Package: BayesianInference (via r-universe)

May 13, 2026

Type Package

Title Bayesian Inference

Version 0.0.1

Description Beta version of 'Bayesian Inference' (BI) using 'python' and BI. It aims to unify the modeling experience by providing an intuitive model-building syntax together with the flexibility of low-level abstraction coding. It also includes pre-built functions for high-level abstraction and supports hardware-accelerated computation for improved scalability, including parallelization, vectorization, and execution on CPU, GPU, or TPU.

Encoding UTF-8

Imports reticulate

SystemRequirements Python (>= 3.6), 'BayesInference' python library.

Depends R (>= 3.5.0)

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

RoxygenNote 7.3.2

License GPL (>= 3)

NeedsCompilation no

Author Sebastian Sosa [aut, cre]

Maintainer Sebastian Sosa <bi@s-sosa.com>

Config/pak/sysreqs python3

Repository https://cran.r-universe.dev

Date/Publication 2026-01-13 18:10:02 UTC

RemoteUrl https://github.com/cran/BayesianInference

RemoteRef HEAD

RemoteSha 6e3e715b9d3fb077ccefb81ddedf6f00ba70131e

Contents

bi.dist.asymmetric_laplace . . . 3
bi.dist.asymmetric_laplace_quantile . . . 5
bi.dist.bernoulli . . . 7
bi.dist.beta . . . 9
bi.dist.beta_binomial . . . 11
bi.dist.beta_proportion . . . 12
bi.dist.binomial . . . 14
bi.dist.car . . . 16
bi.dist.categorical . . . 18
bi.dist.cauchy . . . 19
bi.dist.chi2 . . . 21
bi.dist.delta . . . 23
bi.dist.dirichlet . . . 24
bi.dist.dirichlet_multinomial . . . 26
bi.dist.discrete_uniform . . . 27
bi.dist.euler_maruyama . . . 29
bi.dist.exponential . . . 31
bi.dist.gamma . . . 32
bi.dist.gamma_poisson . . . 34
bi.dist.gaussian_copula . . . 36
bi.dist.gaussian_copula_beta . . . 37
bi.dist.gaussian_random_walk . . . 39
bi.dist.gaussian_state_space . . . 41
bi.dist.geometric . . . 43
bi.dist.gompertz . . . 44
bi.dist.gumbel . . . 46
bi.dist.half_cauchy . . . 47
bi.dist.half_normal . . . 49
bi.dist.inverse_gamma . . . 50
bi.dist.kumaraswamy . . . 52
bi.dist.laplace . . . 54
bi.dist.left_truncated_distribution . . . 55
bi.dist.levy . . . 57
bi.dist.lkj . . . 59
bi.dist.lkj_cholesky . . . 61
bi.dist.log_normal . . . 63
bi.dist.log_uniform . . . 64
bi.dist.logistic . . . 66
bi.dist.low_rank_multivariate_normal . . . 68
bi.dist.lower_truncated_power_law . . . 70
bi.dist.matrix_normal . . . 71
bi.dist.mixture . . . 73
bi.dist.mixture_general . . . 75
bi.dist.mixture_same_family . . . 77
bi.dist.multinomial . . . 79
bi.dist.multinomial_logits . . . 80

bi.dist.multinomial_probs . . . 82
bi.dist.multivariate_normal . . . 84
bi.dist.multivariate_student_t . . . 86
bi.dist.negative_binomial_logits . . . 88
bi.dist.negative_binomial_probs . . . 89
bi.dist.negative_binomial2 . . . 91
bi.dist.normal . . . 93
bi.dist.ordered_logistic . . . 94
bi.dist.pareto . . . 96
bi.dist.poisson . . . 98
bi.dist.projected_normal . . . 100
bi.dist.relaxed_bernoulli . . . 102
bi.dist.relaxed_bernoulli_logits . . . 103
bi.dist.right_truncated_distribution . . . 105
bi.dist.soft_laplace . . . 107
bi.dist.student_t . . . 108
bi.dist.truncated_cauchy . . . 110
bi.dist.truncated_distribution . . . 112
bi.dist.truncated_normal . . . 114
bi.dist.truncated_polya_gamma . . . 116
bi.dist.two_sided_truncated_distribution . . . 117
bi.dist.uniform . . . 119
bi.dist.weibull . . . 121
bi.dist.wishart . . . 122
bi.dist.wishart_cholesky . . . 124
bi.dist.zero_inflated_distribution . . . 126
bi.dist.zero_inflated_negative_binomial . . . 128
bi.dist.zero_inflated_poisson . . . 130
bi.dist.zero_sum_normal . . . 132
bi.doc . . . 134
importBI . . . 134
setup_env . . . 135

Index **136**

`bi.dist.asymmetric_laplace`

Asymmetric Laplace distribution

Description

Samples from an Asymmetric Laplace distribution. The Asymmetric Laplace distribution is a generalization of the Laplace distribution, where the two sides of the distribution are scaled differently. It is defined by a location parameter (loc), a scale parameter (scale), and an asymmetry parameter (asymmetry).

$$f(x, \kappa) = \frac{1}{\kappa + \kappa^{-1}} \exp(-x\kappa), \quad x \geq 0$$

$$= \frac{1}{\kappa + \kappa^{-1}} \exp(x/\kappa), \quad x < 0$$

$$\text{for } -\infty < x < \infty, \kappa > 0.$$

laplace_asymmetric takes 'kappa' as a shape parameter for κ. For $\kappa = 1$, it is identical to a Laplace distribution.

Usage

```
bi.dist.asymmetric_laplace(
  loc = 0,
  scale = 1,
  asymmetry = 1,
  validate_args = py_none(),
  name = "x",
  obs = py_none(),
  mask = py_none(),
  sample = FALSE,
  seed = py_none(),
  shape = c(),
  event = 0,
  create_obj = FALSE,
  to_jax = TRUE
)
```

Arguments

loc	A numeric vector or single numeric value representing the location parameter of the distribution. This corresponds to μ.
scale	A numeric vector or single numeric value representing the scale parameter of the distribution. This corresponds to σ.
asymmetry	A numeric vector or single numeric value representing the asymmetry parameter of the distribution. This corresponds to κ.
validate_args	Logical: Whether to validate parameter values. Defaults to 'reticulate::py_none()'.
name	A character string representing the name of the random variable within a model. This is used to uniquely identify the variable. Defaults to 'x'.
obs	A numeric vector or array of observed values. If provided, the random variable is conditioned on these values. If 'NULL', the variable is treated as a latent (unobserved) variable. Defaults to 'NULL'.
mask	A logical vector indicating which observations to mask.
sample	A logical value that controls the function's behavior. If 'TRUE', the function will directly draw samples from the distribution. If 'FALSE', it will create a random variable within a model. Defaults to 'FALSE'.
seed	An integer used to set the random seed for reproducibility when 'sample = TRUE'. This argument has no effect when 'sample = FALSE', as randomness is handled by the model's inference engine. Defaults to 0.

shape	A numeric vector specifying the shape of the output. This is used to set the batch shape when sample=FALSE (model building) or as 'sample_shape' to draw a raw JAX array when sample=TRUE (direct sampling).
event	Integer specifying the number of batch dimensions to reinterpret as event dimensions (used in model building).
create_obj	Logical; If 'TRUE', returns the raw BI distribution object instead of creating a sample site.
to_jax	Boolean. Indicates whether to return a JAX array or not.

Value

When sample=FALSE: A BI AsymmetricLaplace distribution object (for model building). When sample=TRUE: A JAX array of samples drawn from the AsymmetricLaplace distribution (for direct sampling). When create_obj=TRUE: The raw BI distribution object (for advanced use cases).

See Also

This is a wrapper of https://num.pyro.ai/en/stable/distributions.html#asymmetriclaplace

Examples

```
library(BayesianInference)
m <- importBI(platform = "cpu")
bi.dist.asymmetric_laplace(sample = TRUE)
```

---

bi.dist.asymmetric_laplace_quantile

Asymmetric Laplace Quantile Distribution

Description

Samples from an Asymmetric Laplace Quantile distribution. This distribution is an alternative parameterization of the Asymmetric Laplace distribution, commonly used in Bayesian quantile regression. It utilizes a 'quantile' parameter to define the balance between the left- and right-hand sides of the distribution, representing the proportion of probability density that falls to the left-hand side.

Usage

```
bi.dist.asymmetric_laplace_quantile(
  loc = 0,
  scale = 1,
  quantile = 0.5,
  validate_args = py_none(),
  name = "x",
  obs = py_none(),
```

```
  mask = py_none(),
  sample = FALSE,
  seed = py_none(),
  shape = c(),
  event = 0,
  create_obj = FALSE,
  to_jax = TRUE
)
```

Arguments

loc	The location parameter of the distribution.
scale	The scale parameter of the distribution.
quantile	The quantile parameter, representing the proportion of probability density to the left of the median. Must be between 0 and 1.
validate_args	Logical: Whether to validate parameter values. Defaults to 'reticulate::py_none()'.
name	A character string representing the name of the random variable within a model. This is used to uniquely identify the variable. Defaults to 'x'.
obs	A numeric vector or array of observed values. If provided, the random variable is conditioned on these values. If 'NULL', the variable is treated as a latent (unobserved) variable. Defaults to 'NULL'.
mask	An optional boolean array to mask observations.
sample	A logical value that controls the function's behavior. If 'TRUE', the function will directly draw samples from the distribution. If 'FALSE', it will create a random variable within a model. Defaults to 'FALSE'.
seed	An integer used to set the random seed for reproducibility when 'sample = TRUE'. This argument has no effect when 'sample = FALSE', as randomness is handled by the model's inference engine. Defaults to 0.
shape	A numeric vector. When 'sample=False' (model building), this is used with '.expand(shape)' to set the distribution's batch shape. When 'sample=True' (direct sampling), this is used as 'sample_shape' to draw a raw JAX array of the given shape.
event	The number of batch dimensions to reinterpret as event dimensions (used in model building).
create_obj	If 'TRUE', returns the raw NumPyro distribution object instead of creating a sample site. This is essential for building complex distributions like 'MixtureSameFamily'.
to_jax	Boolean. Indicates whether to return a JAX array or not.

Value

- When `sample=FALSE`, a BI Asymmetric Laplace Quantile distribution object (for model building).

- When `sample=TRUE`, a JAX array of samples drawn from the Asymmetric Laplace Quantile distribution (for direct sampling).

- When `create_obj=TRUE`, the raw BI distribution object (for advanced use cases).

See Also

This is a wrapper of https://num.pyro.ai/en/stable/distributions.html#asymmetriclaplacequantile

Examples

```
library(BayesianInference)
m=importBI(platform='cpu')
bi.dist.asymmetric_laplace_quantile(sample = TRUE)
```

`bi.dist.bernoulli` *Sample from a Bernoulli distribution.*

Description

The Bernoulli distribution models a single binary event (success or failure), parameterized by the log-odds ratio of success. The probability of success is given by the sigmoid function applied to the logit.

$$p = \sigma(\eta) = \frac{1}{1 + e^{-\eta}}$$

where

$$\eta$$

is the log-odds (the *logit*).

Usage

```
bi.dist.bernoulli(
  probs = py_none(),
  logits = py_none(),
  validate_args = py_none(),
  name = "x",
  obs = py_none(),
  mask = py_none(),
  sample = FALSE,
  seed = py_none(),
  shape = c(),
  event = 0,
  create_obj = FALSE,
  to_jax = TRUE
)
```

Arguments

probs	A numeric vector, matrix, or array representing the probability of success for each Bernoulli trial. Must be between 0 and 1.
logits	A numeric vector, matrix, or array representing the log-odds of success for each Bernoulli trial.
validate_args	Logical: Whether to validate parameter values. Defaults to 'reticulate::py_none()'.
name	A character string representing the name of the random variable within a model. This is used to uniquely identify the variable. Defaults to 'x'.
obs	A numeric vector or array of observed values. If provided, the random variable is conditioned on these values. If 'NULL', the variable is treated as a latent (unobserved) variable. Defaults to 'NULL'.
mask	A logical vector, matrix, or array (optional) to mask observations.
sample	A logical value that controls the function's behavior. If 'TRUE', the function will directly draw samples from the distribution. If 'FALSE', it will create a random variable within a model. Defaults to 'FALSE'.
seed	An integer used to set the random seed for reproducibility when 'sample = TRUE'. This argument has no effect when 'sample = FALSE', as randomness is handled by the model's inference engine. Defaults to 0.
shape	A numeric vector specifying the shape of the output. Used with '.expand(shape)' when 'sample=False' (model building) to set the distribution's batch shape. When 'sample=True' (direct sampling), this is used as 'sample_shape' to draw a raw JAX array of the given shape.
event	An integer indicating the number of batch dimensions to reinterpret as event dimensions (used in model building).
create_obj	A logical value (optional). If 'TRUE', returns the raw BI distribution object instead of creating a sample site.
to_jax	Boolean. Indicates whether to return a JAX array or not.

Value

- When `sample=FALSE`, a BI Bernoulli distribution object (for model building).

- When `sample=TRUE`, a JAX array of samples drawn from the Bernoulli distribution (for direct sampling).

- When `create_obj=TRUE`, the raw BI distribution object (for advanced use cases).

Examples

```
library(BayesianInference)
m=importBI(platform='cpu')
bi.dist.bernoulli(probs = 0.5, sample = TRUE)
bi.dist.bernoulli(probs = 0.5, sample = TRUE, seed = 5)
bi.dist.bernoulli(logits = 1, sample = TRUE, seed = 5)
```

`bi.dist.beta` *Beta Distribution*

Description

Samples from a Beta distribution, defined on the interval [0, 1]. The Beta distribution is a versatile distribution often used to model probabilities or proportions. It is parameterized by two positive shape parameters, usually denoted

$$\alpha$$

and

$$\beta > 0$$

, control the shape of the density (how much mass is pushed toward 0, 1, or intermediate).

$$X \sim Beta(\alpha, \beta), f(x) = \frac{x^{\alpha-1}(1-x)^{\beta-1}}{B(\alpha, \beta)}, B(\alpha, \beta) = \frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha+\beta)}, F(x) = I_x(\alpha+\beta)$$

where

$$B(\alpha, \beta)$$

is the Beta function:

$$B(\alpha, \beta) = \int_0^1 x^{\alpha-1}(1-x)^{\beta-1}, dx = \frac{\Gamma(\alpha), \Gamma(\beta)}{\Gamma(\alpha+\beta)}.$$

where

$$\alpha$$

and

$$\beta$$

are the concentration parameters, and

$$B(x, y)$$

is the Beta function.

Usage

```
bi.dist.beta(
  concentration1,
  concentration0,
  validate_args = py_none(),
  name = "x",
  obs = py_none(),
  mask = py_none(),
  sample = FALSE,
  seed = py_none(),
  shape = c(),
  event = 0,
  create_obj = FALSE,
  to_jax = TRUE
)
```

Arguments

concentration1	A numeric vector or array representing the first concentration parameter (shape parameter). Must be positive.
concentration0	A numeric vector or array representing the second concentration parameter (shape parameter). Must be positive.
validate_args	Logical: Whether to validate parameter values. Defaults to 'reticulate::py_none()'.
name	A character string representing the name of the random variable within a model. This is used to uniquely identify the variable. Defaults to 'x'.
obs	A numeric vector or array of observed values. If provided, the random variable is conditioned on these values. If 'NULL', the variable is treated as a latent (unobserved) variable. Defaults to 'NULL'.
mask	A logical vector or array. Optional boolean array to mask observations.
sample	A logical value that controls the function's behavior. If 'TRUE', the function will directly draw samples from the distribution. If 'FALSE', it will create a random variable within a model. Defaults to 'FALSE'.
seed	An integer used to set the random seed for reproducibility when 'sample = TRUE'. This argument has no effect when 'sample = FALSE', as randomness is handled by the model's inference engine. Defaults to 0.
shape	A numeric vector. When 'sample=False' (model building), this is used with '.expand(shape)' to set the distribution's batch shape. When 'sample=True' (direct sampling), this is used as 'sample_shape' to draw a raw JAX array of the given shape.
event	An integer representing the number of batch dimensions to reinterpret as event dimensions (used in model building).
create_obj	A logical value. If 'TRUE', returns the raw BI distribution object instead of creating a sample site. This is essential for building complex distributions like 'MixtureSameFamily'.
to_jax	Boolean. Indicates whether to return a JAX array or not.

Value

- When `sample=FALSE`, a BI Beta distribution object (for model building).

- When `sample=TRUE`, a JAX array of samples drawn from the Beta distribution (for direct sampling).

- When `create_obj=TRUE`, the raw BI distribution object (for advanced use cases).

See Also

This is a wrapper of https://num.pyro.ai/en/stable/distributions.html#beta

Examples

```
library(BayesianInference)
m=importBI(platform='cpu')
bi.dist.beta(concentration1 = 0, concentration0 = 1, sample = TRUE)
```

`bi.dist.beta_binomial` *BetaBinomial*

Description

Samples from a BetaBinomial distribution, a compound distribution where the probability of success in a binomial experiment is drawn from a Beta distribution. This models situations where the underlying probability of success is not fixed but varies according to a prior belief represented by the Beta distribution. It is often used to model over-dispersion relative to the binomial distribution.

Usage

```
bi.dist.beta_binomial(
  concentration1,
  concentration0,
  total_count = 1,
  validate_args = py_none(),
  name = "x",
  obs = py_none(),
  mask = py_none(),
  sample = FALSE,
  seed = py_none(),
  shape = c(),
  event = 0,
  create_obj = FALSE,
  to_jax = TRUE
)
```

Arguments

`concentration1`	A numeric vector, matrix, or array representing the first concentration parameter (alpha) of the Beta distribution.
`concentration0`	A numeric vector, matrix, or array representing the second concentration parameter (beta) of the Beta distribution.
`total_count`	A numeric vector, matrix, or array representing the number of Bernoulli trials in the Binomial part of the distribution.
`validate_args`	Logical: Whether to validate parameter values. Defaults to 'reticulate::py_none()'.
`name`	A character string representing the name of the random variable within a model. This is used to uniquely identify the variable. Defaults to 'x'.
`obs`	A numeric vector or array of observed values. If provided, the random variable is conditioned on these values. If 'NULL', the variable is treated as a latent (unobserved) variable. Defaults to 'NULL'.
`mask`	A logical vector. Optional boolean array to mask observations.
`sample`	A logical value that controls the function's behavior. If 'TRUE', the function will directly draw samples from the distribution. If 'FALSE', it will create a random variable within a model. Defaults to 'FALSE'.

seed	An integer used to set the random seed for reproducibility when ‘sample = TRUE‘. This argument has no effect when ‘sample = FALSE‘, as randomness is handled by the model’s inference engine. Defaults to 0.
shape	A numeric vector. When ‘sample=False‘ (model building), this is used with ‘.expand(shape)‘ to set the distribution’s batch shape. When ‘sample=True‘ (direct sampling), this is used as ‘sample_shape‘ to draw a raw JAX array of the given shape.
event	An integer representing the number of batch dimensions to reinterpret as event dimensions (used in model building).
create_obj	A logical value. If ‘TRUE‘, returns the raw BI distribution object instead of creating a sample site.
to_jax	Boolean. Indicates whether to return a JAX array or not.

Value

- When `sample=FALSE`, a BI Beta-Binomial distribution object (for model building).

- When `sample=TRUE`, a JAX array of samples drawn from the Beta-Binomial distribution (for direct sampling).

- When `create_obj=TRUE`, the raw BI distribution object (for advanced use cases).

See Also

This is a wrapper of https://num.pyro.ai/en/stable/distributions.html#betabinomial

Examples

```
library(BayesianInference)
m=importBI(platform='cpu')
bi.dist.beta_binomial(0,1,sample = TRUE)
```

`bi.dist.beta_proportion`

Samples from a Beta-Proportion distribution.

