	the confidence level for the confidence interval for each component of the expectation (if compute_stats is TRUE)

Value

if compute_stats is TRUE, then a list with the following components: value - The estimate of the expectation ci - The confidence intervals for each component of the expectation n - The number of samples otherwise, just the value of the expectation.

exponential	<i>Construct exponential distribution object.</i>
-------------	---

Description

Construct exponential distribution object.

Usage

```
exponential(rate)
```

Arguments

rate	failure rate
------	--------------

finite_set	<i>Finite set</i>
------------	-------------------

Description

A finite set. It also satisfies the concept of a support.

Public fields

values A vector of values.

Methods**Public methods:**

- `finite_set$new()`
- `finite_set$has()`
- `finite_set$infimum()`
- `finite_set$supremum()`
- `finite_set$dim()`
- `finite_set$clone()`

Method `new()`: Initialize a finite set.

Usage:

```
finite_set$new(values)
```

Arguments:

values A vector of values.

Method `has()`: Determine if a value is contained in the finite set.

Usage:

```
finite_set$has(x)
```

Arguments:

x A vector of values.

Method `infimum()`: Get the infimum of the finite set.

Usage:

```
finite_set$infimum()
```

Returns: A numeric vector of infimums.

Method `supremum()`: Get the supremum of the finite set.

Usage:

```
finite_set$supremum()
```

Returns: A numeric vector of supremums.

Method `dim()`: Get the dimension of the finite set.

Usage:

```
finite_set$dim()
```

Returns: The dimension of the finite set.

Method `clone()`: The objects of this class are cloneable with this method.

Usage:

```
finite_set$clone(deep = FALSE)
```

Arguments:

deep Whether to make a deep clone.

has

Support

Description

support is a class that represents the support of a random element or distribution, i.e. the set of values that it realize.

It's a conceptual class. To satisfy the concept of a support, the following methods must be implemented:

1. `has`: a function that returns a logical vector indicating whether each value in a vector is contained in the support
2. `infimum`: a function that returns the infimum of the support
3. `supremum`: a function that returns the supremum of the support
4. `dim`: a function that returns the dimension of the support

We provide two implementations that satisfy the concept:

- `interval`: a support that is an infinite set of contiguous numeric values
- `finite_set`: a support that is a finite set of values Determine if a value is contained in the support.

Usage

```
has(object, x)
```

Arguments

object	A support object.
x	A vector of values.

has.finite_set	<i>Determine if a value is contained in the finite set.</i>
----------------	---

Description

Determine if a value is contained in the finite set.

Usage

```
## S3 method for class 'finite_set'
has(object, x)
```

Arguments

object	A finite set.
x	A vector of values.

has.interval	<i>Determine if a value is contained in the interval.</i>
--------------	---

Description

Determine if a value is contained in the interval.

Usage

```
## S3 method for class 'interval'
has(object, x)
```

Arguments

object	An interval object.
x	A vector of values.

hazard	<i>Generic method for obtaining the hazard function of an object.</i>
--------	---

Description

Generic method for obtaining the hazard function of an object.

Usage

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

Arguments

x	The object to obtain the hazard function of.
...	Additional arguments to pass.

hazard.exponential	<i>Method to obtain the hazard function of an exponential object.</i>
--------------------	---

Description

Method to obtain the hazard function of an exponential object.

Usage

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

Arguments

x	The exponential object to obtain the hazard function of
...	Additional arguments (not used)

Value

A function that computes the hazard function of the exponential distribution at a given point *t* and rate *rate*. By default, *rate* is the failure rate of object *x*. Also accepts a *log* argument that determines whether to compute the log of the hazard function.

infimum	<i>Get the infimum of the support.</i>
---------	--

Description

Get the infimum of the support.

Usage

```
infimum(object)
```

Arguments

object	A support object.
--------	-------------------

infimum.finite_set	<i>Return the infimum of the finite set.</i>
--------------------	--

Description

Return the infimum of the finite set.

Usage

```
## S3 method for class 'finite_set'
infimum(object)
```

Arguments

object	A finite set.
--------	---------------

infimum.interval	<i>Return the (vector of) infimum of the interval.</i>
------------------	--

Description

Return the (vector of) infimum of the interval.