Description

The Beta Proportion distribution is a reparameterization of the conventional Beta distribution in terms of a the variate mean and a precision parameter. It’s useful for modeling rates and proportions. It’s essentially the same family as the standard Beta

$$(\alpha, \beta)$$

, but the mapping is:

$$\alpha = \mu, \kappa, \quad \beta = (1 - \mu), \kappa.$$

Usage

```
bi.dist.beta_proportion(
  mean,
  concentration,
  validate_args = py_none(),
  name = "x",
  obs = py_none(),
  mask = py_none(),
  sample = FALSE,
  seed = py_none(),
  shape = c(),
  event = 0,
  create_obj = FALSE,
  to_jax = TRUE
)
```

Arguments

mean	A numeric vector, matrix, or array representing the mean of the BetaProportion distribution, must be between 0 and 1.
concentration	A numeric vector, matrix, or array representing the concentration parameter of the BetaProportion distribution.
validate_args	Logical: Whether to validate parameter values. Defaults to ‘reticulate::py_none()‘.
name	A character string representing the name of the random variable within a model. This is used to uniquely identify the variable. Defaults to ’x’.
obs	A numeric vector or array of observed values. If provided, the random variable is conditioned on these values. If ‘NULL‘, the variable is treated as a latent (unobserved) variable. Defaults to ‘NULL‘.
mask	An optional boolean vector to mask observations.
sample	A logical value that controls the function’s behavior. If ‘TRUE‘, the function will directly draw samples from the distribution. If ‘FALSE‘, it will create a random variable within a model. Defaults to ‘FALSE‘.
seed	An integer used to set the random seed for reproducibility when ‘sample = TRUE‘. This argument has no effect when ‘sample = FALSE‘, as randomness is handled by the model’s inference engine. Defaults to 0.
shape	A numeric vector. When ‘sample=False‘ (model building), this is used with ‘.expand(shape)‘ to set the distribution’s batch shape. When ‘sample=True‘ (direct sampling), this is used as ‘sample_shape‘ to draw a raw JAX array of the given shape.
event	An integer representing the number of batch dimensions to reinterpret as event dimensions (used in model building).
create_obj	A logical value. If ‘TRUE‘, returns the raw BI distribution object instead of creating a sample site. This is essential for building complex distributions like ‘MixtureSameFamily‘.
to_jax	Boolean. Indicates whether to return a JAX array or not.

Value

- When sample=FALSE, a BI Beta-Proportion distribution object (for model building).

- When sample=TRUE, a JAX array of samples drawn from the Beta-Proportion distribution (for direct sampling).

- When create_obj=TRUE, the raw BI distribution object (for advanced use cases).

Examples

```
library(BayesianInference)
m=importBI(platform='cpu')
bi.dist.beta_proportion(0, 1, sample = TRUE)
```

`bi.dist.binomial` *Samples from a Binomial distribution.*

Description

The Binomial distribution models the number of successes in a sequence of independent Bernoulli trials. It represents the probability of obtaining exactly *k* successes in *n* trials, where each trial has a probability *p* of success.

$$P(X = k) = \binom{n}{k} p^k (1-p)^{n-k}$$

Usage

```
bi.dist.binomial(
  total_count = 1L,
  probs = py_none(),
  logits = py_none(),
  validate_args = py_none(),
  name = "x",
  obs = py_none(),
  mask = py_none(),
  sample = FALSE,
  seed = py_none(),
  shape = c(),
  event = 0,
  create_obj = FALSE,
  to_jax = TRUE
)
```

Arguments

total_count	(int): The number of trials *n*.
probs	(numeric vector, optional): The probability of success *p* for each trial. Must be between 0 and 1.
logits	(numeric vector, optional): The log-odds of success for each trial.
validate_args	Logical: Whether to validate parameter values. Defaults to 'reticulate::py_none()'.
name	A character string representing the name of the random variable within a model. This is used to uniquely identify the variable. Defaults to 'x'.
obs	A numeric vector or array of observed values. If provided, the random variable is conditioned on these values. If 'NULL', the variable is treated as a latent (unobserved) variable. Defaults to 'NULL'.
mask	(numeric vector of booleans, optional): Optional boolean array to mask observations.
sample	A logical value that controls the function's behavior. If 'TRUE', the function will directly draw samples from the distribution. If 'FALSE', it will create a random variable within a model. Defaults to 'FALSE'.
seed	An integer used to set the random seed for reproducibility when 'sample = TRUE'. This argument has no effect when 'sample = FALSE', as randomness is handled by the model's inference engine. Defaults to 0.
shape	(numeric vector): A multi-purpose argument for shaping. When 'sample=False' (model building), this is used with '.expand(shape)' to set the distribution's batch shape. When 'sample=True' (direct sampling), this is used as 'sample_shape' to draw a raw JAX array of the given shape.
event	(int): The number of batch dimensions to reinterpret as event dimensions (used in model building).
create_obj	(logical, optional): If 'TRUE', returns the raw BI distribution object instead of creating a sample site. This is essential for building complex distributions like 'MixtureSameFamily'.
to_jax	Boolean. Indicates whether to return a JAX array or not.

Value

- When `sample=FALSE`, a BI Binomial distribution object (for model building).

- When `sample=TRUE`, a JAX array of samples drawn from the Binomial distribution (for direct sampling).

- When `create_obj=TRUE`, the raw BI distribution object (for advanced use cases).

Examples

```
library(BayesianInference)
m=importBI(platform='cpu')
bi.dist.binomial(probs = c(0.5,0.5), sample = TRUE)
bi.dist.binomial(logits = 1, sample = TRUE)
```

bi.dist.car *Conditional Autoregressive (CAR) Distribution*

Description

The CAR distribution models a vector of variables where each variable is a linear combination of its neighbors in a graph. The CAR model captures spatial dependence in areal data by modeling each observation as conditionally dependent on its neighbors. It specifies a joint distribution of a vector of random variables

$$\mathbf{y} = (y_1, y_2, \ldots, y_N)$$

based on their conditional distributions, where each

$$y_i$$

is conditionally independent of all other variables given its neighbors. - Application: Widely used in disease mapping, environmental modeling, and spatial econometrics to account for spatial autocorrelation.

Usage

```
bi.dist.car(
  loc,
  correlation,
  conditional_precision,
  adj_matrix,
  is_sparse = FALSE,
  validate_args = py_none(),
  name = "x",
  obs = py_none(),
  mask = py_none(),
  sample = FALSE,
  seed = py_none(),
  shape = c(),
  event = 0,
  create_obj = FALSE,
  to_jax = TRUE
)
```

Arguments

loc	Numeric vector, matrix, or array representing the mean of the distribution.
correlation	Numeric vector, matrix, or array representing the correlation between variables.
conditional_precision	Numeric vector, matrix, or array representing the precision of the distribution.
adj_matrix	Numeric vector, matrix, or array representing the adjacency matrix defining the graph.

is_sparse	Logical indicating whether the adjacency matrix is sparse. Defaults to 'FALSE'.
validate_args	Logical indicating whether to validate arguments. Defaults to 'reticulate::py_none()'.
name	A character string representing the name of the random variable within a model. This is used to uniquely identify the variable. Defaults to 'x'.
obs	A numeric vector or array of observed values. If provided, the random variable is conditioned on these values. If 'NULL', the variable is treated as a latent (unobserved) variable. Defaults to 'NULL'.
mask	An optional boolean vector to mask observations.
sample	A logical value that controls the function's behavior. If 'TRUE', the function will directly draw samples from the distribution. If 'FALSE', it will create a random variable within a model. Defaults to 'FALSE'.
seed	An integer used to set the random seed for reproducibility when 'sample = TRUE'. This argument has no effect when 'sample = FALSE', as randomness is handled by the model's inference engine. Defaults to 0.
shape	A numeric vector. When 'sample=False' (model building), this is used with '.expand(shape)' to set the distribution's batch shape. When 'sample=True' (direct sampling), this is used as 'sample_shape' to draw a raw JAX array of the given shape.
event	An integer representing the number of batch dimensions to reinterpret as event dimensions (used in model building).
create_obj	A logical value. If 'TRUE', returns the raw BI distribution object instead of creating a sample site. This is essential for building complex distributions like 'MixtureSameFamily'.
to_jax	Boolean. Indicates whether to return a JAX array or not.

Value

- When `sample=FALSE`, a BI CAR distribution object (for model building).

- When `sample=TRUE`, a JAX array of samples drawn from the CAR distribution (for direct sampling).

- When `create_obj=TRUE`, the raw BI distribution object (for advanced use cases).

Examples

```
library(BayesianInference)
m=importBI(platform='cpu')
bi.dist.car(
  loc = c(1.,2.),
  correlation = 0.9,
  conditional_precision = 1.,
  adj_matrix = matrix(c(1,0,0,1), nrow = 2),
  sample = TRUE
 )
```

`bi.dist.categorical` *Sample from a Categorical distribution.*

Description

The Categorical distribution, also known as the multinomial distribution, describes the probability of different outcomes from a finite set of possibilities. It is commonly used to model discrete choices or classifications.

$$P(k) = \frac{e^{\log(p_k)}}{\sum_{j=1}^{K} e^{\log(p_j)}}$$

where

$$p_k$$

is the probability of outcome

$$k$$

, and the sum is over all possible outcomes.

Usage

```
bi.dist.categorical(
  probs = py_none(),
  logits = py_none(),
  validate_args = py_none(),
  name = "x",
  obs = py_none(),
  mask = py_none(),
  sample = FALSE,
  seed = py_none(),
  shape = c(),
  event = 0,
  create_obj = FALSE,
  to_jax = TRUE
)
```

Arguments

`probs`	A numeric vector of probabilities for each category. Must sum to 1.
`logits`	A numeric vector of Log-odds of each category.
`validate_args`	Logical: Whether to validate parameter values. Defaults to 'reticulate::py_none()'.
`name`	A character string representing the name of the random variable within a model. This is used to uniquely identify the variable. Defaults to 'x'.
`obs`	A numeric vector or array of observed values. If provided, the random variable is conditioned on these values. If 'NULL', the variable is treated as a latent (unobserved) variable. Defaults to 'NULL'.

mask	An optional boolean vector to mask observations.
sample	A logical value that controls the function's behavior. If 'TRUE', the function will directly draw samples from the distribution. If 'FALSE', it will create a random variable within a model. Defaults to 'FALSE'.
seed	An integer used to set the random seed for reproducibility when 'sample = TRUE'. This argument has no effect when 'sample = FALSE', as randomness is handled by the model's inference engine. Defaults to 0.
shape	A numeric vector specifying the shape. When sample=FALSE (model building), this is used with '.expand(shape)' to set the distribution's batch shape. When sample=TRUE (direct sampling), this is used as 'sample_shape' to draw a raw JAX array of the given shape.
event	The number of batch dimensions to reinterpret as event dimensions (used in model building).
create_obj	Logical; If 'TRUE', returns the raw BI distribution object instead of creating a sample site.
to_jax	Boolean. Indicates whether to return a JAX array or not.

Value

- When sample=FALSE, a BI Categorical distribution object (for model building).

- When sample=TRUE, a JAX array of samples drawn from the Categorical distribution (for direct sampling).

- When create_obj=TRUE, the raw BI distribution object (for advanced use cases).

Examples

```
library(BayesianInference)
m=importBI(platform='cpu')
bi.dist.categorical(probs = c(0.5,0.5), sample = TRUE, shape = c(3))
```

---

bi.dist.cauchy *Cauchy Distribution*

Description

Samples from a Cauchy distribution.

The Cauchy distribution, also known as the Lorentz distribution, is a continuous probability distribution that arises frequently in various fields, including physics and statistics. It is characterized by its heavy tails, which extend indefinitely. This means it has a higher probability of extreme values compared to the normal distribution.

Usage

```
bi.dist.cauchy(
  loc = 0,
  scale = 1,
  validate_args = py_none(),
  name = "x",
  obs = py_none(),
  mask = py_none(),
  sample = FALSE,
  seed = py_none(),
  shape = c(),
  event = 0,
  create_obj = FALSE,
  to_jax = TRUE
)
```

Arguments

loc	A numeric vector or scalar representing the location parameter. Defaults to 0.0.
scale	A numeric vector or scalar representing the scale parameter. Must be positive. Defaults to 1.0.
validate_args	Logical: Whether to validate parameter values. Defaults to ‘reticulate::py_none()‘.
name	A character string representing the name of the random variable within a model. This is used to uniquely identify the variable. Defaults to ’x’.
obs	A numeric vector or array of observed values. If provided, the random variable is conditioned on these values. If ‘NULL‘, the variable is treated as a latent (unobserved) variable. Defaults to ‘NULL‘.
mask	A logical vector, optional, to mask observations. Defaults to ‘reticulate::py_none()‘.
sample	A logical value that controls the function’s behavior. If ‘TRUE‘, the function will directly draw samples from the distribution. If ‘FALSE‘, it will create a random variable within a model. Defaults to ‘FALSE‘.
seed	An integer used to set the random seed for reproducibility when ‘sample = TRUE‘. This argument has no effect when ‘sample = FALSE‘, as randomness is handled by the model’s inference engine. Defaults to 0.
shape	A numeric vector specifying the shape of the distribution. When ‘sample=False‘ (model building), this is used with ‘.expand(shape)‘ to set the distribution’s batch shape. When ‘sample=True‘ (direct sampling), this is used as ‘sample_shape‘ to draw a raw JAX array of the given shape.
event	An integer representing the number of batch dimensions to reinterpret as event dimensions (used in model building). Defaults to ‘reticulate::py_none()‘.
create_obj	A logical value. If ‘TRUE‘, returns the raw BI distribution object instead of creating a sample site. Defaults to ‘FALSE‘.
to_jax	Boolean. Indicates whether to return a JAX array or not.

Value

- When sample=FALSE, a BI Cauchy distribution object (for model building).

- When sample=TRUE, a JAX array of samples drawn from the Cauchy distribution (for direct sampling).

- When create_obj=TRUE, the raw BI distribution object (for advanced use cases).

See Also

This is a wrapper of https://num.pyro.ai/en/stable/distributions.html#cauchy

Examples

```
library(BayesianInference)
m=importBI(platform='cpu')
bi.dist.cauchy(sample = TRUE)
```

---

bi.dist.chi2 *Samples from a Chi-squared distribution.*

Description

The Chi-squared distribution is a continuous probability distribution that arises frequently in hypothesis testing, particularly in ANOVA and chi-squared tests. It is defined by a single positive parameter, degrees of freedom (df), the number of independent standard normal variables squared and summed, which determines the shape of the distribution.

Usage

```
bi.dist.chi2(
  df,
  validate_args = py_none(),
  name = "x",
  obs = py_none(),
  mask = py_none(),
  sample = FALSE,
  seed = py_none(),
  shape = c(),
  event = 0,
  create_obj = FALSE,
  to_jax = TRUE
)
```

Arguments

df	A numeric vector representing the degrees of freedom. Must be positive.
validate_args	Logical: Whether to validate parameter values. Defaults to 'reticulate::py_none()'.
name	A character string representing the name of the random variable within a model. This is used to uniquely identify the variable. Defaults to 'x'.
obs	A numeric vector or array of observed values. If provided, the random variable is conditioned on these values. If 'NULL', the variable is treated as a latent (unobserved) variable. Defaults to 'NULL'.
mask	A logical vector, matrix, or array to mask observations.
sample	A logical value that controls the function's behavior. If 'TRUE', the function will directly draw samples from the distribution. If 'FALSE', it will create a random variable within a model. Defaults to 'FALSE'.
seed	An integer used to set the random seed for reproducibility when 'sample = TRUE'. This argument has no effect when 'sample = FALSE', as randomness is handled by the model's inference engine. Defaults to 0.
shape	A numeric vector used for shaping. When 'sample=FALSE' (model building), this is used with '.expand(shape)' to set the distribution's batch shape. When 'sample=TRUE' (direct sampling), this is used as 'sample_shape' to draw a raw JAX array of the given shape.
event	An integer representing the number of batch dimensions to reinterpret as event dimensions (used in model building).
create_obj	A logical value. If 'TRUE', returns the raw BI distribution object instead of creating a sample site. This is essential for building complex distributions like 'MixtureSameFamily'.
to_jax	Boolean. Indicates whether to return a JAX array or not.

Value

- When `sample=FALSE`, a BI Chi-squared distribution object (for model building).

- When `sample=TRUE`, a JAX array of samples drawn from the Chi-squared distribution (for direct sampling).

- When `create_obj=TRUE`, the raw BI distribution object (for advanced use cases).

See Also

This is a wrapper of https://num.pyro.ai/en/stable/distributions.html#chi2

Examples

```
library(BayesianInference)
m=importBI(platform='cpu')
bi.dist.chi2(c(0,2),sample = TRUE)
```

`bi.dist.delta` *The Delta distribution.*

Description

The Delta distribution, also known as a point mass distribution, assigns probability 1 to a single point and 0 elsewhere. It's useful for representing deterministic variables or as a building block for more complex distributions.

Usage

```
bi.dist.delta(
  v = 0,
  log_density = 0,
  event_dim = 0,
  validate_args = py_none(),
  name = "x",
  obs = py_none(),
  mask = py_none(),
  sample = FALSE,
  seed = py_none(),
  shape = c(),
  event = 0,
  create_obj = FALSE,
  to_jax = TRUE
)
```

Arguments

`v`	A numeric vector representing the location of the point mass.
`log_density`	The log probability density of the point mass. This is primarily for creating distributions that are non-normalized or for specific advanced use cases. For a standard delta distribution, this should be 0. Defaults to 0.0.
`event_dim`	event_dim (A numeric vector, optional): The number of rightmost dimensions of ‘v‘ to interpret as event dimensions. Defaults to 0.
`validate_args`	Logical: Whether to validate parameter values. Defaults to ‘reticulate::py_none()‘.
`name`	A character string representing the name of the random variable within a model. This is used to uniquely identify the variable. Defaults to 'x'.
`obs`	A numeric vector or array of observed values. If provided, the random variable is conditioned on these values. If ‘NULL‘, the variable is treated as a latent (unobserved) variable. Defaults to ‘NULL‘.
`mask`	A boolean vector to mask observations.
`sample`	A logical value that controls the function's behavior. If ‘TRUE‘, the function will directly draw samples from the distribution. If ‘FALSE‘, it will create a random variable within a model. Defaults to ‘FALSE‘.

seed	An integer used to set the random seed for reproducibility when 'sample = TRUE'. This argument has no effect when 'sample = FALSE', as randomness is handled by the model's inference engine. Defaults to 0.
shape	A numeric vector used for shaping. When 'sample=FALSE' (model building), this is used with '.expand(shape)' to set the distribution's batch shape. When 'sample=TRUE' (direct sampling), this is used as 'sample_shape' to draw a raw JAX array of the given shape.
event	The number of batch dimensions to reinterpret as event dimensions (used in model building).
create_obj	A logical value. If 'TRUE', returns the raw BI distribution object instead of creating a sample site.
to_jax	Boolean. Indicates whether to return a JAX array or not.

Value

- When `sample=FALSE`, a BI Delta distribution object (for model building).

- When `sample=TRUE`, a JAX array of samples drawn from the Delta distribution (for direct sampling).

- When `create_obj=TRUE`, the raw BI distribution object (for advanced use cases).

See Also

This is a wrapper of https://num.pyro.ai/en/stable/distributions.html#delta

Examples

```
library(BayesianInference)
m=importBI(platform='cpu')
bi.dist.delta(v = 5, sample = TRUE)
```

`bi.dist.dirichlet` *Samples from a Dirichlet distribution.*

Description

The Dirichlet distribution is a multivariate generalization of the Beta distribution. It is a probability distribution over a simplex, which is a set of vectors where each element is non-negative and sums to one. It is often used as a prior distribution for categorical distributions.

Usage

```
bi.dist.dirichlet(
  concentration,
  validate_args = py_none(),
  name = "x",
  obs = py_none(),
  mask = py_none(),
  sample = FALSE,
  seed = py_none(),
  shape = c(),
  event = 0,
  create_obj = FALSE,
  to_jax = TRUE
)
```

Arguments

concentration	A numeric vector or array representing the concentration parameter(s) of the Dirichlet distribution. Must be positive.
validate_args	Logical: Whether to validate parameter values. Defaults to 'reticulate::py_none()'.
name	A character string representing the name of the random variable within a model. This is used to uniquely identify the variable. Defaults to 'x'.
obs	A numeric vector or array of observed values. If provided, the random variable is conditioned on these values. If 'NULL', the variable is treated as a latent (unobserved) variable. Defaults to 'NULL'.
mask	A logical vector or array to mask observations.
sample	A logical value that controls the function's behavior. If 'TRUE', the function will directly draw samples from the distribution. If 'FALSE', it will create a random variable within a model. Defaults to 'FALSE'.
seed	An integer used to set the random seed for reproducibility when 'sample = TRUE'. This argument has no effect when 'sample = FALSE', as randomness is handled by the model's inference engine. Defaults to 0.
shape	A numeric vector specifying the shape of the distribution.
event	Integer representing the number of batch dimensions to reinterpret as event dimensions (used in model building).
create_obj	Logical; If TRUE, returns the raw BI distribution object instead of creating a sample site.
to_jax	Boolean. Indicates whether to return a JAX array or not.

Value

- When `sample=FALSE`, a BI Dirichlet distribution object (for model building).

- When `sample=TRUE`, a JAX array of samples drawn from the Dirichlet distribution (for direct sampling).

- When `create_obj=TRUE`, the raw BI distribution object (for advanced use cases).

See Also

This is a wrapper of https://num.pyro.ai/en/stable/distributions.html#dirichlet

Examples

```
library(BayesianInference)
m=importBI(platform='cpu')
bi.dist.dirichlet(concentration =  c(0.1,.9), sample = TRUE)
```

---

bi.dist.dirichlet_multinomial

Samples from a Dirichlet Multinomial distribution.

Description

Creates a Dirichlet-Multinomial compound distribution, which is a Multinomial distribution with a Dirichlet prior on its probabilities. It is often used in Bayesian statistics to model count data where the proportions of categories are uncertain.

Usage

```
bi.dist.dirichlet_multinomial(
  concentration,
  total_count = 1,
  validate_args = py_none(),
  name = "x",
  obs = py_none(),
  mask = py_none(),
  sample = FALSE,
  seed = py_none(),
  shape = c(),
  event = 0,
  create_obj = FALSE,
  to_jax = TRUE
)
```

Arguments

concentration	A numeric vector or array representing the concentration parameter (alpha) for the Dirichlet distribution.
total_count	(int, jnp.ndarray, optional): The total number of trials (n). This must be a non-negative integer. Defaults to 1.
validate_args	Logical: Whether to validate parameter values. Defaults to ‘reticulate::py_none()‘.
name	A character string representing the name of the random variable within a model. This is used to uniquely identify the variable. Defaults to ’x’.

obs	A numeric vector or array of observed values. If provided, the random variable is conditioned on these values. If 'NULL', the variable is treated as a latent (unobserved) variable. Defaults to 'NULL'.
mask	A logical vector or array to mask observations.
sample	A logical value that controls the function's behavior. If 'TRUE', the function will directly draw samples from the distribution. If 'FALSE', it will create a random variable within a model. Defaults to 'FALSE'.
seed	An integer used to set the random seed for reproducibility when 'sample = TRUE'. This argument has no effect when 'sample = FALSE', as randomness is handled by the model's inference engine. Defaults to 0.
shape	A numeric vector specifying the shape of the distribution. When 'sample=False' (model building), this is used with '.expand(shape)' to set the distribution's batch shape. When 'sample=True' (direct sampling), this is used as 'sample_shape' to draw a raw JAX array of the given shape.
event	The number of batch dimensions to reinterpret as event dimensions (used in model building).
create_obj	A logical value. If 'TRUE', returns the raw BI distribution object instead of creating a sample site.
to_jax	Boolean. Indicates whether to return a JAX array or not.

Value

- When `sample=FALSE`, a BI Dirichlet Multinomial distribution object (for model building).

- When `sample=TRUE`, a JAX array of samples drawn from the Dirichlet Multinomial distribution (for direct sampling).

- When `create_obj=TRUE`, the raw BI distribution object (for advanced use cases).

Examples

```
library(BayesianInference)
m=importBI(platform='cpu')
bi.dist.dirichlet_multinomial(concentration = c(0,1), sample = TRUE, shape = c(3))
```

`bi.dist.discrete_uniform`

Samples from a Discrete Uniform distribution.

Description

The Discrete Uniform distribution defines a uniform distribution over a range of integers. It is characterized by a lower bound ('low') and an upper bound ('high'), inclusive.

$$P(X = k) = \frac{1}{high - low + 1}, \quad k \in \{low, low + 1, ..., high\}$$

Otherwise (if

$$k$$

is outside the range), $ P(X = k) = 0 $.

Samples from a Discrete Uniform distribution.

Usage

```
bi.dist.discrete_uniform(
  low = 0,
  high = 1,
  validate_args = py_none(),
  name = "x",
  obs = py_none(),
  mask = py_none(),
  sample = FALSE,
  seed = py_none(),
  shape = c(),
  event = 0,
  create_obj = FALSE,
  to_jax = TRUE
)
```

Arguments

low	A numeric vector representing the lower bound of the uniform range, inclusive.
high	A numeric vector representing the upper bound of the uniform range, inclusive.
validate_args	Logical: Whether to validate parameter values. Defaults to ‘reticulate::py_none()‘.
name	A character string representing the name of the random variable within a model. This is used to uniquely identify the variable. Defaults to ’x’.
obs	A numeric vector or array of observed values. If provided, the random variable is conditioned on these values. If ‘NULL‘, the variable is treated as a latent (unobserved) variable. Defaults to ‘NULL‘.
mask	A logical vector. Optional boolean array to mask observations.
sample	A logical value that controls the function’s behavior. If ‘TRUE‘, the function will directly draw samples from the distribution. If ‘FALSE‘, it will create a random variable within a model. Defaults to ‘FALSE‘.
seed	An integer used to set the random seed for reproducibility when ‘sample = TRUE‘. This argument has no effect when ‘sample = FALSE‘, as randomness is handled by the model’s inference engine. Defaults to 0.
shape	A numeric vector. When sample=FALSE (model building), this is used with ‘.expand(shape)‘ to set the distribution’s batch shape. When sample=TRUE (direct sampling), this is used as ‘sample_shape‘ to draw a raw JAX array of the given shape.
event	Integer representing the number of batch dimensions to reinterpret as event dimensions (used in model building).

create_obj	Logical. If TRUE, returns the raw BI distribution object instead of creating a sample site. This is essential for building complex distributions like 'MixtureSameFamily'.
to_jax	Boolean. Indicates whether to return a JAX array or not.

Value

- When sample=FALSE, a BI Discrete Uniform distribution object (for model building).

- When sample=TRUE, a JAX array of samples drawn from the Discrete Uniform distribution (for direct sampling).

- When create_obj=TRUE, the raw BI distribution object (for advanced use cases).

Examples

```
library(BayesianInference)
m=importBI(platform='cpu')
bi.dist.discrete_uniform(sample = TRUE)
```

---

bi.dist.euler_maruyama

Euler-Maruyama method

Description

Euler-Maruyama methode is a method for the approximate numerical solution of a stochastic differential equation (SDE). It simulates the solution to an SDE by iteratively applying the Euler method to each time step, incorporating a random perturbation to account for the diffusion term.

Usage

```
bi.dist.euler_maruyama(
  t,
  sde_fn,
  init_dist,
  validate_args = py_none(),
  name = "x",
  obs = py_none(),
  mask = py_none(),
  sample = FALSE,
  seed = py_none(),
  shape = c(),
  event = 0,
  create_obj = FALSE,
  to_jax = TRUE
)
```

Arguments

t	A numeric vector representing the discretized time steps.
sde_fn	A function that takes the current state and time as input and returns the drift and diffusion coefficients.
init_dist	The initial distribution of the system.
validate_args	A logical value indicating whether to validate the arguments. Defaults to 'TRUE'.
name	A character string representing the name of the random variable within a model. This is used to uniquely identify the variable. Defaults to 'x'.
obs	A numeric vector or array of observed values. If provided, the random variable is conditioned on these values. If 'NULL', the variable is treated as a latent (unobserved) variable. Defaults to 'NULL'.
mask	A logical vector. Optional boolean array to mask observations.
sample	A logical value that controls the function's behavior. If 'TRUE', the function will directly draw samples from the distribution. If 'FALSE', it will create a random variable within a model. Defaults to 'FALSE'.
seed	An integer used to set the random seed for reproducibility when 'sample = TRUE'. This argument has no effect when 'sample = FALSE', as randomness is handled by the model's inference engine. Defaults to 0.
shape	A numeric vector specifying the shape of the output tensor. Defaults to 'NULL'.
event	Integer representing the number of batch dimensions to reinterpret as event dimensions (used in model building).
create_obj	Logical. If TRUE, returns the raw BI distribution object instead of creating a sample site. This is essential for building complex distributions like 'MixtureSameFamily'.
to_jax	Boolean. Indicates whether to return a JAX array or not.

Value

- When `sample=FALSE`, a BI Euler-Maruyama distribution object (for model building).

- When `sample=TRUE`, a JAX array of samples drawn from the Euler-Maruyama distribution (for direct sampling).

- When `create_obj=TRUE`, the raw BI distribution object (for advanced use cases).

Examples

```
library(BayesianInference)
m=importBI(platform='cpu')
ornstein_uhlenbeck_sde <- function(x, t) {
 # This function models dX = -theta * X dt + sigma dW
 theta <- 1.0
 sigma <- 0.5

 drift <- -theta * x
 diffusion <- sigma
```

```
 # Return a list of two elements: drift and diffusion
 # reticulate will convert this to a Python tuple
 return(list(drift, diffusion))
}
bi.dist.euler_maruyama(
t=c(0.0, 0.1, 0.2),
sde_fn = ornstein_uhlenbeck_sde,
init_dist=bi.dist.normal(0.0, 1.0, create_obj=TRUE),
sample = TRUE)
```

`bi.dist.exponential` *Samples from an Exponential distribution.*

Description

The Exponential distribution is a continuous probability distribution that models the time until an event occurs in a Poisson process, where events occur continuously and independently at a constant average rate. It is often used to model the duration of events, such as the time until a machine fails or the length of a phone call.

Usage

```
bi.dist.exponential(
  rate = 1,
  validate_args = py_none(),
  name = "x",
  obs = py_none(),
  mask = py_none(),
  sample = FALSE,
  seed = py_none(),
  shape = c(),
  event = 0,
  create_obj = FALSE,
  to_jax = TRUE
)
```

Arguments

`rate`	A numeric vector, matrix, or array representing the rate parameter, 'λ'. Must be positive.
`validate_args`	A logical value indicating whether to validate the arguments. Defaults to 'TRUE'.
`name`	A character string representing the name of the random variable within a model. This is used to uniquely identify the variable. Defaults to 'x'.
`obs`	A numeric vector or array of observed values. If provided, the random variable is conditioned on these values. If 'NULL', the variable is treated as a latent (unobserved) variable. Defaults to 'NULL'.

mask	An optional boolean vector to mask observations.
sample	A logical value that controls the function's behavior. If 'TRUE', the function will directly draw samples from the distribution. If 'FALSE', it will create a random variable within a model. Defaults to 'FALSE'.
seed	An integer used to set the random seed for reproducibility when 'sample = TRUE'. This argument has no effect when 'sample = FALSE', as randomness is handled by the model's inference engine. Defaults to 0.
shape	A numeric vector used to shape the distribution. When 'sample=False' (model building), this is used with '.expand(shape)' to set the distribution's batch shape. When 'sample=True' (direct sampling), this is used as 'sample_shape' to draw a raw JAX array of the given shape.
event	An integer representing the number of batch dimensions to reinterpret as event dimensions (used in model building).
create_obj	A logical value. If 'TRUE', returns the raw BI distribution object instead of creating a sample site. This is essential for building complex distributions like 'MixtureSameFamily'.
to_jax	Boolean. Indicates whether to return a JAX array or not.

Value

- When sample=FALSE, a BI Exponential distribution object (for model building).

- When sample=TRUE, a JAX array of samples drawn from the Exponential distribution (for direct sampling).

- When create_obj=TRUE, the raw BI distribution object (for advanced use cases).

See Also

This is a wrapper of https://num.pyro.ai/en/stable/distributions.html#exponential

Examples

```
library(BayesianInference)
m=importBI(platform='cpu')
bi.dist.exponential(rate = c(0.1,1,2),sample = TRUE)
```

---

bi.dist.gamma *Gamma Distribution*

Description

The Gamma distribution is a continuous probability distribution frequently used in Bayesian statistics, particularly as a prior distribution for variance parameters. It is defined by two positive shape parameters: concentration (k) and rate (

$$\lambda$$

).

Usage

```
bi.dist.gamma(
  concentration,
  rate = 1,
  validate_args = py_none(),
  name = "x",
  obs = py_none(),
  mask = py_none(),
  sample = FALSE,
  seed = py_none(),
  shape = c(),
  event = 0,
  create_obj = FALSE,
  to_jax = TRUE
)
```

Arguments

concentration	A numeric vector representing the shape parameter of the Gamma distribution (k > 0).
rate	A numeric vector representing the rate parameter of the Gamma distribution (theta > 0).
validate_args	Logical: Whether to validate parameter values. Defaults to 'reticulate::py_none()'.
name	A character string representing the name of the random variable within a model. This is used to uniquely identify the variable. Defaults to 'x'.
obs	A numeric vector or array of observed values. If provided, the random variable is conditioned on these values. If 'NULL', the variable is treated as a latent (unobserved) variable. Defaults to 'NULL'.
mask	A logical vector. Optional boolean array to mask observations.
sample	A logical value that controls the function's behavior. If 'TRUE', the function will directly draw samples from the distribution. If 'FALSE', it will create a random variable within a model. Defaults to 'FALSE'.
seed	An integer used to set the random seed for reproducibility when 'sample = TRUE'. This argument has no effect when 'sample = FALSE', as randomness is handled by the model's inference engine. Defaults to 0.
shape	A numeric vector. When 'sample=False' (model building), this is used with '.expand(shape)' to set the distribution's batch shape. When 'sample=True' (direct sampling), this is used as 'sample_shape' to draw a raw JAX array of the given shape.
event	An integer representing the number of batch dimensions to reinterpret as event dimensions (used in model building).
create_obj	A logical value. If 'TRUE', returns the raw BI distribution object instead of creating a sample site. This is essential for building complex distributions like 'MixtureSameFamily'.
to_jax	Boolean. Indicates whether to return a JAX array or not.

Value

- When sample=FALSE, a BI Gamma distribution object (for model building).

- When sample=TRUE, a JAX array of samples drawn from the Gamma distribution (for direct sampling).

- When create_obj=TRUE, the raw BI distribution object (for advanced use cases).

Examples

```
library(BayesianInference)
m=importBI(platform='cpu')
bi.dist.gamma(concentration = 1 , sample = TRUE)
```

---

bi.dist.gamma_poisson *Gamma-Poisson Distribution*

Description

The Gamma-Poisson distribution, also known as the Negative Binomial distribution, models overdispersed count data. It arises from a hierarchical process where the rate parameter of a Poisson distribution is itself a random variable following a Gamma distribution. This structure allows the model to capture variability in count data that exceeds what is predicted by the Poisson distribution, making it suitable for applications like modeling RNA-sequencing data and microbial count.

Usage

```
bi.dist.gamma_poisson(
  concentration,
  rate = 1,
  validate_args = py_none(),
  name = "x",
  obs = py_none(),
  mask = py_none(),
  sample = FALSE,
  seed = py_none(),
  shape = c(),
  event = 0,
  create_obj = FALSE,
  to_jax = TRUE
)
```

Arguments

concentration	A numeric vector, matrix, or array representing the shape parameter (alpha) of the Gamma distribution.
rate	A numeric vector, matrix, or array representing the rate parameter (beta) for the Gamma distribution.

validate_args	Logical: Whether to validate parameter values. Defaults to 'reticulate::py_none()'.
name	A character string representing the name of the random variable within a model. This is used to uniquely identify the variable. Defaults to 'x'.
obs	A numeric vector or array of observed values. If provided, the random variable is conditioned on these values. If 'NULL', the variable is treated as a latent (unobserved) variable. Defaults to 'NULL'.
mask	An optional boolean vector to mask observations.
sample	A logical value that controls the function's behavior. If 'TRUE', the function will directly draw samples from the distribution. If 'FALSE', it will create a random variable within a model. Defaults to 'FALSE'.
seed	An integer used to set the random seed for reproducibility when 'sample = TRUE'. This argument has no effect when 'sample = FALSE', as randomness is handled by the model's inference engine. Defaults to 0.
shape	A numeric vector used to shape the distribution. When 'sample=FALSE' (model building), this is used with '.expand(shape)' to set the distribution's batch shape. When 'sample=TRUE' (direct sampling), this is used as 'sample_shape' to draw a raw JAX array of the given shape.
event	An integer representing the number of batch dimensions to reinterpret as event dimensions (used in model building).
create_obj	A logical value. If 'TRUE', returns the raw BI distribution object instead of creating a sample site.
to_jax	Boolean. Indicates whether to return a JAX array or not.

Value

- When `sample=FALSE`, a BI Gamma-Poisson distribution object (for model building).

- When `sample=TRUE`, a JAX array of samples drawn from the Gamma-Poisson distribution (for direct sampling).

- When `create_obj=TRUE`, the raw BI distribution object (for advanced use cases).

See Also

This is a wrapper of https://num.pyro.ai/en/stable/distributions.html#gammapoisson

Examples

```
library(BayesianInference)
m=importBI(platform='cpu')
bi.dist.gamma_poisson(concentration = 1, sample = TRUE)
```

`bi.dist.gaussian_copula` *Gaussian Copula Distribution*

Description

A distribution that links the 'batch_shape[:-1]' of a marginal distribution with a multivariate Gaussian copula, odelling the correlation between the axes. A copula is a multivariate distribution over the uniform distribution on [0, 1]. The Gaussian copula links the marginal distributions through a multivariate normal distribution.