Usage

```
## S3 method for class 'interval'
infimum(object)
```

Arguments

object	An interval object.
--------	---------------------

interval

*Interval***Description**

An interval is a support that is a finite union of intervals.

Public fields

lower A numeric vector of lower bounds.

upper A numeric vector of upper bounds.

lower_closed A logical vector indicating whether the lower bound is closed.

upper_closed A logical vector indicating whether the upper bound is closed.

Methods**Public methods:**

- `interval$new()`
- `interval$is_empty()`
- `interval$has()`
- `interval$infimum()`
- `interval$supremum()`
- `interval$dim()`
- `interval$clone()`

Method `new()`: Initialize an interval.

Usage:

```
interval$new(
  lower = -Inf,
  upper = Inf,
  lower_closed = FALSE,
  upper_closed = FALSE
)
```

Arguments:

lower A numeric vector of lower bounds.

upper A numeric vector of upper bounds.

lower_closed A logical vector indicating whether the lower bound is closed.

upper_closed A logical vector indicating whether the upper bound is closed.

Method `is_empty()`: Determine if the interval is empty

Usage:

```
interval$is_empty()
```

Returns: A logical vector indicating whether the interval is empty.

Method `has()`: Determine if a value is contained in the interval.

Usage:

`interval$has(x)`

Arguments:

`x` A numeric vector of values.

Returns: A logical vector indicating whether each value is contained

Method `infimum()`: Get the infimum of the interval.

Usage:

`interval$infimum()`

Returns: A numeric vector of infimums.

Method `supremum()`: Get the supremum of the interval.

Usage:

`interval$supremum()`

Returns: A numeric vector of supremums.

Method `dim()`: Get the dimension of the interval.

Usage:

`interval$dim()`

Returns: The dimension of the interval.

Method `clone()`: The objects of this class are cloneable with this method.

Usage:

`interval$clone(deep = FALSE)`

Arguments:

`deep` Whether to make a deep clone.

inv_cdf

Generic method for obtaining the quantile (inverse cdf) of an object.

Description

Generic method for obtaining the quantile (inverse cdf) of an object.

Usage

`inv_cdf(x, ...)`

Arguments

<code>x</code>	The object to obtain the quantile of.
<code>...</code>	Additional arguments to pass.

inv_cdf.exponential	<i>Method to obtain the inverse cdf of an exponential object.</i>
---------------------	---

Description

Method to obtain the inverse cdf of an exponential object.

Usage

```
## S3 method for class 'exponential'  
inv_cdf(x, ...)
```

Arguments

x	The object to obtain the inverse cdf of
...	Additional arguments (not used)

Value

A function that computes the inverse cdf of the exponential distribution. Accepts as input a vector p probabilities to compute the inverse cdf, a rate value denoting the failure rate of the exponential distribution, and a logical log.p indicating whether input p denotes probability or log-probability. By default, rate is the failure rate of object x.

inv_cdf.normal	<i>Method for obtaining the inverse cdf of an normal object.</i>
----------------	--

Description

Method for obtaining the inverse cdf of an normal object.

Usage

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

Arguments

x	The object to obtain the inverse cdf of
...	Additional arguments to pass (not used)

Value

A function that computes the inverse cdf of the normal distribution.

is_dist	<i>Function to determine whether an object x is a dist object.</i>
---------	--

Description

Function to determine whether an object x is a dist object.

Usage

```
is_dist(x)
```

Arguments

x	The object to test
---	--------------------

Value

Logical indicating whether x is a dist object.

is_edist	<i>Function to determine whether an object x is an edist object.</i>
----------	--

Description

Function to determine whether an object x is an edist object.

Usage

```
is_edist(x)
```

Arguments

x	The object to test
---	--------------------

is_empirical_dist	<i>Function to determine whether an object x is an empirical_dist object.</i>
-------------------	---

Description

Function to determine whether an object x is an empirical_dist object.

Usage

```
is_empirical_dist(x)
```

Arguments

x	The object to test
---	--------------------

is_exponential	<i>Function to determine whether an object x is an exponential object.</i>
----------------	--

Description

Function to determine whether an object x is an exponential object.

Usage

```
is_exponential(x)
```

Arguments

x	The object to test
---	--------------------

is_mvn	<i>Function to determine whether an object x is an mvn object.</i>
--------	--

Description

Function to determine whether an object x is an mvn object.

Usage

```
is_mvn(x)
```

Arguments

x	The object to test
---	--------------------

is_normal	<i>Function to determine whether an object x is an normal object.</i>
-----------	---

Description

Function to determine whether an object x is an normal object.

Usage

```
is_normal(x)
```

Arguments

x	The object to test
---	--------------------

marginal	<i>Generic method for obtaining the marginal distribution of a distribu- tion object x over components indices.</i>
----------	---

Description

Generic method for obtaining the marginal distribution of a distribution object x over components indices.

Usage

```
marginal(x, indices)
```

Arguments

x	The distribution object.
indices	The indices of the marginal distribution to obtain.

`marginal.empirical_dist`

Method for obtaining the marginal distribution of empirical_dist object x.

Description

Method for obtaining the marginal distribution of empirical_dist object x.

Usage

```
## S3 method for class 'empirical_dist'  
marginal(x, indices)
```

Arguments

x	The empirical distribution object.
indices	The indices of the marginal distribution to obtain.

`marginal.mvn`

Generic method for obtaining the marginal distribution of an mvn object x over components indices.

Description

Generic method for obtaining the marginal distribution of an mvn object x over components indices.

Usage

```
## S3 method for class 'mvn'  
marginal(x, indices)
```

Arguments

x	The mvn object.
indices	The indices of the marginal distribution to obtain.

mean.edist	<i>Method for obtaining the mean of an edist object.</i>
------------	--

Description

Method for obtaining the mean of an edist object.

Usage

```
## S3 method for class 'edist'  
mean(x, n = 10000, ...)
```

Arguments

x	The edist object to retrieve the mean from
n	The number of samples to take (default: 1000)
...	Additional arguments to pass (not used)

Value

The mean of the edist object

mean.empirical_dist	<i>Method for obtaining the mean of empirical_dist object x.</i>
---------------------	--

Description

Method for obtaining the mean of empirical_dist object x.

Usage

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

Arguments

x	The distribution object.
...	Additional arguments to pass (not used).

mean.exponential	<i>Method to obtain the mean of an exponential object.</i>
------------------	--

Description

Method to obtain the mean of an exponential object.

Usage

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

Arguments

x	The exponential object to obtain the mean of
...	Additional arguments (not used)

mean.mvn	<i>Retrieve the mean of a mvn object.</i>
----------	---

Description

Retrieve the mean of a mvn object.

Usage

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

Arguments

x	The mvn object to retrieve the mean from
...	Additional arguments to pass (not used)

Value

The mean of the mvn object

mean.normal	<i>Retrieve the mean of a normal object.</i>
-------------	--

Description

Retrieve the mean of a normal object.

Usage

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

Arguments

x	The normal object to retrieve the mean from
...	Additional arguments to pass (not used)

Value

The mean of the normal object

mean.univariate_dist	<i>Method for obtaining the mean of univariate_dist object x.</i>
----------------------	---

Description

Method for obtaining the mean of univariate_dist object x.

Usage

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

Arguments

x	The distribution object.
...	Additional arguments to pass into expectation.

mvn	<i>Construct a multivariate or univariate normal distribution object.</i>
-----	---

Description

This function constructs an object representing a normal distribution. If the length of the mean vector `mu` is 1, it creates a univariate normal distribution. Otherwise, it creates a multivariate normal distribution.

Usage

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

Arguments

mu	A numeric vector specifying the means of the distribution. If <code>mu</code> has length 1, a univariate normal distribution is created. If <code>mu</code> has length > 1 , a multivariate normal distribution is created.
sigma	A numeric matrix specifying the variance-covariance matrix of the distribution. It must be a square matrix with the same number of rows and columns as the length of <code>mu</code> . Default is the identity matrix of size equal to the length of <code>mu</code> .

Value

If `mu` has length 1, it returns a normal object. If `mu` has length > 1 , it returns an mvn object. Both types of objects contain `mu` and `sigma` as their properties.

nobs.empirical_dist	<i>Method for obtaining the number of observations used to construct a empirical_dist object.</i>
---------------------	---

Description

Method for obtaining the number of observations used to construct a `empirical_dist` object.

Usage

```
## S3 method for class 'empirical_dist'
nobs(object, ...)
```

Arguments

object	The empirical distribution object.
...	Additional arguments to pass (not used).

normal	<i>Construct univariate normal distribution object.</i>
--------	---

Description

Construct univariate normal distribution object.