Usage

```
bi.dist.gaussian_copula(
  marginal_dist,
  correlation_matrix = py_none(),
  correlation_cholesky = py_none(),
  validate_args = py_none(),
  name = "x",
  obs = py_none(),
  mask = py_none(),
  sample = FALSE,
  seed = py_none(),
  shape = c(),
  event = 0,
  create_obj = FALSE,
  to_jax = TRUE
)
```

Arguments

`marginal_dist`	Distribution: Distribution whose last batch axis is to be coupled.
`correlation_matrix`	array_like, optional: Correlation matrix of the coupling multivariate normal distribution. Defaults to 'reticulate::py_none()'.
`correlation_cholesky`	array_like, optional: Correlation Cholesky factor of the coupling multivariate normal distribution. Defaults to 'reticulate::py_none()'.
`validate_args`	Logical: Whether to validate parameter values. Defaults to 'reticulate::py_none()'.
`name`	A character string representing the name of the random variable within a model. This is used to uniquely identify the variable. Defaults to 'x'.
`obs`	A numeric vector or array of observed values. If provided, the random variable is conditioned on these values. If 'NULL', the variable is treated as a latent (unobserved) variable. Defaults to 'NULL'.
`mask`	jnp.ndarray, bool, optional: Optional boolean array to mask observations. Defaults to 'reticulate::py_none()'.

sample	A logical value that controls the function's behavior. If 'TRUE', the function will directly draw samples from the distribution. If 'FALSE', it will create a random variable within a model. Defaults to 'FALSE'.
seed	An integer used to set the random seed for reproducibility when 'sample = TRUE'. This argument has no effect when 'sample = FALSE', as randomness is handled by the model's inference engine. Defaults to 0.
shape	numeric vector: A multi-purpose argument for shaping. When 'sample=FALSE' (model building), this is used with '.expand(shape)' to set the distribution's batch shape. When 'sample=TRUE' (direct sampling), this is used as 'sample_shape' to draw a raw JAX array of the given shape.
event	int: The number of batch dimensions to reinterpret as event dimensions (used in model building).
create_obj	bool, optional: If 'TRUE', returns the raw BI distribution object instead of creating a sample site. This is essential for building complex distributions like 'MixtureSameFamily'. Defaults to 'FALSE'.
to_jax	Boolean. Indicates whether to return a JAX array or not.

Value

- When `sample=FALSE`, a BI Gaussian Copula distribution object (for model building).

- When `sample=TRUE`, a JAX array of samples drawn from the Gaussian Copula distribution (for direct sampling).

- When `create_obj=TRUE`, the raw BI distribution object (for advanced use cases).

Examples

```
library(BayesianInference)
m=importBI(platform='cpu')
bi.dist.gaussian_copula(
  marginal_dist = bi.dist.gamma(concentration = 1 ,  create_obj = TRUE) ,
  correlation_matrix =  matrix(c(1.0, 0.7, 0.7, 1.0),, nrow = 2, byrow = TRUE),
  sample = TRUE)
```

`bi.dist.gaussian_copula_beta`

Gaussian Copula Beta distribution.

Description

This distribution combines a Gaussian copula with a Beta distribution. The Gaussian copula models the dependence structure between random variables, while the Beta distribution defines the marginal distributions of each variable.

Usage

```
bi.dist.gaussian_copula_beta(
  concentration1,
  concentration0,
  correlation_matrix = py_none(),
  correlation_cholesky = py_none(),
  validate_args = FALSE,
  name = "x",
  obs = py_none(),
  mask = py_none(),
  sample = FALSE,
  seed = py_none(),
  shape = c(),
  event = 0,
  create_obj = FALSE,
  to_jax = TRUE
)
```

Arguments

concentration1	A numeric vector or matrix representing the first shape parameter of the Beta distribution.
concentration0	A numeric vector or matrix representing the second shape parameter of the Beta distribution.
correlation_matrix	array_like, optional: Correlation matrix of the coupling multivariate normal distribution. Defaults to 'reticulate::py_none()'.
correlation_cholesky	A numeric vector, matrix, or array representing the Cholesky decomposition of the correlation matrix.
validate_args	Logical: Whether to validate parameter values. Defaults to 'reticulate::py_none()'.
name	A character string representing the name of the random variable within a model. This is used to uniquely identify the variable. Defaults to 'x'.
obs	A numeric vector or array of observed values. If provided, the random variable is conditioned on these values. If 'NULL', the variable is treated as a latent (unobserved) variable. Defaults to 'NULL'.
mask	A logical vector. Optional boolean array to mask observations.
sample	A logical value that controls the function's behavior. If 'TRUE', the function will directly draw samples from the distribution. If 'FALSE', it will create a random variable within a model. Defaults to 'FALSE'.
seed	An integer used to set the random seed for reproducibility when 'sample = TRUE'. This argument has no effect when 'sample = FALSE', as randomness is handled by the model's inference engine. Defaults to 0.
shape	A numeric vector. This is used as 'sample_shape' to draw a raw JAX array of the given shape when 'sample=True'.

event	Integer indicating the number of batch dimensions to reinterpret as event dimensions (used in model building).
create_obj	Logical. If 'TRUE', returns the raw BI distribution object instead of creating a sample site.
to_jax	Boolean. Indicates whether to return a JAX array or not.

Value

- When sample=FALSE, a BI Gaussian Copula Beta distribution object (for model building).

- When sample=TRUE, a JAX array of samples drawn from the Gaussian Copula Beta distribution (for direct sampling).

- When create_obj=TRUE, the raw BI distribution object (for advanced use cases).

See Also

This is a wrapper of https://num.pyro.ai/en/stable/distributions.html#gaussiancopulabetadistribution

Examples

```
library(BayesianInference)
m=importBI(platform='cpu')
bi.dist.gaussian_copula_beta(
  concentration1 = c(2.0, 3.0),
  concentration0 = c(5.0, 3.0),
  correlation_matrix = matrix(c(1.0, 0.7, 0.7, 1.0), nrow = 2, byrow = TRUE),
  sample = TRUE)
```

---

bi.dist.gaussian_random_walk

Samples from a Gaussian Random Walk distribution.

Description

Creates a distribution over a Gaussian random walk of a specified number of steps. This is a discrete-time stochastic process where the value at each step is the previous value plus a Gaussian-distributed increment. The distribution is over the entire path.

Usage

```
bi.dist.gaussian_random_walk(
  scale = 1,
  num_steps = 1,
  validate_args = py_none(),
  name = "x",
  obs = py_none(),
  mask = py_none(),
```

```
  sample = FALSE,
  seed = py_none(),
  shape = c(),
  event = 0,
  create_obj = FALSE,
  to_jax = TRUE
)
```

Arguments

scale	A numeric value representing the standard deviation of the Gaussian increments.
num_steps	(int, optional): The number of steps in the random walk, which determines the event shape of the distribution. Must be positive. Defaults to 1.
validate_args	Logical: Whether to validate parameter values. Defaults to 'reticulate::py_none()'.
name	A character string representing the name of the random variable within a model. This is used to uniquely identify the variable. Defaults to 'x'.
obs	A numeric vector or array of observed values. If provided, the random variable is conditioned on these values. If 'NULL', the variable is treated as a latent (unobserved) variable. Defaults to 'NULL'.
mask	A logical vector, matrix, or array. Optional boolean array to mask observations.
sample	A logical value that controls the function's behavior. If 'TRUE', the function will directly draw samples from the distribution. If 'FALSE', it will create a random variable within a model. Defaults to 'FALSE'.
seed	An integer used to set the random seed for reproducibility when 'sample = TRUE'. This argument has no effect when 'sample = FALSE', as randomness is handled by the model's inference engine. Defaults to 0.
shape	A numeric vector. When 'sample=False' (model building), this is used with '.expand(shape)' to set the distribution's batch shape. When 'sample=True' (direct sampling), this is used as 'sample_shape' to draw a raw JAX array of the given shape.
event	An integer representing the number of batch dimensions to reinterpret as event dimensions (used in model building).
create_obj	A logical value. If 'TRUE', returns the raw BI distribution object instead of creating a sample site. This is essential for building complex distributions like 'MixtureSameFamily'.
to_jax	Boolean. Indicates whether to return a JAX array or not.

Value

- When `sample=FALSE`, a BI Gaussian Random Walk distribution object (for model building).

- When `sample=TRUE`, a JAX array of samples drawn from the Gaussian Random Walk distribution (for direct sampling).

- When `create_obj=TRUE`, the raw BI distribution object (for advanced use cases).

Examples

```
library(BayesianInference)
m=importBI(platform='cpu')
bi.dist.gaussian_random_walk(scale = c(1,5,10), sample = TRUE)
```

`bi.dist.gaussian_state_space` *Gaussian State Space Distribution*

Description

Samples from a Gaussian state space model.

Usage

```
bi.dist.gaussian_state_space(
  num_steps,
  transition_matrix,
  covariance_matrix = py_none(),
  precision_matrix = py_none(),
  scale_tril = py_none(),
  validate_args = py_none(),
  name = "x",
  obs = py_none(),
  mask = py_none(),
  sample = FALSE,
  seed = py_none(),
  shape = c(),
  event = 0,
  create_obj = FALSE,
  to_jax = TRUE
)
```

Arguments

`num_steps`	An integer representing the number of steps.
`transition_matrix`	A numeric vector, matrix, or array representing the state space transition matrix A.
`covariance_matrix`	A numeric vector, matrix, or array representing the covariance of the innovation noise ϵ. Defaults to 'reticulate::py_none()'.
`precision_matrix`	A numeric vector, matrix, or array representing the precision matrix of the innovation noise ϵ. Defaults to 'reticulate::py_none()'.

scale_tril	A numeric vector, matrix, or array representing the scale matrix of the innovation noise ϵ. Defaults to 'reticulate::py_none()'.
validate_args	Logical: Whether to validate parameter values. Defaults to 'reticulate::py_none()'.
name	A character string representing the name of the random variable within a model. This is used to uniquely identify the variable. Defaults to 'x'.
obs	A numeric vector or array of observed values. If provided, the random variable is conditioned on these values. If 'NULL', the variable is treated as a latent (unobserved) variable. Defaults to 'NULL'.
mask	A logical vector, matrix, or array representing an optional boolean array to mask observations. Defaults to 'reticulate::py_none()'.
sample	A logical value that controls the function's behavior. If 'TRUE', the function will directly draw samples from the distribution. If 'FALSE', it will create a random variable within a model. Defaults to 'FALSE'.
seed	An integer used to set the random seed for reproducibility when 'sample = TRUE'. This argument has no effect when 'sample = FALSE', as randomness is handled by the model's inference engine. Defaults to 0.
shape	A numeric vector specifying the shape. When 'sample=FALSE' (model building), this is used with '.expand(shape)' to set the distribution's batch shape. When 'sample=TRUE' (direct sampling), this is used as 'sample_shape' to draw a raw JAX array of the given shape.
event	An integer representing the number of batch dimensions to reinterpret as event dimensions (used in model building).
create_obj	A logical value. If 'TRUE', returns the raw BI distribution object instead of creating a sample site. Defaults to 'FALSE'.
to_jax	Boolean. Indicates whether to return a JAX array or not.

Value

When 'sample=FALSE': - When 'sample=FALSE', a BI Gaussian State Space distribution object (for model building).

- When 'sample=TRUE', a JAX array of samples drawn from the Gaussian State Space distribution (for direct sampling).

- When 'create_obj=TRUE', the raw BI distribution object (for advanced use cases).

See Also

This is a wrapper of https://num.pyro.ai/en/stable/distributions.html#gaussianstatespace

Examples

```
library(BayesianInference)
m=importBI(platform='cpu')
bi.dist.gaussian_state_space(
  num_steps = 1,
  transition_matrix = matrix(c(0.5), nrow = 1, byrow = TRUE),
  covariance_matrix = matrix(c(1.0), nrow = 1, byrow = TRUE),
```

```
  sample = TRUE)
```

`bi.dist.geometric` *Samples from a Geometric distribution.*

Description

Samples from a Geometric distribution, which models the number of failures before the first success in a sequence of independent Bernoulli trials. It is parameterized by logits, which are transformed into probabilities using the sigmoid function.

Usage

```
bi.dist.geometric(
  probs = py_none(),
  logits = py_none(),
  validate_args = py_none(),
  name = "x",
  obs = py_none(),
  mask = py_none(),
  sample = FALSE,
  seed = py_none(),
  shape = c(),
  event = 0,
  create_obj = FALSE,
  to_jax = TRUE
)
```

Arguments

`probs`	A numeric vector, matrix, or array representing the probability of success on each trial. Must be between 0 and 1.
`logits`	A numeric vector, matrix, or array representing the log-odds of success on each trial. ‘probs = jax.nn.sigmoid(logits)‘.
`validate_args`	Logical: Whether to validate parameter values. Defaults to ‘reticulate::py_none()‘.
`name`	A character string representing the name of the random variable within a model. This is used to uniquely identify the variable. Defaults to ’x’.
`obs`	A numeric vector or array of observed values. If provided, the random variable is conditioned on these values. If ‘NULL‘, the variable is treated as a latent (unobserved) variable. Defaults to ‘NULL‘.
`mask`	A logical vector, matrix, or array to mask observations.
`sample`	A logical value that controls the function’s behavior. If ‘TRUE‘, the function will directly draw samples from the distribution. If ‘FALSE‘, it will create a random variable within a model. Defaults to ‘FALSE‘.

seed	An integer used to set the random seed for reproducibility when 'sample = TRUE'. This argument has no effect when 'sample = FALSE', as randomness is handled by the model's inference engine. Defaults to 0.
shape	A numeric vector specifying the shape of the output. Used to set the distribution's batch shape when sample=FALSE (model building) or as 'sample_shape' to draw a raw JAX array of the given shape when sample=TRUE (direct sampling).
event	An integer representing the number of batch dimensions to reinterpret as event dimensions (used in model building).
create_obj	A logical value. If 'TRUE', returns the raw BI distribution object instead of creating a sample site.
to_jax	Boolean. Indicates whether to return a JAX array or not.

Value

When sample=FALSE: A BI Geometric distribution object (for model building).

When sample=TRUE: A JAX array of samples drawn from the Geometric distribution (for direct sampling).

When create_obj=TRUE: The raw BI distribution object (for advanced use cases).

See Also

This is a wrapper of https://num.pyro.ai/en/stable/distributions.html#geometric

Examples

```
library(BayesianInference)
m=importBI(platform='cpu')
bi.dist.geometric(logits = 0.5, sample = TRUE)
bi.dist.geometric(probs = 0.5, sample = TRUE)
```

---

bi.dist.gompertz *Gompertz Distribution*

Description

The Gompertz distribution is a distribution with support on the positive real line that is closely related to the Gumbel distribution. This implementation follows the notation used in the Wikipedia entry for the Gompertz distribution. See https://en.wikipedia.org/wiki/Gompertz_distribution.

Usage

```
bi.dist.gompertz(
  concentration,
  rate = 1,
  validate_args = py_none(),
  name = "x",
  obs = py_none(),
  mask = py_none(),
  sample = FALSE,
  seed = py_none(),
  shape = c(),
  event = 0,
  create_obj = FALSE,
  to_jax = TRUE
)
```

Arguments

concentration	A positive numeric vector, matrix, or array representing the concentration parameter.
rate	A positive numeric vector, matrix, or array representing the rate parameter.
validate_args	Logical: Whether to validate parameter values. Defaults to 'reticulate::py_none()'.
name	A character string representing the name of the random variable within a model. This is used to uniquely identify the variable. Defaults to 'x'.
obs	A numeric vector or array of observed values. If provided, the random variable is conditioned on these values. If 'NULL', the variable is treated as a latent (unobserved) variable. Defaults to 'NULL'.
mask	A boolean vector, matrix, or array representing an optional mask for observations.
sample	A logical value that controls the function's behavior. If 'TRUE', the function will directly draw samples from the distribution. If 'FALSE', it will create a random variable within a model. Defaults to 'FALSE'.
seed	An integer used to set the random seed for reproducibility when 'sample = TRUE'. This argument has no effect when 'sample = FALSE', as randomness is handled by the model's inference engine. Defaults to 0.
shape	A numeric vector representing the shape parameter.
event	Integer representing the number of batch dimensions to reinterpret as event dimensions.
create_obj	Logical value. If 'TRUE', returns the raw BI distribution object instead of creating a sample site.
to_jax	Boolean. Indicates whether to return a JAX array or not.

Value

A BI Gompertz distribution object when `sample=FALSE` (for model building).

A JAX array when `sample=TRUE` (for direct sampling).

A BI distribution object when `create_obj=TRUE` (for advanced use cases).

See Also

This is a wrapper of https://num.pyro.ai/en/stable/distributions.html#gompertz

Examples

```
library(BayesianInference)
m=importBI(platform='cpu')
bi.dist.gompertz(concentration = 5., sample = TRUE)
```

`bi.dist.gumbel` *Samples from a Gumbel (or Extreme Value) distribution.*

Description

The Gumbel distribution is a continuous probability distribution named after German mathematician Carl Gumbel. It is often used to model the distribution of maximum values in a sequence of independent random variables.

Usage

```
bi.dist.gumbel(
  loc = 0,
  scale = 1,
  validate_args = py_none(),
  name = "x",
  obs = py_none(),
  mask = py_none(),
  sample = FALSE,
  seed = py_none(),
  shape = c(),
  event = 0,
  create_obj = FALSE,
  to_jax = TRUE
)
```

Arguments

`loc`	Location parameter.
`scale`	Scale parameter. Must be positive.
`validate_args`	Logical: Whether to validate parameter values. Defaults to ‘reticulate::py_none()‘.
`name`	A character string representing the name of the random variable within a model. This is used to uniquely identify the variable. Defaults to ’x’.

obs	A numeric vector or array of observed values. If provided, the random variable is conditioned on these values. If 'NULL', the variable is treated as a latent (unobserved) variable. Defaults to 'NULL'.
mask	Optional boolean array to mask observations.
sample	A logical value that controls the function's behavior. If 'TRUE', the function will directly draw samples from the distribution. If 'FALSE', it will create a random variable within a model. Defaults to 'FALSE'.
seed	An integer used to set the random seed for reproducibility when 'sample = TRUE'. This argument has no effect when 'sample = FALSE', as randomness is handled by the model's inference engine. Defaults to 0.
shape	A numeric vector. When sample=FALSE (model building), this is used with .expand(shape) to set the distribution's batch shape. When sample=TRUE (direct sampling), this is used as sample_shape to draw a raw JAX array of the given shape.
event	The number of batch dimensions to reinterpret as event dimensions (used in model building).
create_obj	If TRUE, returns the raw BI distribution object instead of creating a sample site.
to_jax	Boolean. Indicates whether to return a JAX array or not.

Value

- When sample=FALSE, a BI Gumbel distribution object (for model building).

- When sample=TRUE, a JAX array of samples drawn from the Gumbel distribution (for direct sampling).

- When create_obj=TRUE, the raw BI distribution object (for advanced use cases).

Examples

```
library(BayesianInference)
m=importBI(platform='cpu')
bi.dist.gumbel(sample = TRUE)
```

---

bi.dist.half_cauchy *HalfCauchy Distribution*

Description

The HalfCauchy distribution is a probability distribution that is half of the Cauchy distribution. It is defined on the positive real numbers and is often used in situations where only positive values are relevant.

Usage

```
bi.dist.half_cauchy(
  scale = 1,
  validate_args = py_none(),
  name = "x",
  obs = py_none(),
  mask = py_none(),
  sample = FALSE,
  seed = py_none(),
  shape = c(),
  event = 0,
  create_obj = FALSE,
  to_jax = TRUE
)
```

Arguments

scale	A numeric vector representing the scale parameter of the Cauchy distribution. Must be positive.
validate_args	Logical: Whether to validate parameter values. Defaults to 'reticulate::py_none()'.
name	A character string representing the name of the random variable within a model. This is used to uniquely identify the variable. Defaults to 'x'.
obs	A numeric vector or array of observed values. If provided, the random variable is conditioned on these values. If 'NULL', the variable is treated as a latent (unobserved) variable. Defaults to 'NULL'.
mask	A logical vector, optionally used to mask observations.
sample	A logical value that controls the function's behavior. If 'TRUE', the function will directly draw samples from the distribution. If 'FALSE', it will create a random variable within a model. Defaults to 'FALSE'.
seed	An integer used to set the random seed for reproducibility when 'sample = TRUE'. This argument has no effect when 'sample = FALSE', as randomness is handled by the model's inference engine. Defaults to 0.
shape	A numeric vector used for shaping. When 'sample=False' (model building), this is used with '.expand(shape)' to set the distribution's batch shape. When 'sample=True' (direct sampling), this is used as 'sample_shape' to draw a raw JAX array of the given shape.
event	Integer specifying the number of batch dimensions to reinterpret as event dimensions (used in model building).
create_obj	Logical; If 'TRUE', returns the raw BI distribution object instead of creating a sample site.
to_jax	Boolean. Indicates whether to return a JAX array or not.

Value

- When sample=FALSE, a BI HalfCauchy distribution object (for model building).

- When sample=TRUE, a JAX array of samples drawn from the HalfCauchy distribution (for direct sampling).

- When create_obj=TRUE, the raw BI distribution object (for advanced use cases).

See Also

This is a wrapper of https://num.pyro.ai/en/stable/distributions.html#halfcauchy

Examples

```
library(BayesianInference)
m=importBI(platform='cpu')
bi.dist.half_cauchy(sample = TRUE)
```

---

bi.dist.half_normal *Samples from a HalfNormal distribution.*

Description

The HalfNormal distribution is a distribution of the absolute value of a normal random variable. It is defined by a location parameter (implicitly 0) and a scale parameter.

Usage

```
bi.dist.half_normal(
  scale = 1,
  validate_args = py_none(),
  name = "x",
  obs = py_none(),
  mask = py_none(),
  sample = FALSE,
  seed = py_none(),
  shape = c(),
  event = 0,
  create_obj = FALSE,
  to_jax = TRUE
)
```

Arguments

scale	A numeric vector or array representing the scale parameter of the distribution. Must be positive.
validate_args	Logical: Whether to validate parameter values. Defaults to ‘reticulate::py_none()‘.
name	A character string representing the name of the random variable within a model. This is used to uniquely identify the variable. Defaults to ’x’.

obs	A numeric vector or array of observed values. If provided, the random variable is conditioned on these values. If 'NULL', the variable is treated as a latent (unobserved) variable. Defaults to 'NULL'.
mask	A logical vector or array representing an optional boolean array to mask observations.
sample	A logical value that controls the function's behavior. If 'TRUE', the function will directly draw samples from the distribution. If 'FALSE', it will create a random variable within a model. Defaults to 'FALSE'.
seed	An integer used to set the random seed for reproducibility when 'sample = TRUE'. This argument has no effect when 'sample = FALSE', as randomness is handled by the model's inference engine. Defaults to 0.
shape	A numeric vector used for shaping. When `sample=FALSE` (model building), this is used with '.expand(shape)' to set the distribution's batch shape. When `sample=TRUE` (direct sampling), this is used as 'sample_shape' to draw a raw JAX array of the given shape.
event	An integer representing the number of batch dimensions to reinterpret as event dimensions (used in model building).
create_obj	A logical value. If 'TRUE', returns the raw BI distribution object instead of creating a sample site. This is essential for building complex distributions like 'MixtureSameFamily'.
to_jax	Boolean. Indicates whether to return a JAX array or not.

Value

- When `sample=FALSE`, a BI HalfNormal distribution object (for model building).

- When `sample=TRUE`, a JAX array of samples drawn from the HalfNormal distribution (for direct sampling).

- When `create_obj=TRUE`, the raw BI distribution object (for advanced use cases).

Examples

```
library(BayesianInference)
m=importBI(platform='cpu')
bi.dist.half_normal(sample = TRUE)
```

`bi.dist.inverse_gamma` *InverseGamma Distribution*

Description

The InverseGamma distribution is a two-parameter family of continuous probability distributions. It is defined by its shape

$$\alpha$$

and rate

$$\beta$$

parameters. It is often used as a prior distribution for precision parameters (inverse variance) in Bayesian statistics.

Usage

```
bi.dist.inverse_gamma(
  concentration,
  rate = 1,
  validate_args = py_none(),
  name = "x",
  obs = py_none(),
  mask = py_none(),
  sample = FALSE,
  seed = py_none(),
  shape = c(),
  event = 0,
  create_obj = FALSE,
  to_jax = TRUE
)
```

Arguments

concentration	A numeric vector representing the shape parameter (\alpha) of the InverseGamma distribution. Must be positive.
rate	A numeric vector representing the rate parameter (\beta) of the InverseGamma distribution. Must be positive.
validate_args	Logical: Whether to validate parameter values. Defaults to 'reticulate::py_none()'.
name	A character string representing the name of the random variable within a model. This is used to uniquely identify the variable. Defaults to 'x'.
obs	A numeric vector or array of observed values. If provided, the random variable is conditioned on these values. If 'NULL', the variable is treated as a latent (unobserved) variable. Defaults to 'NULL'.
mask	A logical vector. Optional boolean array to mask observations.
sample	A logical value that controls the function's behavior. If 'TRUE', the function will directly draw samples from the distribution. If 'FALSE', it will create a random variable within a model. Defaults to 'FALSE'.
seed	An integer used to set the random seed for reproducibility when 'sample = TRUE'. This argument has no effect when 'sample = FALSE', as randomness is handled by the model's inference engine. Defaults to 0.

shape	A numeric vector. When ‘sample=False‘ (model building), this is used with ‘.expand(shape)‘ to set the distribution’s batch shape. When ‘sample=True‘ (direct sampling), this is used as ‘sample_shape‘ to draw a raw JAX array of the given shape.
event	An integer representing the number of batch dimensions to reinterpret as event dimensions (used in model building).
create_obj	A logical value. If ‘TRUE‘, returns the raw BI distribution object instead of creating a sample site. This is essential for building complex distributions like ‘MixtureSameFamily‘.
to_jax	Boolean. Indicates whether to return a JAX array or not.

Value

- When `sample=FALSE`: A BI InverseGamma distribution object (for model building).

- When `sample=TRUE`: A JAX array of samples drawn from the InverseGamma distribution (for direct sampling).

- When `create_obj=TRUE`: The raw BI distribution object (for advanced use cases).

See Also

This is a wrapper of https://num.pyro.ai/en/stable/distributions.html#inversegamma

Examples

```
library(BayesianInference)
m=importBI(platform='cpu')
bi.dist.inverse_gamma(concentration = 5., sample = TRUE)
```

---

bi.dist.kumaraswamy *Kumaraswamy Distribution*

Description

The Kumaraswamy distribution is a continuous probability distribution defined on the interval [0, 1]. It is a flexible distribution that can take on various shapes depending on its parameters.

$$f(x; a, b) = abx^{ab-1}(1-x)^{b-1}$$

Usage

```
bi.dist.kumaraswamy(
  concentration1,
  concentration0,
  validate_args = py_none(),
  name = "x",
```

```
  obs = py_none(),
  mask = py_none(),
  sample = FALSE,
  seed = py_none(),
  shape = c(),
  event = 0,
  create_obj = FALSE,
  to_jax = TRUE
)
```

Arguments

concentration1	A numeric vector, matrix, or array representing the first shape parameter. Must be positive.
concentration0	A numeric vector, matrix, or array representing the second shape parameter. Must be positive.
validate_args	Logical: Whether to validate parameter values. Defaults to 'reticulate::py_none()'.
name	A character string representing the name of the random variable within a model. This is used to uniquely identify the variable. Defaults to 'x'.
obs	A numeric vector or array of observed values. If provided, the random variable is conditioned on these values. If 'NULL', the variable is treated as a latent (unobserved) variable. Defaults to 'NULL'.
mask	A logical vector, matrix, or array. Optional boolean array to mask observations.
sample	A logical value that controls the function's behavior. If 'TRUE', the function will directly draw samples from the distribution. If 'FALSE', it will create a random variable within a model. Defaults to 'FALSE'.
seed	An integer used to set the random seed for reproducibility when 'sample = TRUE'. This argument has no effect when 'sample = FALSE', as randomness is handled by the model's inference engine. Defaults to 0.
shape	A numeric vector. When sample=FALSE (model building), this is used with '.expand(shape)' to set the distribution's batch shape. When sample=TRUE (direct sampling), this is used as 'sample_shape' to draw a raw JAX array of the given shape.
event	An integer representing the number of batch dimensions to reinterpret as event dimensions (used in model building).
create_obj	A logical value. If 'TRUE', returns the raw BI distribution object instead of creating a sample site. This is essential for building complex distributions like 'MixtureSameFamily'.
to_jax	Boolean. Indicates whether to return a JAX array or not.

Value

- When sample=FALSE: A BI Kumaraswamy distribution object (for model building).

- When sample=TRUE: A JAX array of samples drawn from the Kumaraswamy distribution (for direct sampling).

- When create_obj=TRUE: The raw BI distribution object (for advanced use cases).

See Also

This is a wrapper of https://num.pyro.ai/en/stable/distributions.html#kumaraswamy

Examples

```
library(BayesianInference)
m=importBI(platform='cpu')
bi.dist.kumaraswamy(concentration1 = 5, concentration0 = 1., sample = TRUE)
```

---

bi.dist.laplace *Laplace Distribution*

Description

Samples from a Laplace distribution, also known as the double exponential distribution. The Laplace distribution is defined by its location parameter (loc) and scale parameter (scale).

Usage

```
bi.dist.laplace(
  loc = 0,
  scale = 1,
  validate_args = py_none(),
  name = "x",
  obs = py_none(),
  mask = py_none(),
  sample = FALSE,
  seed = py_none(),
  shape = c(),
  event = 0,
  create_obj = FALSE,
  to_jax = TRUE
)
```

Arguments

loc	A numeric vector representing the location parameter of the Laplace distribution.
scale	A numeric vector representing the scale parameter of the Laplace distribution. Must be positive.
validate_args	Logical: Whether to validate parameter values. Defaults to 'reticulate::py_none()'.
name	A character string representing the name of the random variable within a model. This is used to uniquely identify the variable. Defaults to 'x'.
obs	A numeric vector or array of observed values. If provided, the random variable is conditioned on these values. If 'NULL', the variable is treated as a latent (unobserved) variable. Defaults to 'NULL'.

mask	A logical vector, optionally used to mask observations.
sample	A logical value that controls the function's behavior. If 'TRUE', the function will directly draw samples from the distribution. If 'FALSE', it will create a random variable within a model. Defaults to 'FALSE'.
seed	An integer used to set the random seed for reproducibility when 'sample = TRUE'. This argument has no effect when 'sample = FALSE', as randomness is handled by the model's inference engine. Defaults to 0.
shape	A numeric vector used for shaping. When `sample=FALSE` (model building), this is used with '.expand(shape)' to set the distribution's batch shape. When `sample=TRUE` (direct sampling), this is used as 'sample_shape' to draw a raw JAX array of the given shape.
event	Integer representing the number of batch dimensions to reinterpret as event dimensions (used in model building).
create_obj	Logical; If TRUE, returns the raw BI distribution object instead of creating a sample site. This is essential for building complex distributions like 'MixtureSameFamily'.
to_jax	Logical. Defaults to TRUE.

Value

- When `sample=FALSE`: A BI Laplace distribution object (for model building).

- When `sample=TRUE`: A JAX array of samples drawn from the Laplace distribution (for direct sampling).

- When `create_obj=TRUE`: The raw BI distribution object (for advanced use cases).

See Also

This is a wrapper of https://num.pyro.ai/en/stable/distributions.html#laplace

Examples

```
library(BayesianInference)
m <- importBI(platform = "cpu")
bi.dist.laplace(sample = TRUE)
```

`bi.dist.left_truncated_distribution`

Samples from a left-truncated distribution.

Description

A left-truncated distribution is a probability distribution obtained by restricting the support of another distribution to values greater than a specified lower bound. This is useful when dealing with data that is known to be greater than a certain value. All the "mass" below (or equal to) (a) is excluded (not just unobserved, but removed from the sample/analysis).

$$f(x) = \begin{cases} \frac{f(x)}{P(X>\text{low})} & \text{if } x > \text{low} \\ 0 & \text{otherwise} \end{cases}$$

Usage

```
bi.dist.left_truncated_distribution(
  base_dist,
  low = 0,
  validate_args = py_none(),
  name = "x",
  obs = py_none(),
  mask = py_none(),
  sample = FALSE,
  seed = py_none(),
  shape = c(),
  event = 0,
  create_obj = FALSE,
  to_jax = TRUE
)
```

Arguments

base_dist	The base distribution to truncate. Must be univariate and have real support.
low	The lower truncation bound. Values less than this are excluded from the distribution.
validate_args	Logical: Whether to validate parameter values. Defaults to 'reticulate::py_none()'.
name	A character string representing the name of the random variable within a model. This is used to uniquely identify the variable. Defaults to 'x'.
obs	A numeric vector or array of observed values. If provided, the random variable is conditioned on these values. If 'NULL', the variable is treated as a latent (unobserved) variable. Defaults to 'NULL'.
mask	An optional boolean vector to mask observations.
sample	A logical value that controls the function's behavior. If 'TRUE', the function will directly draw samples from the distribution. If 'FALSE', it will create a random variable within a model. Defaults to 'FALSE'.
seed	An integer used to set the random seed for reproducibility when 'sample = TRUE'. This argument has no effect when 'sample = FALSE', as randomness is handled by the model's inference engine. Defaults to 0.

shape	A numeric vector. When sample=FALSE (model building), this is used with '.expand(shape)' to set the distribution's batch shape. When sample=TRUE (direct sampling), this is used as 'sample_shape' to draw a raw JAX array of the given shape.
event	The number of batch dimensions to reinterpret as event dimensions (used in model building).
create_obj	A logical value. If 'TRUE', returns the raw BI distribution object instead of creating a sample site.
to_jax	Boolean. Indicates whether to return a JAX array or not.

Value

- When sample=FALSE: A BI LeftTruncatedDistribution distribution object (for model building).

- When sample=TRUE: A JAX array of samples drawn from the LeftTruncatedDistribution distribution (for direct sampling).

- When create_obj=TRUE: The raw BI distribution object (for advanced use cases).

See Also

This is a wrapper of https://num.pyro.ai/en/stable/distributions.html#lefttruncateddistribution

Examples

```
library(BayesianInference)
m=importBI(platform='cpu')
bi.dist.left_truncated_distribution(
base_dist = bi.dist.normal(loc = 1, scale = 10 ,  create_obj = TRUE),
sample = TRUE)
```

---

bi.dist.levy *Levy distribution*

Description

The Levy distribution (or L\'evy) is a continuous probability distribution on the positive real line (or shifted positive line) that is heavy-tailed, skewed, and arises naturally in connection with stable distributions, specifically the case with stability index $\alpha = 1/2$. It is often used in contexts such as hitting-time problems for Brownian motion, physics (e.g., van der Waals line-shapes), and modelling very heavy-tailed phenomena. Let X be a Levy-distributed random variable with location parameter μ and scale parameter $c > 0$. The support is $x \geq \mu$.

Usage

```
bi.dist.levy(
  loc,
  scale,
  validate_args = py_none(),
  name = "x",
  obs = py_none(),
  mask = py_none(),
  sample = FALSE,
  seed = py_none(),
  shape = c(),
  event = 0,
  create_obj = FALSE,
  to_jax = TRUE
)
```

Arguments

loc	A numeric vector, matrix, or array representing the location parameter.
scale	A numeric vector, matrix, or array representing the scale parameter.
validate_args	Logical: Whether to validate parameter values. Defaults to 'reticulate::py_none()'.
name	A character string representing the name of the random variable within a model. This is used to uniquely identify the variable. Defaults to 'x'.
obs	A numeric vector or array of observed values. If provided, the random variable is conditioned on these values. If 'NULL', the variable is treated as a latent (unobserved) variable. Defaults to 'NULL'.
mask	A logical vector, matrix, or array to mask observations.
sample	A logical value that controls the function's behavior. If 'TRUE', the function will directly draw samples from the distribution. If 'FALSE', it will create a random variable within a model. Defaults to 'FALSE'.
seed	An integer used to set the random seed for reproducibility when 'sample = TRUE'. This argument has no effect when 'sample = FALSE', as randomness is handled by the model's inference engine. Defaults to 0.
shape	A numeric vector used for shaping. When 'sample=FALSE' (model building), this is used with '.expand(shape)' to set the distribution's batch shape. When 'sample=TRUE' (direct sampling), this is used as 'sample_shape' to draw a raw JAX array of the given shape.
event	Integer representing the number of batch dimensions to reinterpret as event dimensions (used in model building).
create_obj	Logical; If 'TRUE', returns the raw BI distribution object instead of creating a sample site.
to_jax	Logical. Defaults to TRUE.

Value

- When `sample=FALSE`, a BI Levy distribution object (for model building).
- When `sample=TRUE`, a JAX array of samples drawn from the Levy distribution (for direct sampling).
- When `create_obj=TRUE`, the raw BI distribution object (for advanced use cases).

See Also

This is a wrapper of https://num.pyro.ai/en/stable/distributions.html#levy

Examples

```
library(BayesianInference)
m <- importBI(platform = "cpu")
bi.dist.levy(loc = 1, scale = 10, sample = TRUE)
```

---

bi.dist.lkj — *Samples from an LKJ (Lewandowski, Kurowicka, Joe) distribution for correlation matrices.*

Description

The LKJ distribution is controlled by the concentration parameter

$$\eta$$

to make the probability of the correlation matrix M proportional to

$$\det(M)^{\eta-1}$$

. When

$$\eta = 1$$

, the distribution is uniform over correlation matrices. When

$$\eta > 1$$

, the distribution favors samples with large determinants. When

$$\eta < 1$$

, the distribution favors samples with small determinants.

Usage

```
bi.dist.lkj(
  dimension,
  concentration = 1,
  sample_method = "onion",
  validate_args = py_none(),
  name = "x",
  obs = py_none(),
  mask = py_none(),
  sample = FALSE,
  seed = py_none(),
  shape = c(),
  event = 0,
  create_obj = FALSE,
  to_jax = TRUE
)
```

Arguments

dimension	An integer representing the dimension of the correlation matrices.
concentration	A numeric vector representing the concentration/shape parameter of the distribution (often referred to as eta). Must be positive.
sample_method	(str): Either "cvine" or "onion". Methods proposed offer the same distribution over correlation matrices. But they are different in how to generate samples. Defaults to "onion".
validate_args	Logical: Whether to validate parameter values. Defaults to ‘reticulate::py_none()‘.
name	A character string representing the name of the random variable within a model. This is used to uniquely identify the variable. Defaults to ’x’.
obs	A numeric vector or array of observed values. If provided, the random variable is conditioned on these values. If ‘NULL‘, the variable is treated as a latent (unobserved) variable. Defaults to ‘NULL‘.
mask	An optional boolean vector to mask observations.
sample	A logical value that controls the function’s behavior. If ‘TRUE‘, the function will directly draw samples from the distribution. If ‘FALSE‘, it will create a random variable within a model. Defaults to ‘FALSE‘.
seed	An integer used to set the random seed for reproducibility when ‘sample = TRUE‘. This argument has no effect when ‘sample = FALSE‘, as randomness is handled by the model’s inference engine. Defaults to 0.
shape	A numeric vector used for shaping. When ‘sample=False‘ (model building), this is used with ‘.expand(shape)‘ to set the distribution’s batch shape. When ‘sample=True‘ (direct sampling), this is used as ‘sample_shape‘ to draw a raw JAX array of the given shape.
event	An integer representing the number of batch dimensions to reinterpret as event dimensions (used in model building).

create_obj	A logical value. If 'TRUE', returns the raw BI distribution object instead of creating a sample site.
to_jax	Boolean. Indicates whether to return a JAX array or not.

Value

- When sample=FALSE: A BI LKJ distribution object (for model building).

- When sample=TRUE: A JAX array of samples drawn from the LKJ distribution (for direct sampling).

- When create_obj=TRUE: The raw BI distribution object (for advanced use cases).

See Also

This is a wrapper of https://num.pyro.ai/en/stable/distributions.html#lkj

Examples

```
library(BayesianInference)
m=importBI(platform='cpu')
bi.dist.lkj( dimension = 2, concentration=1.0, shape = c(1), sample = TRUE)
```

---

bi.dist.lkj_cholesky *LKJ Cholesky Distribution*

Description

The LKJ (Leonard-Kjærgaard-Jørgensen) Cholesky distribution is a family of distributions on symmetric matrices, often used as a prior for the Cholesky decomposition of a symmetric matrix. It is particularly useful in Bayesian inference for models with covariance structure.