Usage

```
normal(mu = 0, var = 1)
```

Arguments

mu	mean
var	variance

nparams	<i>Generic method for obtaining the number of parameters of distribution-like object x.</i>
---------	---

Description

Generic method for obtaining the number of parameters of distribution-like object x.

Usage

```
nparams(x)
```

Arguments

x	the object to obtain the number of parameters for
---	---

```
nparams.empirical_dist
```

Method for obtaining the name of a `empirical_dist` object. Since the empirical distribution is parameter-free, this function returns 0.

Description

Method for obtaining the name of a `empirical_dist` object. Since the empirical distribution is parameter-free, this function returns 0.

Usage

```
## S3 method for class 'empirical_dist'
nparams(x)
```

Arguments

`x` The empirical distribution object.

<code>obs</code>	<i>Retrieve the observations used to construct a distribution-like object. This is useful for obtaining the data used to construct an empirical distribution, but it is also useful for, say, retrieving the sample that was used by a fitted object, like an maximum likelihood estimate.</i>
------------------	--

Description

Retrieve the observations used to construct a distribution-like object. This is useful for obtaining the data used to construct an empirical distribution, but it is also useful for, say, retrieving the sample that was used by a fitted object, like an maximum likelihood estimate.

Usage

```
obs(x)
```

Arguments

`x` the object to retrieve the observations from

obs.empirical_dist	<i>Method for obtaining the observations used to construct a empirical_dist object.</i>
--------------------	---

Description

Method for obtaining the observations used to construct a empirical_dist object.

Usage

```
## S3 method for class 'empirical_dist'  
obs(x)
```

Arguments

x	The empirical distribution object.
---	------------------------------------

params	<i>Generic method for obtaining the parameters of an object.</i>
--------	--

Description

Generic method for obtaining the parameters of an object.

Usage

```
params(x)
```

Arguments

x	The object to obtain the parameters of.
---	---

params.edist	<i>Method for obtaining the parameters of an edist object.</i>
--------------	--

Description

Method for obtaining the parameters of an edist object.

Usage

```
## S3 method for class 'edist'  
params(x)
```

Arguments

x The object to obtain the parameters of

Value

A named vector of parameters

params.empirical_dist	<i>empirical_dist objects have no parameters, so this function returns NULL.</i>
-----------------------	--

Description

empirical_dist objects have no parameters, so this function returns NULL.

Usage

```
## S3 method for class 'empirical_dist'  
params(x)
```

Arguments

x The empirical distribution object.

params.exponential	<i>Method for obtaining the parameters of an exponential object.</i>
--------------------	--

Description

Method for obtaining the parameters of an exponential object.

Usage

```
## S3 method for class 'exponential'
params(x)
```

Arguments

x	The object to obtain the parameters of
---	--

Value

A named vector of parameters

params.mvn	<i>Method for obtaining the parameters of a mvn object.</i>
------------	---

Description

Method for obtaining the parameters of a mvn object.

Usage

```
## S3 method for class 'mvn'
params(x)
```

Arguments

x	The object to obtain the parameters of
---	--

Value

A named vector of parameters

params.normal	<i>Method for obtaining the parameters of a normal object.</i>
---------------	--

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
---	--

Value

A named vector of parameters

print.edist	<i>Method for printing an edist object.</i>
-------------	---

Description

Method for printing an edist object.

Usage

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

Arguments

x	The object to print
...	Additional arguments to pass (not used)

print.empirical_dist *Method for printing a dist object*

Description

Method for printing a dist object

Usage

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

Arguments

x	The object to print
...	Additional arguments to pass

print.exponential *Print method for exponential objects.*

Description

Print method for exponential objects.

Usage

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

Arguments

x	The exponential object to print.
...	Additional arguments (not used)

print.interval	<i>Print the interval.</i>
----------------	----------------------------

Description

Print the interval.

Usage

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

Arguments

x	An interval object.
...	Additional arguments.

print.mvn	<i>Method for printing an mvn object.</i>
-----------	---

Description

Method for printing an mvn object.

Usage

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

Arguments

x	The object to print
...	Additional arguments to pass to print

print.normal	<i>Method for printing a normal object.</i>
--------------	---

Description

Method for printing a normal object.

Usage

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

Arguments

x	The object to print
...	Additional arguments to pass (not used)

print.summary_dist	<i>Print method for summary_dist objects.</i>
--------------------	---

Description

Print method for summary_dist objects.

Usage

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

Arguments

x	The object to print
...	Additional arguments

rmap

Generic method for applying a map f to distribution object x.

Description

Generic method for applying a map f to distribution object x.

Usage

```
rmap(x, g, ...)
```

Arguments

x	The distribution object.
g	The function to apply.
...	Additional arguments to pass into g.

rmap.dist

Method for obtaining $g(x)$ where x is a dist object.

Description

We just sample from x and place the sample in empirical_dist and then apply rmap with g to it.

Usage

```
## S3 method for class 'dist'
rmap(x, g, n = 10000L, ...)
```

Arguments

x	The distribution object.
g	The function to apply to the distribution.
n	The number of samples to generate for the MC estimate of the conditional distribution $x P$. Defaults to 10000.
...	additional arguments to pass into g.

rmap.empirical_dist	<i>Method for obtaining the empirical distribution of a function of the observations of empirical_dist object x.</i>
---------------------	--

Description

Method for obtaining the empirical distribution of a function of the observations of empirical_dist object x.

Usage

```
## S3 method for class 'empirical_dist'
rmap(x, g, ...)
```

Arguments

x	The empirical distribution object.
g	The function to apply to each observation.
...	Additional arguments to pass into function g.

rmap.mvn	<i>Computes the distribution of $g(x)$ where x is an mvn object.</i>
----------	---

Description

By the invariance property, if x is an mvn object, then under the right conditions, asymptotically, $g(x)$ is an MVN distributed, $g(x) \sim \text{normal}(g(\text{mean}(x)), \text{sigma})$ where sigma is the variance-covariance of $g(x)$

Usage

```
## S3 method for class 'mvn'
rmap(x, g, n = 10000L, ...)
```

Arguments

x	The mvn object to apply g to
g	The function to apply to x
n	number of samples to take to estimate distribution of $g(x)$ if method is mc or empirical. Defaults to 10000.
...	additional arguments to pass into the g function.

sampler	<i>Generic method for sampling from distribution-like objects.</i>
---------	--

Description

It creates a sampler for the `x` object. It returns a function that accepts a parameter `n` denoting the number of samples to draw from the `x` object and also any additional parameters `...` are passed to the generated function.

Usage

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

Arguments

<code>x</code>	the <code>x</code> object to create a sampler for
<code>...</code>	additional arguments to pass

sampler.default	<i>Sampler for non-dist objects (degenerate distributions).</i>
-----------------	---

Description

Sampler for non-dist objects (degenerate distributions).

Usage

```
## Default S3 method:
sampler(x, ...)
```

Arguments

<code>x</code>	The object to sample from
<code>...</code>	Additional arguments to pass

Value

A function that takes `n` and returns `n` copies of `x`

sampler.edist	<i>Method for obtaining the sampler of an edist object.</i>
---------------	---

Description

Method for obtaining the sampler of an edist object.

Usage

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

Arguments

x	The edist object to obtain the sampler of.
...	Additional arguments to pass into each of the sampler function generators.

Value

A function that takes a number of samples n, ... which is passed into the expression x\$e and returns the result of applying the expression x\$e to the sampled values.

sampler.empirical_dist	<i>Method for obtaining the sampler for a empirical_dist object.</i>
------------------------	--

Description

Method for obtaining the sampler for a empirical_dist object.

Usage

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

Arguments

x	The object to obtain the sampler of.
...	Additional arguments to pass (not used).

sampler.exponential	<i>Method to sample from an exponential object.</i>
---------------------	---

Description

Method to sample from an exponential object.

Usage

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

Arguments

x	The exponential object to sample from.
...	Additional arguments to pass (not used)

Value

A function that allows sampling from the exponential distribution. Accepts an argument n denoting sample size and rate denoting the failure rate. Defaults to the failure rate of object x.

sampler.mvn	<i>Function generator for sampling from a mvn (multivariate normal) object.</i>
-------------	---

Description

Function generator for sampling from a mvn (multivariate normal) object.

Usage

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

Arguments

x	The mvn object to sample from
...	Additional arguments to pass to the generated function that will be fixed during all calls.