Usage

```
bi.dist.lkj_cholesky(
  dimension,
  concentration = 1,
  sample_method = "onion",
  validate_args = py_none(),
  name = "x",
  obs = py_none(),
  mask = py_none(),
  sample = FALSE,
  seed = py_none(),
  shape = c(),
  event = 0,
  create_obj = FALSE,
  to_jax = TRUE
)
```

Arguments

dimension	Numeric for the dimensions of the LKJ Cholesky matrix.
concentration	Numeric. A parameter controlling the concentration of the distribution around the identity matrix. Higher values indicate greater concentration. Must be greater than 1.
sample_method	onion
validate_args	Logical: Whether to validate parameter values. Defaults to 'reticulate::py_none()'.
name	A character string representing the name of the random variable within a model. This is used to uniquely identify the variable. Defaults to 'x'.
obs	A numeric vector or array of observed values. If provided, the random variable is conditioned on these values. If 'NULL', the variable is treated as a latent (unobserved) variable. Defaults to 'NULL'.
mask	Logical vector; Optional boolean array to mask observations.
sample	A logical value that controls the function's behavior. If 'TRUE', the function will directly draw samples from the distribution. If 'FALSE', it will create a random variable within a model. Defaults to 'FALSE'.
seed	An integer used to set the random seed for reproducibility when 'sample = TRUE'. This argument has no effect when 'sample = FALSE', as randomness is handled by the model's inference engine. Defaults to 0.
shape	Numeric vector; A multi-purpose argument for shaping. When sample=FALSE (model building), is used with '.expand(shape)' to set the distribution's batch shape. When sample=TRUE (direct sampling), this is used as 'sample_shape' to draw a raw JAX array of the given shape.
event	Numeric; The number of batch dimensions to reinterpret as event dimensions (used in model building).
create_obj	Logical; If TRUE, returns the raw BI distribution object instead of creating a sample site. This is essential for building complex distributions like 'MixtureSameFamily'.
to_jax	Boolean. Indicates whether to return a JAX array or not.

Value

- When sample=FALSE: A BI LKJ Cholesky distribution object (for model building).

- When sample=TRUE: A JAX array of samples drawn from the LKJ Cholesky distribution (for direct sampling).

- When create_obj=TRUE: The raw BI distribution object (for advanced use cases).

Examples

```
library(BayesianInference)
m <- importBI(platform='cpu')
bi.dist.lkj_cholesky(dimension = 2, concentration = 1., sample = TRUE)
```

`bi.dist.log_normal` *Log Normal distribution*

Description

The Log-Normal distribution is a probability distribution defined for positive real-valued random variables, parameterized by a location parameter (loc:

$$\mu$$

) and a scale parameter (scale). It arises when the logarithm of a random variable is normally distributed. It is often used when modeling parameters that must be positive.

Usage

```
bi.dist.log_normal(
  loc = 0,
  scale = 1,
  validate_args = py_none(),
  name = "x",
  obs = py_none(),
  mask = py_none(),
  sample = FALSE,
  seed = py_none(),
  shape = c(),
  event = 0,
  create_obj = FALSE,
  to_jax = TRUE
)
```

Arguments

`loc`	Numeric; Location parameter.
`scale`	Numeric; Scale parameter.
`validate_args`	Logical: Whether to validate parameter values. Defaults to 'reticulate::py_none()'.
`name`	A character string representing the name of the random variable within a model. This is used to uniquely identify the variable. Defaults to 'x'.
`obs`	A numeric vector or array of observed values. If provided, the random variable is conditioned on these values. If 'NULL', the variable is treated as a latent (unobserved) variable. Defaults to 'NULL'.
`mask`	Logical vector; Optional boolean array to mask observations.
`sample`	A logical value that controls the function's behavior. If 'TRUE', the function will directly draw samples from the distribution. If 'FALSE', it will create a random variable within a model. Defaults to 'FALSE'.

seed	An integer used to set the random seed for reproducibility when ‘sample = TRUE‘. This argument has no effect when ‘sample = FALSE‘, as randomness is handled by the model’s inference engine. Defaults to 0.
shape	Numeric vector; A multi-purpose argument for shaping. When ‘sample=False‘ (model building), this is used with ‘.expand(shape)‘ to set the distribution’s batch shape. When ‘sample=True‘ (direct sampling), this is used as ‘sample_shape‘ to draw a raw JAX array of the given shape.
event	Numeric; The number of batch dimensions to reinterpret as event dimensions (used in model building).
create_obj	Logical; If True, returns the raw BI distribution object instead of creating a sample site. This is essential for building complex distributions like ‘MixtureSameFamily‘.
to_jax	Boolean. Indicates whether to return a JAX array or not.

Value

- When sample=FALSE, a BI Log Normal distribution object (for model building).

- When sample=TRUE, a JAX array of samples drawn from the Log Normal distribution (for direct sampling).

- When create_obj=TRUE, the raw BI distribution object (for advanced use cases).

See Also

This is a wrapper of https://num.pyro.ai/en/stable/distributions.html#lognormal

Examples

```
library(BayesianInference)
m=importBI(platform='cpu')
bi.dist.log_normal(sample = TRUE)
```

---

bi.dist.log_uniform *Samples from a Log Uniform distribution.*

Description

A random variable X is **log-uniform** on

$$[a, b]$$

, with $0 < a < b$, if

$$\ln X$$

is uniformly distributed on

$$[\ln a, \ln b]$$

. Equivalently, the density of X is proportional to $1/x$ over that interval. This distribution is sometimes called the *reciprocal distribution*. It is useful in modeling scales spanning several orders of magnitude, where you want every decade (or log-interval) to have equal weight.

Usage

```
bi.dist.log_uniform(
  low,
  high,
  validate_args = py_none(),
  name = "x",
  obs = py_none(),
  mask = py_none(),
  sample = FALSE,
  seed = py_none(),
  shape = c(),
  event = 0,
  create_obj = FALSE,
  to_jax = TRUE
)
```

Arguments

low	A numeric vector representing the lower bound of the uniform distribution's log-space. Must be positive.
high	A numeric vector representing the upper bound of the uniform distribution's log-space. Must be positive.
validate_args	Logical: Whether to validate parameter values. Defaults to 'reticulate::py_none()'.
name	A character string representing the name of the random variable within a model. This is used to uniquely identify the variable. Defaults to 'x'.
obs	A numeric vector or array of observed values. If provided, the random variable is conditioned on these values. If 'NULL', the variable is treated as a latent (unobserved) variable. Defaults to 'NULL'.
mask	A logical vector to mask observations.
sample	A logical value that controls the function's behavior. If 'TRUE', the function will directly draw samples from the distribution. If 'FALSE', it will create a random variable within a model. Defaults to 'FALSE'.
seed	An integer used to set the random seed for reproducibility when 'sample = TRUE'. This argument has no effect when 'sample = FALSE', as randomness is handled by the model's inference engine. Defaults to 0.
shape	A numeric vector specifying the shape of the output. When sample=FALSE (model building), this is used with '.expand(shape)' to set the distribution's batch shape. When sample=TRUE (direct sampling), this is used as 'sample_shape' to draw a raw JAX array of the given shape.
event	Integer specifying the number of batch dimensions to reinterpret as event dimensions (used in model building).
create_obj	Logical; If TRUE, returns the raw BI distribution object instead of creating a sample site.
to_jax	Boolean. Indicates whether to return a JAX array or not.

Value

- When sample=FALSE, a BI Log Uniform distribution object (for model building).

- When sample=TRUE, a JAX array of samples drawn from the Log Uniform distribution (for direct sampling).

- When create_obj=TRUE, the raw BI distribution object (for advanced use cases).

See Also

This is a wrapper of https://num.pyro.ai/en/stable/distributions.html#loguniform

Examples

```
library(BayesianInference)
m=importBI(platform='cpu')
bi.dist.log_uniform(1,2, sample = TRUE)
```

---

bi.dist.logistic *Samples from a Logistic distribution.*

Description

The Logistic distribution is a continuous probability distribution defined by two parameters: location and scale. It is often used to model growth processes and is closely related to the normal distribution. Its CDF is the logistic (sigmoid) function, which makes it appealing in modeling probabilities, logistic regression, and various growth models. It resembles the normal distribution in shape (bell-shaped, symmetric) but has **heavier tails** (i.e. more probability in the extremes) and simpler closed-form expressions for the CDF.

Usage

```
bi.dist.logistic(
  loc = 0,
  scale = 1,
  validate_args = py_none(),
  name = "x",
  obs = py_none(),
  mask = py_none(),
  sample = FALSE,
  seed = py_none(),
  shape = c(),
  event = 0,
  create_obj = FALSE,
  to_jax = TRUE
)
```

Arguments

loc	Numeric vector or single number. The location parameter, specifying the median of the distribution. Defaults to 0.0.
scale	Numeric vector or single number. The scale parameter, which determines the spread of the distribution. Must be positive. Defaults to 1.0.
validate_args	Logical: Whether to validate parameter values. Defaults to 'reticulate::py_none()'.
name	A character string representing the name of the random variable within a model. This is used to uniquely identify the variable. Defaults to 'x'.
obs	A numeric vector or array of observed values. If provided, the random variable is conditioned on these values. If 'NULL', the variable is treated as a latent (unobserved) variable. Defaults to 'NULL'.
mask	Logical vector. Optional boolean array to mask observations.
sample	A logical value that controls the function's behavior. If 'TRUE', the function will directly draw samples from the distribution. If 'FALSE', it will create a random variable within a model. Defaults to 'FALSE'.
seed	An integer used to set the random seed for reproducibility when 'sample = TRUE'. This argument has no effect when 'sample = FALSE', as randomness is handled by the model's inference engine. Defaults to 0.
shape	Numeric vector. A multi-purpose argument for shaping. When 'sample=False' (model building), this is used with '.expand(shape)' to set the distribution's batch shape. When 'sample=True' (direct sampling), this is used as 'sample_shape' to draw a raw JAX array of the given shape.
event	Integer. The number of batch dimensions to reinterpret as event dimensions (used in model building).
create_obj	Logical. If True, returns the raw BI distribution object instead of creating a sample site. This is essential for building complex distributions like 'MixtureSameFamily'.
to_jax	Boolean. Indicates whether to return a JAX array or not.

Value

- When `sample=FALSE`, a BI Logistic distribution object (for model building).

- When `sample=TRUE`, a JAX array of samples drawn from the Logistic distribution (for direct sampling).

- When `create_obj=TRUE`, the raw BI distribution object (for advanced use cases).

Examples

```
library(BayesianInference)
m <- importBI(platform = "cpu")
bi.dist.logistic(sample = TRUE)
```

`bi.dist.low_rank_multivariate_normal`
Low Rank Multivariate Normal Distribution

Description

The *Low-Rank Multivariate Normal* (LRMVN) distribution is a parameterizaton of the multivariate normal distribution where the covariance matrix is expressed as a low-rank plus diagonal decomposition:

$$\Sigma = FF^\top + D$$

where F is a low-rank matrix (capturing correlations) and D is a diagonal matrix (capturing independent noise). This representation is often used in probabilistic modeling and variational inference to efficiently handle high-dimensional Gaussian distributions with structured covariance.

Usage

```
bi.dist.low_rank_multivariate_normal(
  loc,
  cov_factor,
  cov_diag,
  validate_args = py_none(),
  name = "x",
  obs = py_none(),
  mask = py_none(),
  sample = FALSE,
  seed = py_none(),
  shape = c(),
  event = 0,
  create_obj = FALSE,
  to_jax = TRUE
)
```

Arguments

`loc`	A numeric vector representing the mean vector.
`cov_factor`	A numeric vector or matrix used to construct the covariance matrix.
`cov_diag`	A numeric vector representing the diagonal elements of the covariance matrix.
`validate_args`	Logical: Whether to validate parameter values. Defaults to 'reticulate::py_none()'.
`name`	A character string representing the name of the random variable within a model. This is used to uniquely identify the variable. Defaults to 'x'.
`obs`	A numeric vector or array of observed values. If provided, the random variable is conditioned on these values. If 'NULL', the variable is treated as a latent (unobserved) variable. Defaults to 'NULL'.
`mask`	Logical vector. Optional boolean array to mask observations.

sample	A logical value that controls the function's behavior. If 'TRUE', the function will directly draw samples from the distribution. If 'FALSE', it will create a random variable within a model. Defaults to 'FALSE'.
seed	An integer used to set the random seed for reproducibility when 'sample = TRUE'. This argument has no effect when 'sample = FALSE', as randomness is handled by the model's inference engine. Defaults to 0.
shape	Numeric vector. A multi-purpose argument for shaping. When 'sample=False' (model building), this is used with '.expand(shape)' to set the distribution's batch shape. When 'sample=True' (direct sampling), this is used as 'sample_shape' to draw a raw JAX array of the given shape.
event	Integer. The number of batch dimensions to reinterpret as event dimensions (used in model building).
create_obj	Logical. If True, returns the raw BI distribution object instead of creating a sample site. This is essential for building complex distributions like 'MixtureSameFamily'.
to_jax	Boolean. Indicates whether to return a JAX array or not.

Value

- When `sample=FALSE`, a BI Low Rank Multivariate Normal distribution object (for model building).

- When `sample=TRUE`, a JAX array of samples drawn from the Low Rank Multivariate Normal distribution (for direct sampling).

- When `create_obj=TRUE`, the raw BI distribution object (for advanced use cases).

See Also

This is a wrapper of https://num.pyro.ai/en/stable/distributions.html#lowrankmultivariatenormal

Examples

```
library(BayesianInference)
m=importBI(platform='cpu')
event_size = 10
rank = 5
bi.dist.low_rank_multivariate_normal(
  loc = bi.dist.normal(0,1,shape = c(event_size), sample = TRUE)*2,
  cov_factor = bi.dist.normal(0,1,shape = c(event_size, rank), sample = TRUE),
  cov_diag = bi.dist.normal(10,0.5,shape = c(event_size), sample = TRUE),
  sample = TRUE)
```

`bi.dist.lower_truncated_power_law`

Lower Truncated Power Law Distribution

Description

The *Lower-Truncated Power-Law* distribution (also known as the *Pareto Type I* or *power-law with a lower bound*) models quantities that follow a heavy-tailed power-law behavior but are bounded below by a minimum value

$$x_{\min}$$

. It is commonly used to describe phenomena such as wealth distributions, city sizes, and biological scaling laws.

Usage

```
bi.dist.lower_truncated_power_law(
  alpha,
  low,
  validate_args = py_none(),
  name = "x",
  obs = py_none(),
  mask = py_none(),
  sample = FALSE,
  seed = py_none(),
  shape = c(),
  event = 0,
  create_obj = FALSE,
  to_jax = TRUE
)
```

Arguments

`alpha`	A numeric vector: index of the power law distribution. Must be less than -1.
`low`	A numeric vector: lower bound of the distribution. Must be greater than 0.
`validate_args`	Logical: Whether to validate parameter values. Defaults to ‘reticulate::py_none()‘.
`name`	A character string representing the name of the random variable within a model. This is used to uniquely identify the variable. Defaults to ’x’.
`obs`	A numeric vector or array of observed values. If provided, the random variable is conditioned on these values. If ‘NULL‘, the variable is treated as a latent (unobserved) variable. Defaults to ‘NULL‘.
`mask`	A logical vector: Optional boolean array to mask observations.
`sample`	A logical value that controls the function’s behavior. If ‘TRUE‘, the function will directly draw samples from the distribution. If ‘FALSE‘, it will create a random variable within a model. Defaults to ‘FALSE‘.

seed	An integer used to set the random seed for reproducibility when 'sample = TRUE'. This argument has no effect when 'sample = FALSE', as randomness is handled by the model's inference engine. Defaults to 0.
shape	A numeric vector: A multi-purpose argument for shaping. When 'sample=False' (model building), this is used with '.expand(shape)' to set the distribution's batch shape. When 'sample=True' (direct sampling), this is used as 'sample_shape' to draw a raw JAX array of the given shape.
event	Integer: The number of batch dimensions to reinterpret as event dimensions (used in model building).
create_obj	Logical: If 'TRUE', returns the raw BI distribution object instead of creating a sample site. This is essential for building complex distributions like 'MixtureSameFamily'.
to_jax	Boolean. Indicates whether to return a JAX array or not.

Value

- When `sample=FALSE`, a BI Lower Truncated Power Law distribution object (for model building).

- When `sample=TRUE`, a JAX array of samples drawn from the Lower Truncated Power Law distribution (for direct sampling).

- When `create_obj=TRUE`, the raw BI distribution object (for advanced use cases).

See Also

This is a wrapper of https://num.pyro.ai/en/stable/distributions.html#lowertruncatedpowerlaw

Examples

```
library(BayesianInference)
m <- importBI(platform = "cpu")
bi.dist.lower_truncated_power_law(alpha = c(-2, 2), low = c(1, 0.5), sample = TRUE)
```

---

bi.dist.matrix_normal *Matrix Normal Distribution*

Description

Samples from a Matrix Normal distribution, which is a multivariate normal distribution over matrices. The distribution is characterized by a location matrix and two lower triangular matrices that define the correlation structure. The distribution is related to the multivariate normal distribution in the following way. If

$$X \sim MN(loc, U, V) \implies vec(X) \sim MVN(vec(loc), kron(V, U))$$

.

Usage

```
bi.dist.matrix_normal(
  loc,
  scale_tril_row,
  scale_tril_column,
  validate_args = py_none(),
  name = "x",
  obs = py_none(),
  mask = py_none(),
  sample = FALSE,
  seed = py_none(),
  shape = c(),
  event = 0,
  create_obj = FALSE,
  to_jax = TRUE
)
```

Arguments

loc	A numeric vector, matrix, or array representing the location of the distribution.
scale_tril_row	A numeric vector, matrix, or array representing the lower cholesky of rows correlation matrix.
scale_tril_column	A numeric vector, matrix, or array representing the lower cholesky of columns correlation matrix.
validate_args	Logical: Whether to validate parameter values. Defaults to 'reticulate::py_none()'.
name	A character string representing the name of the random variable within a model. This is used to uniquely identify the variable. Defaults to 'x'.
obs	A numeric vector or array of observed values. If provided, the random variable is conditioned on these values. If 'NULL', the variable is treated as a latent (unobserved) variable. Defaults to 'NULL'.
mask	A logical vector, matrix, or array (.BI_envjnparray) to mask observations.
sample	A logical value that controls the function's behavior. If 'TRUE', the function will directly draw samples from the distribution. If 'FALSE', it will create a random variable within a model. Defaults to 'FALSE'.
seed	An integer used to set the random seed for reproducibility when 'sample = TRUE'. This argument has no effect when 'sample = FALSE', as randomness is handled by the model's inference engine. Defaults to 0.
shape	A numeric vector specifying the shape of the distribution. Must be a vector.
event	An integer representing the number of batch dimensions to reinterpret as event dimensions.
create_obj	A logical value. If 'TRUE', returns the raw BI distribution object instead of creating a sample site.
to_jax	Boolean. Indicates whether to return a JAX array or not.

Value

- When sample=FALSE, a BI Matrix Normal distribution object (for model building).

- When sample=TRUE, a JAX array of samples drawn from the Matrix Normal distribution (for direct sampling).

- When create_obj=TRUE, the raw BI distribution object (for advanced use cases).

See Also

This is a wrapper of https://num.pyro.ai/en/stable/distributions.html#matrixnormal_lowercase

Examples

```
library(BayesianInference)
m <- importBI(platform = "cpu")
n_rows <- 3
n_cols <- 4
loc <- matrix(rep(0, n_rows * n_cols), nrow = n_rows, ncol = n_cols, byrow = TRUE)

U_row_cov <-
  matrix(c(1.0, 0.5, 0.2, 0.5, 1.0, 0.3, 0.2, 0.3, 1.0),
    nrow = n_rows, ncol = n_rows, byrow = TRUE
  )
scale_tril_row <- chol(U_row_cov)

V_col_cov <- matrix(
  c(
    2.0, -0.8, 0.1, 0.4, -0.8, 2.0, 0.2, -0.2, 0.1,
    0.2, 2.0, 0.0, 0.4, -0.2, 0.0, 2.0
  ),
  nrow = n_cols, ncol = n_cols, byrow = TRUE
)
scale_tril_column <- chol(V_col_cov)


bi.dist.matrix_normal(
  loc = loc,
  scale_tril_row = scale_tril_row,
  scale_tril_column = scale_tril_column,
  sample = TRUE
)
```

---

bi.dist.mixture *A marginalized finite mixture of component distributions.*

Description

This distribution represents a mixture of component distributions, where the mixing weights are determined by a Categorical distribution. The resulting distribution can be either a MixtureGeneral (when component distributions are a list) or a MixtureSameFamily (when component distributions are a single distribution).

Usage

```
bi.dist.mixture(
  mixing_distribution,
  component_distributions,
  validate_args = py_none(),
  name = "x",
  obs = py_none(),
  mask = py_none(),
  sample = FALSE,
  seed = py_none(),
  shape = c(),
  event = 0,
  create_obj = FALSE,
  to_jax = TRUE
)
```

Arguments

mixing_distribution	A 'Categorical' distribution specifying the weights for each mixture component. The size of this distribution specifies the number of components in the mixture.
component_distributions	A list of distributions representing the components of the mixture.
validate_args	Logical: Whether to validate parameter values. Defaults to 'reticulate::py_none()'.
name	A character string representing the name of the random variable within a model. This is used to uniquely identify the variable. Defaults to 'x'.
obs	A numeric vector or array of observed values. If provided, the random variable is conditioned on these values. If 'NULL', the variable is treated as a latent (unobserved) variable. Defaults to 'NULL'.
mask	A logical vector used to mask observations.
sample	A logical value that controls the function's behavior. If 'TRUE', the function will directly draw samples from the distribution. If 'FALSE', it will create a random variable within a model. Defaults to 'FALSE'.
seed	An integer used to set the random seed for reproducibility when 'sample = TRUE'. This argument has no effect when 'sample = FALSE', as randomness is handled by the model's inference engine. Defaults to 0.
shape	A numeric vector specifying the shape of the distribution.
event	An integer representing the number of batch dimensions to reinterpret as event dimensions.

create_obj	A logical value. If 'TRUE', returns the raw BI distribution object.
to_jax	Boolean. Indicates whether to return a JAX array or not.

Value

When sample=FALSE: A BI Mixture distribution object (for model building). When sample=TRUE: A JAX array of samples drawn from the Mixture distribution (for direct sampling). When create_obj=TRUE: The raw BI distribution object (for advanced use cases).

See Also

- When sample=FALSE, a BI marginalized finite mixture distribution object (for model building).

- When sample=TRUE, a JAX array of samples drawn from the marginalized finite mixture distribution (for direct sampling).

- When create_obj=TRUE, the raw BI distribution object (for advanced use cases).

Examples

```
library(BayesianInference)
m=importBI(platform='cpu')
bi.dist.mixture(
  mixing_distribution = bi.dist.categorical(probs = c(0.3, 0, 7),create_obj = TRUE),
  component_distributions = c(
  bi.dist.normal(0,1,create_obj = TRUE),
  bi.dist.normal(0,1,create_obj = TRUE),
  bi.dist.normal(0,1,create_obj = TRUE)
  ),
  sample = TRUE
)
```

---

bi.dist.mixture_general

A finite mixture of component distributions from different families.

Description

A mixture distribution is a probability distribution constructed by selecting one of several component distributions according to specified weights, and then drawing a sample from the chosen component. It allows modelling of heterogeneous populations and multimodal data.

Usage

```
bi.dist.mixture_general(
  mixing_distribution,
  component_distributions,
  support = py_none(),
  validate_args = py_none(),
```

```
  name = "x",
  obs = py_none(),
  mask = py_none(),
  sample = FALSE,
  seed = py_none(),
  shape = c(),
  event = 0,
  create_obj = FALSE,
  to_jax = TRUE
)
```

Arguments

mixing_distribution	A 'Categorical' distribution specifying the weights for each mixture component. The size of this distribution specifies the number of components in the mixture.
component_distributions	A list of distributions representing the components of the mixture.
support	A constraint object specifying the support of the mixture distribution. If not provided, the support will be inferred from the component distributions.
validate_args	Logical: Whether to validate parameter values. Defaults to 'reticulate::py_none()'.
name	A character string representing the name of the random variable within a model. This is used to uniquely identify the variable. Defaults to 'x'.
obs	A numeric vector or array of observed values. If provided, the random variable is conditioned on these values. If 'NULL', the variable is treated as a latent (unobserved) variable. Defaults to 'NULL'.
mask	Logical vector. Optional boolean array to mask observations.
sample	A logical value that controls the function's behavior. If 'TRUE', the function will directly draw samples from the distribution. If 'FALSE', it will create a random variable within a model. Defaults to 'FALSE'.
seed	An integer used to set the random seed for reproducibility when 'sample = TRUE'. This argument has no effect when 'sample = FALSE', as randomness is handled by the model's inference engine. Defaults to 0.
shape	Numeric vector. A multi-purpose argument for shaping. When 'sample=False' (model building), this is used with '.expand(shape)' to set the distribution's batch shape. When 'sample=True' (direct sampling), this is used as 'sample_shape' to draw a raw JAX array of the given shape.
event	Integer. The number of batch dimensions to reinterpret as event dimensions (used in model building).
create_obj	Logical. If True, returns the raw BI distribution object instead of creating a sample site. This is essential for building complex distributions like 'MixtureSameFamily'.
to_jax	Boolean. Indicates whether to return a JAX array or not.

Value

- When sample=FALSE, a BI MixtureGeneral distribution object (for model building).

- When sample=TRUE, a JAX array of samples drawn from the MixtureGeneral distribution (for direct sampling).

- When create_obj=TRUE, the raw BI distribution object (for advanced use cases).

Examples

```
library(BayesianInference)
m=importBI(platform='cpu')
bi.dist.mixture_general(
mixing_distribution = bi.dist.categorical(probs = c(0.3, 0, 7),create_obj = TRUE),
component_distributions = c(
bi.dist.normal(0,1,create_obj = TRUE),
bi.dist.normal(0,1,create_obj = TRUE),
bi.dist.normal(0,1,create_obj = TRUE)),
sample = TRUE)
```

---

bi.dist.mixture_same_family

A finite mixture of component distributions from the same family.

Description

A *Mixture (Same-Family)* distribution is a finite mixture in which **all components come from the same parametric family** (for example, all Normal distributions but with different parameters), and are combined via mixing weights.

Usage

```
bi.dist.mixture_same_family(
  mixing_distribution,
  component_distribution,
  validate_args = py_none(),
  name = "x",
  obs = py_none(),
  mask = py_none(),
  sample = FALSE,
  seed = py_none(),
  shape = c(),
  event = 0,
  create_obj = FALSE,
  to_jax = TRUE
)
```

Arguments

mixing_distribution	A distribution specifying the weights for each mixture component. The size of this distribution specifies the number of components in the mixture.
component_distribution	A list of distributions representing the components of the mixture.
validate_args	Logical: Whether to validate parameter values. Defaults to 'reticulate::py_none()'.
name	A character string representing the name of the random variable within a model. This is used to uniquely identify the variable. Defaults to 'x'.
obs	A numeric vector or array of observed values. If provided, the random variable is conditioned on these values. If 'NULL', the variable is treated as a latent (unobserved) variable. Defaults to 'NULL'.
mask	A logical vector, matrix, or array to mask observations.
sample	A logical value that controls the function's behavior. If 'TRUE', the function will directly draw samples from the distribution. If 'FALSE', it will create a random variable within a model. Defaults to 'FALSE'.
seed	An integer used to set the random seed for reproducibility when 'sample = TRUE'. This argument has no effect when 'sample = FALSE', as randomness is handled by the model's inference engine. Defaults to 0.
shape	A numeric vector specifying the shape of the distribution.
event	Integer representing the number of batch dimensions to reinterpret as event dimensions (used in model building).
create_obj	Logical; If TRUE, returns the raw BI distribution object instead of creating a sample site.
to_jax	Boolean. Indicates whether to return a JAX array or not.

Value

- When `sample=FALSE`, a BI MixtureSameFamily distribution object (for model building).

- When `sample=TRUE`, a JAX array of samples drawn from the MixtureSameFamily distribution (for direct sampling).

- When `create_obj=TRUE`, the raw BI distribution object (for advanced use cases).

See Also

This is a wrapper of https://num.pyro.ai/en/stable/distributions.html#mixture-same-family

Examples

```
library(BayesianInference)
m <- importBI(platform = "cpu")
bi.dist.mixture_same_family(
  mixing_distribution = bi.dist.categorical(probs = c(0.3, 0.7), create_obj = TRUE),
  component_distribution = bi.dist.normal(0, 1, shape = c(2), create_obj = TRUE),
  sample = TRUE
)
```

`bi.dist.multinomial` *Multinomial distribution.*

Description

Samples from a Multinomial distribution, which models the probability of different outcomes in a sequence of independent trials, each with a fixed number of trials and a fixed set of possible outcomes. It generalizes the binomial distribution to multiple categories.

Usage

```
bi.dist.multinomial(
  total_count = 1,
  probs = py_none(),
  logits = py_none(),
  total_count_max = py_none(),
  validate_args = py_none(),
  name = "x",
  obs = py_none(),
  mask = py_none(),
  sample = FALSE,
  seed = py_none(),
  shape = c(),
  event = 0,
  create_obj = FALSE,
  to_jax = TRUE
)
```

Arguments

`total_count`	An integer or numeric vector representing the number of trials.
`probs`	A numeric vector representing event probabilities. Must sum to 1.
`logits`	A numeric vector representing event log probabilities.
`total_count_max`	(int, optional): An optional integer providing an upper bound on 'total_count'. This is used for performance optimization with 'lax.scan' when 'total_count' is a dynamic JAX tracer, helping to avoid recompilation.
`validate_args`	Logical: Whether to validate parameter values. Defaults to 'reticulate::py_none()'.
`name`	A character string representing the name of the random variable within a model. This is used to uniquely identify the variable. Defaults to 'x'.
`obs`	A numeric vector or array of observed values. If provided, the random variable is conditioned on these values. If 'NULL', the variable is treated as a latent (unobserved) variable. Defaults to 'NULL'.
`mask`	A logical vector, optional, to mask observations.

sample	A logical value that controls the function's behavior. If 'TRUE', the function will directly draw samples from the distribution. If 'FALSE', it will create a random variable within a model. Defaults to 'FALSE'.
seed	An integer used to set the random seed for reproducibility when 'sample = TRUE'. This argument has no effect when 'sample = FALSE', as randomness is handled by the model's inference engine. Defaults to 0.
shape	A numeric vector used for shaping. When sample=FALSE (model building), this is used with '.expand(shape)' to set the distribution's batch shape. When sample=TRUE (direct sampling), this is used as 'sample_shape' to draw a raw JAX array of the given shape.
event	An integer representing the number of batch dimensions to reinterpret as event dimensions (used in model building).
create_obj	A logical value, optional. If 'TRUE', returns the raw BI distribution object instead of creating a sample site.
to_jax	Boolean. Indicates whether to return a JAX array or not.

Value

- When sample=FALSE, a BI Multinomial distribution object (for model building).

- When sample=TRUE, a JAX array of samples drawn from the Multinomial distribution (for direct sampling).

- When create_obj=TRUE, the raw BI distribution object (for advanced use cases).

Examples

```
library(BayesianInference)
m=importBI(platform='cpu')
bi.dist.multinomial(probs = c(0.5,0.1), sample = TRUE)
```

---

bi.dist.multinomial_logits

Multinomial logit

Description

A *multinomial logits* distribution refers to a categorical (or more generally multinomial) distribution over K classes whose probabilities are given via the softmax of a vector of logits. That is, given a vector of real-valued logits $\ell = (\ell_1, \ldots, \ell_K)$, the class probabilities are:

$$p_k = \frac{\exp(\ell_k)}{\sum_{j=1}^{K} \exp(\ell_j)}.$$

Then a single draw from the distribution yields one of the K classes (or for a multinomial count version, counts over the classes) with those probabilities.#' @export

Usage

```
bi.dist.multinomial_logits(
  logits,
  total_count = 1,
  total_count_max = py_none(),
  validate_args = py_none(),
  name = "x",
  obs = py_none(),
  mask = py_none(),
  sample = FALSE,
  seed = py_none(),
  shape = c(),
  event = 0,
  create_obj = FALSE,
  to_jax = TRUE
)
```

Arguments

logits	A numeric vector, matrix, or array representing the logits for each outcome.
total_count	A numeric vector, matrix, or array representing the total number of trials.
total_count_max	(int, optional): An optional integer providing an upper bound on 'total_count'. This is used for performance optimization with 'lax.scan' when 'total_count' is a dynamic JAX tracer, helping to avoid recompilation.
validate_args	Logical: Whether to validate parameter values. Defaults to 'reticulate::py_none()'.
name	A character string representing the name of the random variable within a model. This is used to uniquely identify the variable. Defaults to 'x'.
obs	A numeric vector or array of observed values. If provided, the random variable is conditioned on these values. If 'NULL', the variable is treated as a latent (unobserved) variable. Defaults to 'NULL'.
mask	A logical vector, matrix, or array to mask observations.
sample	A logical value that controls the function's behavior. If 'TRUE', the function will directly draw samples from the distribution. If 'FALSE', it will create a random variable within a model. Defaults to 'FALSE'.
seed	An integer used to set the random seed for reproducibility when 'sample = TRUE'. This argument has no effect when 'sample = FALSE', as randomness is handled by the model's inference engine. Defaults to 0.
shape	A numeric vector specifying the shape of the distribution. Use a vector (e.g., 'c(10)') to define the shape.
event	Integer specifying the number of batch dimensions to reinterpret as event dimensions (used in model building).
create_obj	Logical; If TRUE, returns the raw BI distribution object instead of creating a sample site.
to_jax	Boolean. Indicates whether to return a JAX array or not.

Value

- When `sample=FALSE`, a BI MultinomialLogits distribution object (for model building).

- When `sample=TRUE`, a JAX array of samples drawn from the MultinomialLogits distribution (for direct sampling).

- When `create_obj=TRUE`, the raw BI distribution object (for advanced use cases).

See Also

This is a wrapper of https://num.pyro.ai/en/stable/distributions.html#multinomiallogits

Examples

```
library(BayesianInference)
m=importBI(platform='cpu')
bi.dist.multinomial_logits(logits =  c(0.2, 0.3, 0.5), total_count = 10, sample = TRUE)
```

---

bi.dist.multinomial_probs

Samples from a Multinomial distribution.

Description

The Multinomial distribution models the number of times each of several discrete outcomes occurs in a fixed number of trials. Each trial independently results in one of several outcomes, and each outcome has a probability of occurring.

Usage

```
bi.dist.multinomial_probs(
  probs,
  total_count = 1,
  total_count_max = py_none(),
  validate_args = py_none(),
  name = "x",
  obs = py_none(),
  mask = py_none(),
  sample = FALSE,
  seed = py_none(),
  shape = c(),
  event = 0,
  create_obj = FALSE,
  to_jax = TRUE
)
```

Arguments

probs	A numeric vector of probabilities for each outcome. Must sum to 1.
total_count	The number of trials.
total_count_max	(int, optional): An optional integer providing an upper bound on 'total_count'. This is used for performance optimization with 'lax.scan' when 'total_count' is a dynamic JAX tracer, helping to avoid recompilation.
validate_args	Logical: Whether to validate parameter values. Defaults to 'reticulate::py_none()'.
name	A character string representing the name of the random variable within a model. This is used to uniquely identify the variable. Defaults to 'x'.
obs	A numeric vector or array of observed values. If provided, the random variable is conditioned on these values. If 'NULL', the variable is treated as a latent (unobserved) variable. Defaults to 'NULL'.
mask	An optional boolean vector to mask observations.
sample	A logical value that controls the function's behavior. If 'TRUE', the function will directly draw samples from the distribution. If 'FALSE', it will create a random variable within a model. Defaults to 'FALSE'.
seed	An integer used to set the random seed for reproducibility when 'sample = TRUE'. This argument has no effect when 'sample = FALSE', as randomness is handled by the model's inference engine. Defaults to 0.
shape	A numeric vector. When 'sample=False' (model building), this is used with '.expand(shape)' to set the distribution's batch shape. When 'sample=True' (direct sampling), this is used as 'sample_shape' to draw a raw JAX array of the given shape.
event	The number of batch dimensions to reinterpret as event dimensions (used in model building).
create_obj	Logical. If 'TRUE', returns the raw BI distribution object instead of creating a sample site. This is essential for building complex distributions like 'MixtureSameFamily'.
to_jax	Boolean. Indicates whether to return a JAX array or not.

Value

- When `sample=FALSE`, a BI MultinomialProbs distribution object (for model building).

- When `sample=TRUE`, a JAX array of samples drawn from the MultinomialProbs distribution (for direct sampling).

- When `create_obj=TRUE`, the raw BI distribution object (for advanced use cases).

See Also

https://num.pyro.ai/en/stable/distributions.html#multinomialprobs

Examples

```
library(BayesianInference)
m=importBI(platform='cpu')
bi.dist.multinomial_probs(probs =  c(0.2, 0.3, 0.5), total_count = c(10,10), sample = TRUE)
```

`bi.dist.multivariate_normal`

Samples from a Multivariate Normal distribution.

Description

The Multivariate Normal distribution, also known as the Gaussian distribution in multiple dimensions, is a probability distribution that arises frequently in statistics and machine learning. It is defined by its mean vector and covariance matrix, which describe the central tendency and spread of the distribution, respectively.

Usage

```
bi.dist.multivariate_normal(
  loc = 0,
  covariance_matrix = py_none(),
  precision_matrix = py_none(),
  scale_tril = py_none(),
  validate_args = py_none(),
  name = "x",
  obs = py_none(),
  mask = py_none(),
  sample = FALSE,
  seed = py_none(),
  shape = c(),
  event = 0,
  create_obj = FALSE,
  to_jax = TRUE
)
```

Arguments

`loc`	A numeric vector representing the mean vector of the distribution.
`covariance_matrix`	A numeric vector, matrix, or array representing the covariance matrix of the distribution. Must be positive definite.
`precision_matrix`	A numeric vector, matrix, or array representing the precision matrix (inverse of the covariance matrix) of the distribution. Must be positive definite.
`scale_tril`	A numeric vector, matrix, or array representing the lower triangular Cholesky decomposition of the covariance matrix.

validate_args	Logical: Whether to validate parameter values. Defaults to 'reticulate::py_none()'.
name	A character string representing the name of the random variable within a model. This is used to uniquely identify the variable. Defaults to 'x'.
obs	A numeric vector or array of observed values. If provided, the random variable is conditioned on these values. If 'NULL', the variable is treated as a latent (unobserved) variable. Defaults to 'NULL'.
mask	A logical vector representing an optional boolean array to mask observations.
sample	A logical value that controls the function's behavior. If 'TRUE', the function will directly draw samples from the distribution. If 'FALSE', it will create a random variable within a model. Defaults to 'FALSE'.
seed	An integer used to set the random seed for reproducibility when 'sample = TRUE'. This argument has no effect when 'sample = FALSE', as randomness is handled by the model's inference engine. Defaults to 0.
shape	A numeric vector representing the shape of the distribution.
event	Integer representing the number of batch dimensions to reinterpret as event dimensions (used in model building).
create_obj	Logical; If TRUE, returns the raw BI distribution object instead of creating a sample site.
to_jax	Boolean. Indicates whether to return a JAX array or not.