Value

A function that samples from the mvn distribution. It accepts as input: - n: number of samples to generate. Defaults to 1. - mu: a vector denoting the population mean. Defaults to the mean of x (an mvn object) - sigma: a matrix denoting the covariance of observations. Defaults to the variance-covariance of x. - p: probability region to sample from. Defaults to 1, which corresponds to the entire distribution. sample_mvn_region method. It's used when p is less than 1. - ...: any additional parameters to pass to rmvnorm or sample_mvn_region which can be different during each call.

sampler.normal	<i>Method for sampling from a normal object.</i>
----------------	--

Description

Method for sampling from a normal object.

Usage

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

Arguments

x	The normal object to sample from
...	Additional arguments to pass (not used)

Value

A function that samples from the normal distribution. As input, it accepts a sample size n, a numeric mu, and a variance numeric var. By default, mu and var are the mean and variance of object x.

sample_mvn_region	<i>Function for obtaining sample points for an mvn object that is within the p-probability region. That is, it samples from the smallest region of the distribution that contains p probability mass. This is done by first sampling from the entire distribution, then rejecting samples that are not in the probability region (using the statistical distance mahalanobis from mu).</i>
-------------------	--

Description

Function for obtaining sample points for an mvn object that is within the p-probability region. That is, it samples from the smallest region of the distribution that contains p probability mass. This is done by first sampling from the entire distribution, then rejecting samples that are not in the probability region (using the statistical distance mahalanobis from mu).

Usage

```
sample_mvn_region(n, mu, sigma, p = 0.95, ...)
```

Arguments

n	the sample size
mu	mean vector
sigma	variance-covariance matrix
p	the probability region
...	additional arguments to pass into mahalanobis

simplify	<i>Generic method for simplifying distributions.</i>
----------	--

Description

Generic method for simplifying distributions.

Usage

```
simplify(x, ...)
```

Arguments

x	The distribution to simplify
...	Additional arguments to pass

Value

The simplified distribution

simplify.dist	<i>Default Method for simplifming a dist object. Just returns the object.</i>
---------------	---

Description

Default Method for simplifming a dist object. Just returns the object.

Usage

```
## S3 method for class 'dist'
simplify(x, ...)
```

Arguments

x	The dist object to simplify
...	Additional arguments to pass (not used)

Value

The dist object

simplify.edist	<i>Method for simplifying an edist object.</i>
----------------	--

Description

Attempts to reduce expression distributions to closed-form distributions when mathematical identities apply. For example:

- normal + normal = normal (sum of independent normals)
- normal - normal = normal (difference of independent normals)

Usage

```
## S3 method for class 'edist'
simplify(x, ...)
```

Arguments

x	The edist object to simplify
...	Additional arguments to pass (not used)

Value

The simplified distribution, or unchanged edist if no rule applies

summary.dist	<i>Method for obtaining a summary of a dist object.</i>
--------------	---

Description

Method for obtaining a summary of a dist object.

Usage

```
## S3 method for class 'dist'
summary(object, ..., name = NULL, nobs = NULL)
```

Arguments

object	The object to obtain the summary of
...	Additional arguments to pass
name	The name of the distribution, defaults to the class of the object.
nobs	The number of observations to report for the summary, if applicable.

Value

A summary_dist object

summary_dist	<i>Method for constructing a summary_dist object.</i>
--------------	---

Description

Method for constructing a summary_dist object.

Usage

```
summary_dist(name, mean, vcov, nobs = NULL)
```

Arguments

name	The name of the distribution
mean	The mean of the distribution
vcov	The variance of the distribution
nobs	The number of observations used to construct the distribution, if applicable.

Value

A summary_dist object

sup	<i>Generic method for retrieving the support of a (dist) object x.</i>
-----	--

Description

The returned value should have the following operations:

- min: a vector, the minimum value of the support for each component.
- max: a vector, the maximum value of the support for each component.
- call: a predicate function, which returns TRUE if the value is in the support, and FALSE otherwise.
- sample: a function, which returns a sample from the support. Note that the returned value is not guaranteed to be in the support of x. You may need to call call to check.

Usage

```
sup(x)
```

Arguments

x	The object to obtain the support of.
---	--------------------------------------

Value

A support object for x.

sup.empirical_dist	<i>Method for obtaining the support of empirical_dist object x.</i>
--------------------	---

Description

Method for obtaining the support of empirical_dist object x.

Usage

```
## S3 method for class 'empirical_dist'
sup(x)
```

Arguments

x	The empirical distribution object.
---	------------------------------------

Value

A finite_set object containing the support of x.

sup.exponential	<i>Support for exponential distribution, the positive real numbers, (0, Inf).</i>
-----------------	---

Description

Support for exponential distribution, the positive real numbers, (0, Inf).

Usage

```
## S3 method for class 'exponential'
sup(x)
```

Arguments

x	The object to obtain the support of
---	-------------------------------------

Value

An interval object representing the support of the exponential

sup.mvn	<i>Method for obtaining the support of a mvn object, where the support is defined as values that have non-zero probability density.</i>
---------	---

Description

Method for obtaining the support of a mvn object, where the support is defined as values that have non-zero probability density.

Usage

```
## S3 method for class 'mvn'
sup(x, ...)
```

Arguments

x	The mvn object to obtain the support of
...	Additional arguments to pass (not used)

Value

A support-type object (see support.R), in this case an interval object for each component.

sup.normal	<i>Method for obtaining the support of a normal object, where the support is defined as values that have non-zero probability density.</i>
------------	--

Description

Method for obtaining the support of a normal object, where the support is defined as values that have non-zero probability density.

Usage

```
## S3 method for class 'normal'
sup(x)
```

Arguments

x The normal object to obtain the support of

Value

A support-type object (see support.R), in this case an interval object for each component.

supremum	<i>Get the supremum of the support.</i>
----------	---

Description

Get the supremum of the support.

Usage

```
supremum(object)
```

Arguments

object A support object.

supremum.finite_set	<i>Return the supremum of the finite set.</i>
---------------------	---

Description

Return the supremum of the finite set.

Usage

```
## S3 method for class 'finite_set'
supremum(object)
```

Arguments

object	A finite set.
--------	---------------

supremum.interval	<i>Return the (vector of) supremum of the interval.</i>
-------------------	---

Description

Return the (vector of) supremum of the interval.

Usage

```
## S3 method for class 'interval'
supremum(object)
```

Arguments

object	An interval object.
--------	---------------------

surv	<i>Generic method for obtaining the survival function of an object.</i>
------	---

Description

Generic method for obtaining the survival function of an object.

Usage

```
surv(x, ...)
```

Arguments

x	The object to obtain the survival function of.
...	Additional arguments to pass.

surv.exponential	<i>Method to obtain the cdf of an exponential object.</i>
------------------	---

Description

Method to obtain the cdf of an exponential object.

Usage

```
## S3 method for class 'exponential'
surv(x, ...)
```

Arguments

x	The object to obtain the pdf of
...	Additional arguments (not used)

Value

A function that computes the cdf of the exponential. Accepts as input a vector t at which to compute the cdf, an input rate denoting the failure rate of the exponential distribution, and a logical log indicating whether to compute the log of the cdf. By default, rate is the failure rate of object x.

vcov.default	<i>Variance-covariance for non-dist objects (degenerate distributions).</i>
--------------	---

Description

Variance-covariance for non-dist objects (degenerate distributions).

Usage

```
## Default S3 method:
vcov(object, ...)
```

Arguments

object	The object (returns 0 for constants)
...	Additional arguments to pass (not used)

Value

0 (degenerate distributions have no variance)

vcov.edist	<i>Method for obtaining the variance-covariance matrix (or scalar)</i>
------------	--

Description

Method for obtaining the variance-covariance matrix (or scalar)

Usage

```
## S3 method for class 'edist'
vcov(object, n = 10000, ...)
```

Arguments

object	The edist object to retrieve the variance-covariance matrix from
n	The number of samples to take (default: 1000)
...	Additional arguments to pass (not used)

Value

The variance-covariance matrix of the edist object

vcov.empirical_dist	<i>Method for obtaining the variance of empirical_dist object x.</i>
---------------------	--

Description

Method for obtaining the variance of empirical_dist object x.

Usage

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

Arguments

object	The empirical distribution object.
...	Additional arguments to pass (not used).

vcov.exponential	<i>Retrieve the variance of a exponential object.</i>
------------------	---

Description

Retrieve the variance of a exponential object.

Usage

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

Arguments

object	The exponential object to retrieve the variance for
...	Additional arguments to pass (not used)

Value

The variance-covariance matrix of the normal object

vcov.mvn	<i>Retrieve the variance-covariance matrix of an mvn object.</i>
----------	--

Description

Retrieve the variance-covariance matrix of an mvn object.

Usage

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

Arguments

object	The mvn object to retrieve the variance-covariance matrix of
...	Additional arguments to pass (not used)

Value

The variance-covariance matrix of the mvn object

vcov.normal	<i>Retrieve the variance-covariance matrix (or scalar) of a normal object.</i>
-------------	--

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
...	Additional arguments to pass (not used)

Value

The variance-covariance matrix of the normal object

vcov.univariate_dist	<i>Method for obtaining the variance of univariate_dist object.</i>
----------------------	---

Description

Method for obtaining the variance of univariate_dist object.

Usage

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

Arguments

object	The distribution object.
...	Additional arguments to pass into expectation.

Index

`+.dist`, 4
`-.dist`, 5

`cdf`, 5
`cdf.empirical_dist`, 6
`cdf.exponential`, 6
`cdf.mvn`, 7
`cdf.normal`, 7
`conditional`, 8
`conditional.dist`, 8
`conditional.empirical_dist`, 9

`density.empirical_dist`, 9
`density.exponential`, 10
`density.mvn`, 10
`density.normal`, 11
`dim.empirical_dist`, 11
`dim.exponential`, 12
`dim.finite_set`, 12
`dim.interval`, 13
`dim.mvn`, 13
`dim.normal`, 14

`edist`, 14
`empirical_dist`, 15
`expectation`, 15
`expectation.dist`, 16
`expectation.empirical_dist`, 16
`expectation.univariate_dist`, 17
`expectation_data`, 18
`exponential`, 19

`finite_set`, 19

`has`, 20
`has.finite_set`, 21
`has.interval`, 21
`hazard`, 22
`hazard.exponential`, 22

`infimum`, 23
`infimum.finite_set`, 23
`infimum.interval`, 23
`interval`, 24
`inv_cdf`, 25
`inv_cdf.exponential`, 26
`inv_cdf.normal`, 26
`is_dist`, 27
`is_edist`, 27
`is_empirical_dist`, 28
`is_exponential`, 28
`is_mvn`, 28
`is_normal`, 29

`marginal`, 29
`marginal.empirical_dist`, 30
`marginal.mvn`, 30
`mean.edist`, 31
`mean.empirical_dist`, 31
`mean.exponential`, 32
`mean.mvn`, 32
`mean.normal`, 33
`mean.univariate_dist`, 33
`mvn`, 34

`nobs.empirical_dist`, 34
`normal`, 35
`nparams`, 35
`nparams.empirical_dist`, 36

`obs`, 36
`obs.empirical_dist`, 37

`params`, 37
`params.edist`, 38
`params.empirical_dist`, 38
`params.exponential`, 39
`params.mvn`, 39
`params.normal`, 40
`print.edist`, 40
`print.empirical_dist`, 41

`print.exponential`, 41
`print.interval`, 42
`print.mvn`, 42
`print.normal`, 43
`print.summary_dist`, 43

`rmap`, 44
`rmap.dist`, 44
`rmap.empirical_dist`, 45
`rmap.mvn`, 45

`sample_mvn_region`, 49
`sampler`, 46
`sampler.default`, 46
`sampler.edist`, 47
`sampler.empirical_dist`, 47
`sampler.exponential`, 48
`sampler.mvn`, 48
`sampler.normal`, 49
`simplify`, 50
`simplify.dist`, 50
`simplify.edist`, 51
`summary.dist`, 51
`summary_dist`, 52
`sup`, 53
`sup.empirical_dist`, 53
`sup.exponential`, 54
`sup.mvn`, 54
`sup.normal`, 55
`supremum`, 55
`supremum.finite_set`, 56
`supremum.interval`, 56
`surv`, 56
`surv.exponential`, 57

`vcov.default`, 57
`vcov.edist`, 58
`vcov.empirical_dist`, 58
`vcov.exponential`, 59
`vcov.mvn`, 59
`vcov.normal`, 60
`vcov.univariate_dist`, 60