Value

- When `sample=FALSE`, a BI Multivariate Normal distribution object (for model building).

- When `sample=TRUE`, a JAX array of samples drawn from the Multivariate Normal distribution (for direct sampling).

- When `create_obj=TRUE`, the raw BI distribution object (for advanced use cases).

See Also

https://num.pyro.ai/en/stable/distributions.html#multivariate-normal

Examples

```
library(BayesianInference)
m=importBI(platform='cpu')
bi.dist.multivariate_normal(
loc =  c(1.0, 0.0, -2.0),
covariance_matrix = matrix(
c( 2.0,  0.7, -0.3, 0.7,  1.0,  0.5, -0.3,  0.5,  1.5),
nrow = 3, byrow = TRUE),
sample = TRUE)
```

`bi.dist.multivariate_student_t`
Multivariate Student's t Distribution

Description

The Multivariate Student's t distribution is a generalization of the Student's t distribution to multiple dimensions. It is a heavy-tailed distribution that is often used to model data that is not normally distributed. The PDF of the multivariate Student's t-distribution for a random vector

$$x \in R^d$$

is given by:

$$f(x) = \frac{\Gamma\left(\frac{\nu+d}{2}\right)}{\Gamma\left(\frac{\nu}{2}\right)\nu^{d/2}\pi^{d/2}|\Sigma|^{1/2}}\left(1+\frac{1}{\nu}(x-\mu)^T\Sigma^{-1}(x-\mu)\right)^{-(\nu+d)/2}$$

where: *

$$\Gamma(\cdot)$$

is the Gamma function. *

$$\mu$$

is the mean vector. *

$$\Sigma$$

is the scale (covariance) matrix. *

$$\nu$$

is the degrees of freedom. *

$$d$$

is the dimensionality of

$$\mathbf{x}$$

.

Usage

```
bi.dist.multivariate_student_t(
  df,
  loc = 0,
  scale_tril = py_none(),
  validate_args = py_none(),
  name = "x",
  obs = py_none(),
  mask = py_none(),
  sample = FALSE,
  seed = py_none(),
  shape = c(),
  event = 0,
  create_obj = FALSE,
  to_jax = TRUE
)
```

Arguments

df	A numeric vector representing degrees of freedom, must be positive.
loc	A numeric vector representing the location vector (mean) of the distribution.
scale_tril	A numeric matrix defining the scale (lower triangular matrix).
validate_args	Logical: Whether to validate parameter values. Defaults to 'reticulate::py_none()'.
name	A character string representing the name of the random variable within a model. This is used to uniquely identify the variable. Defaults to 'x'.
obs	A numeric vector or array of observed values. If provided, the random variable is conditioned on these values. If 'NULL', the variable is treated as a latent (unobserved) variable. Defaults to 'NULL'.
mask	A logical vector. Optional boolean array to mask observations.
sample	A logical value that controls the function's behavior. If 'TRUE', the function will directly draw samples from the distribution. If 'FALSE', it will create a random variable within a model. Defaults to 'FALSE'.
seed	An integer used to set the random seed for reproducibility when 'sample = TRUE'. This argument has no effect when 'sample = FALSE', as randomness is handled by the model's inference engine. Defaults to 0.
shape	A numeric vector. When 'sample=False' (model building), this is used with '.expand(shape)' to set the distribution's batch shape. When 'sample=True' (direct sampling), this is used as 'sample_shape' to draw a raw JAX array of the given shape.
event	An integer representing the number of batch dimensions to reinterpret as event dimensions (used in model building).
create_obj	A logical value. If 'TRUE', returns the raw BI distribution object instead of creating a sample site.
to_jax	Boolean. Indicates whether to return a JAX array or not.

Value

- When `sample=FALSE`, a BI Multivariate Student's t distribution object (for model building).

- When `sample=TRUE`, a JAX array of samples drawn from the Multivariate Student's t distribution (for direct sampling).

- When `create_obj=TRUE`, the raw BI distribution object (for advanced use cases).

See Also

https://num.pyro.ai/en/stable/distributions.html#multivariatestudentt

Examples

```
library(BayesianInference)
m=importBI(platform='cpu')
bi.dist.multivariate_student_t(
df = 2,
loc =  c(1.0, 0.0, -2.0),
```

```
scale_tril = chol(
matrix(c( 2.0,  0.7, -0.3, 0.7,  1.0,  0.5, -0.3,  0.5,  1.5),
nrow = 3, byrow = TRUE)),
sample = TRUE)
```

`bi.dist.negative_binomial_logits` *Samples from a Negative Binomial Logits distribution.*

Description

The Negative Binomial Logits distribution is a generalization of the Negative Binomial distribution where the parameter 'r' (number of successes) is expressed as a function of a logit parameter. This allows for more flexible modeling of count data.

Usage

```
bi.dist.negative_binomial_logits(
  total_count,
  logits,
  validate_args = py_none(),
  name = "x",
  obs = py_none(),
  mask = py_none(),
  sample = FALSE,
  seed = py_none(),
  shape = c(),
  event = 0,
  create_obj = FALSE,
  to_jax = TRUE
)
```

Arguments

`total_count`	A numeric vector, matrix, or array representing the parameter controlling the shape of the distribution. Represents the total number of trials.
`logits`	A numeric vector, matrix, or array representing the log-odds parameter. Related to the probability of success.
`validate_args`	Logical: Whether to validate parameter values. Defaults to 'reticulate::py_none()'.
`name`	A character string representing the name of the random variable within a model. This is used to uniquely identify the variable. Defaults to 'x'.
`obs`	A numeric vector or array of observed values. If provided, the random variable is conditioned on these values. If 'NULL', the variable is treated as a latent (unobserved) variable. Defaults to 'NULL'.
`mask`	A logical vector, matrix, or array. Optional boolean array to mask observations.

sample	A logical value that controls the function's behavior. If 'TRUE', the function will directly draw samples from the distribution. If 'FALSE', it will create a random variable within a model. Defaults to 'FALSE'.
seed	An integer used to set the random seed for reproducibility when 'sample = TRUE'. This argument has no effect when 'sample = FALSE', as randomness is handled by the model's inference engine. Defaults to 0.
shape	A numeric vector. A multi-purpose argument for shaping. When 'sample=False' (model building), this is used with '.expand(shape)' to set the distribution's batch shape. When 'sample=True' (direct sampling), this is used as 'sample_shape' to draw a raw JAX array of the given shape.
event	An integer representing the number of batch dimensions to reinterpret as event dimensions (used in model building).
create_obj	A logical value. If 'TRUE', returns the raw BI distribution object instead of creating a sample site. This is essential for building complex distributions like 'MixtureSameFamily'.
to_jax	Boolean. Indicates whether to return a JAX array or not.

Value

- When `sample=FALSE`, a BI Negative Binomial Logits distribution object (for model building).

- When `sample=TRUE`, a JAX array of samples drawn from the Negative Binomial Logits distribution (for direct sampling).

- When `create_obj=TRUE`, the raw BI distribution object (for advanced use cases).

See Also

this is a wrapper from https://num.pyro.ai/en/stable/distributions.html#negativebinomiallogits

Examples

```
library(BayesianInference)
m=importBI(platform='cpu')
bi.dist.negative_binomial_logits(logits =  c(0.2, 0.3, 0.5), total_count = 10, sample = TRUE)
```

---

bi.dist.negative_binomial_probs

Sample from a Negative Binomial distribution with probabilities.

Description

Negative Binomial distribution models the number of failures before the first success in a sequence of independent Bernoulli trials. It is characterized by two parameters: 'concentration' (r) and 'rate' (p). In this implementation, the 'concentration' parameter is derived from 'total_count' and the 'rate' parameter is derived from 'probs'.

Usage

```
bi.dist.negative_binomial_probs(
  total_count,
  probs,
  validate_args = py_none(),
  name = "x",
  obs = py_none(),
  mask = py_none(),
  sample = FALSE,
  seed = py_none(),
  shape = c(),
  event = 0,
  create_obj = FALSE,
  to_jax = TRUE
)
```

Arguments

total_count	A numeric vector, matrix, or array representing the parameter controlling the shape of the distribution. Represents the total number of trials.
probs	A numeric vector representing event probabilities. Must sum to 1.
validate_args	Logical: Whether to validate parameter values. Defaults to 'reticulate::py_none()'.
name	A character string representing the name of the random variable within a model. This is used to uniquely identify the variable. Defaults to 'x'.
obs	A numeric vector or array of observed values. If provided, the random variable is conditioned on these values. If 'NULL', the variable is treated as a latent (unobserved) variable. Defaults to 'NULL'.
mask	A logical vector or array. Optional boolean array to mask observations.
sample	A logical value that controls the function's behavior. If 'TRUE', the function will directly draw samples from the distribution. If 'FALSE', it will create a random variable within a model. Defaults to 'FALSE'.
seed	An integer used to set the random seed for reproducibility when 'sample = TRUE'. This argument has no effect when 'sample = FALSE', as randomness is handled by the model's inference engine. Defaults to 0.
shape	A numeric vector. When 'sample=False' (model building), this is used with '.expand(shape)' to set the distribution's batch shape. When 'sample=True' (direct sampling), this is used as 'sample_shape' to draw a raw JAX array of the given shape.
event	An integer representing the number of batch dimensions to reinterpret as event dimensions (used in model building).
create_obj	A logical value. If 'TRUE', returns the raw BI distribution object instead of creating a sample site. This is essential for building complex distributions like 'MixtureSameFamily'.
to_jax	Boolean. Indicates whether to return a JAX array or not.

Value

- When sample=FALSE, a BI Negative Binomial distribution object (for model building).

- When sample=TRUE, a JAX array of samples drawn from the Negative Binomial distribution (for direct sampling).

- When create_obj=TRUE, the raw BI distribution object (for advanced use cases).

See Also

https://num.pyro.ai/en/stable/distributions.html#negativebinomialprobs

Examples

```
library(BayesianInference)
m=importBI(platform='cpu')
bi.dist.negative_binomial_probs(probs =  c(0.2, 0.3, 0.5), total_count = 10, sample = TRUE)
```

`bi.dist.negative_binomial2`

Samples from a Negative Binomial distribution.

Description

The NB2 parameterisation of the negative-binomial distribution is a count distribution used for modelling over-dispersed count data (variance > mean). It is defined such that the variance grows **quadratically** in the mean:

$$Var(Y) = \mu + \alpha, \mu^2,$$

where

$$\mu = E[Y])and(\alpha > 0)$$

is the dispersion (heterogeneity) parameter. Because of this quadratic variance growth, it is called the NB2 family.

$$P(k) = \frac{\Gamma(k+\alpha)}{\Gamma(k+1)\Gamma(\alpha)} \left(\frac{\beta}{\alpha+\beta}\right)^k \left(1-\frac{\beta}{\alpha+\beta}\right)^k$$

Usage

```
bi.dist.negative_binomial2(
  total_count,
  probs,
  logits = py_none(),
  validate_args = py_none(),
  name = "x",
  obs = py_none(),
  mask = py_none(),
```

```
  sample = FALSE,
  seed = py_none(),
  shape = c(),
  event = 0,
  create_obj = FALSE,
  to_jax = TRUE
)
```

Arguments

total_count	(int): The number of trials *n*.
probs	A numeric vector, matrix, or array representing the probability of success for each Bernoulli trial. Must be between 0 and 1.
logits	A numeric vector, matrix, or array representing the log-odds of success for each trial.
validate_args	Logical: Whether to validate parameter values. Defaults to 'reticulate::py_none()'.
name	A character string representing the name of the random variable within a model. This is used to uniquely identify the variable. Defaults to 'x'.
obs	A numeric vector or array of observed values. If provided, the random variable is conditioned on these values. If 'NULL', the variable is treated as a latent (unobserved) variable. Defaults to 'NULL'.
mask	An optional logical vector to mask observations.
sample	A logical value that controls the function's behavior. If 'TRUE', the function will directly draw samples from the distribution. If 'FALSE', it will create a random variable within a model. Defaults to 'FALSE'.
seed	An integer used to set the random seed for reproducibility when 'sample = TRUE'. This argument has no effect when 'sample = FALSE', as randomness is handled by the model's inference engine. Defaults to 0.
shape	A numeric vector. Used with '.expand(shape)' when 'sample=False' (model building) to set the distribution's batch shape. When 'sample=True' (direct sampling), used as 'sample_shape' to draw a raw JAX array of the given shape.
event	An integer representing the number of batch dimensions to reinterpret as event dimensions (used in model building).
create_obj	A logical value. If 'TRUE', returns the raw BI distribution object instead of creating a sample site.
to_jax	Boolean. Indicates whether to return a JAX array or not.

Value

- When `sample=FALSE`, a BI Negative Binomial distribution object (for model building).

- When `sample=TRUE`, a JAX array of samples drawn from the Negative Binomial distribution (for direct sampling).

- When `create_obj=TRUE`, the raw BI distribution object (for advanced use cases).

See Also

https://num.pyro.ai/en/stable/distributions.html#negativebinomial2

Examples

```
library(BayesianInference)
m=importBI(platform='cpu')
bi.dist.negative_binomial2(total_count = 100, probs = 0.5, sample = TRUE)
```

`bi.dist.normal` *Samples from a Normal (Gaussian) distribution.*

Description

The Normal distribution is characterized by its mean (loc) and standard deviation (scale). It's a continuous probability distribution that arises frequently in statistics and probability theory.

$$f(x \mid \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} \exp! \left(-\frac{(x-\mu)^2}{2\sigma^2} \right)$$

Usage

```
bi.dist.normal(
  loc = 0,
  scale = 1,
  validate_args = py_none(),
  name = "x",
  obs = py_none(),
  mask = py_none(),
  sample = FALSE,
  seed = py_none(),
  shape = c(),
  event = 0,
  create_obj = FALSE,
  to_jax = TRUE
)
```

Arguments

`loc`	A numeric vector, matrix, or array representing the mean of the distribution.
`scale`	A numeric vector, matrix, or array representing the standard deviation of the distribution.
`validate_args`	Logical: Whether to validate parameter values. Defaults to 'reticulate::py_none()'.
`name`	A character string representing the name of the random variable within a model. This is used to uniquely identify the variable. Defaults to 'x'.

obs	A numeric vector or array of observed values. If provided, the random variable is conditioned on these values. If 'NULL', the variable is treated as a latent (unobserved) variable. Defaults to 'NULL'.
mask	A logical vector, matrix, or array. Optional boolean array to mask observations.
sample	A logical value that controls the function's behavior. If 'TRUE', the function will directly draw samples from the distribution. If 'FALSE', it will create a random variable within a model. Defaults to 'FALSE'.
seed	An integer used to set the random seed for reproducibility when 'sample = TRUE'. This argument has no effect when 'sample = FALSE', as randomness is handled by the model's inference engine. Defaults to 0.
shape	A numeric vector specifying the shape of the distribution. Use a vector (e.g., 'c(10)') to define the shape.
event	An integer representing the number of batch dimensions to reinterpret as event dimensions (used in model building).
create_obj	A logical value. If 'TRUE', returns the raw BI distribution object instead of creating a sample site.
to_jax	Boolean. Indicates whether to return a JAX array or not.

Value

- When `sample=FALSE`: A BI Normal distribution object (for model building).

- When `sample=TRUE`: A JAX array of samples drawn from the Normal distribution (for direct sampling).

- When `create_obj=TRUE`: The raw BI distribution object (for advanced use cases).

Examples

```
library(BayesianInference)
m=importBI(platform='cpu')
bi.dist.normal(sample = TRUE)
```

`bi.dist.ordered_logistic`

Ordered Logistic Distribution

Description

The ordered logistic distribution is used for modeling **ordinal** outcome variables

$$Y \in 1, 2, \ldots, K$$

(i.e., categories with a natural order) via a latent continuous predictor

$$\eta$$

and a set of increasing *cut-points*

$$c_1 < c_2 < \cdots < c_{K-1}$$

. When

$$\eta$$

crosses thresholds, the observed

$$Y$$

.

Usage

```
bi.dist.ordered_logistic(
  predictor,
  cutpoints,
  validate_args = py_none(),
  name = "x",
  obs = py_none(),
  mask = py_none(),
  sample = FALSE,
  seed = py_none(),
  shape = c(),
  event = 0,
  create_obj = FALSE,
  to_jax = TRUE
)
```

Arguments

predictor	A numeric vector, matrix, or array representing the prediction in real domain; typically this is output of a linear model.
cutpoints	A numeric vector, matrix, or array representing the positions in real domain to separate categories.
validate_args	Logical: Whether to validate parameter values. Defaults to ‘reticulate::py_none()‘.
name	A character string representing the name of the random variable within a model. This is used to uniquely identify the variable. Defaults to ’x’.
obs	A numeric vector or array of observed values. If provided, the random variable is conditioned on these values. If ‘NULL‘, the variable is treated as a latent (unobserved) variable. Defaults to ‘NULL‘.
mask	An optional boolean vector to mask observations.
sample	A logical value that controls the function’s behavior. If ‘TRUE‘, the function will directly draw samples from the distribution. If ‘FALSE‘, it will create a random variable within a model. Defaults to ‘FALSE‘.
seed	An integer used to set the random seed for reproducibility when ‘sample = TRUE‘. This argument has no effect when ‘sample = FALSE‘, as randomness is handled by the model’s inference engine. Defaults to 0.

shape	A numeric vector used to shape the distribution. When ‘sample=False‘ (model building), this is used with ‘.expand(shape)‘ to set the distribution’s batch shape. When ‘sample=True‘ (direct sampling), this is used as ‘sample_shape‘ to draw a raw JAX array of the given shape.
event	An integer representing the number of batch dimensions to reinterpret as event dimensions (used in model building).
create_obj	A logical value. If ‘TRUE‘, returns the raw BI distribution object instead of creating a sample site.
to_jax	Boolean. Indicates whether to return a JAX array or not.

Value

- When `sample=FALSE`, a BI Ordered Logistic distribution object (for model building).

- When `sample=TRUE`, a JAX array of samples drawn from the Ordered Logistic distribution (for direct sampling).

- When `create_obj=TRUE`, the raw BI distribution object (for advanced use cases).

See Also

https://num.pyro.ai/en/stable/distributions.html#orderedlogistic

Examples

```
library(BayesianInference)
m=importBI(platform='cpu')
bi.dist.ordered_logistic(predictor = c(0.2, 0.5, 0.8), cutpoints = c(-1.0, 0.0, 1.0), sample = TRUE)
```

`bi.dist.pareto` *Samples from a Pareto distribution.*

Description

The **Pareto distribution**, named after economist Vilfredo Pareto, is a **power-law** probability distribution used to describe phenomena with "rich-get-richer" or "heavy-tail" properties - for example, income distribution, city sizes, or wealth concentration. It is characterized by: * a **scale parameter**

$$x_m > 0$$

(the minimum possible value), and * a **shape parameter**

$$\alpha > 0$$

(which controls the tail heaviness). A random variable

$$X \sim \text{Pareto}(\alpha, x_m)$$

takes values

$$x \geq x_m$$

.

Usage

```
bi.dist.pareto(
  scale,
  alpha,
  validate_args = py_none(),
  name = "x",
  obs = py_none(),
  mask = py_none(),
  sample = FALSE,
  seed = py_none(),
  shape = c(),
  event = 0,
  create_obj = FALSE,
  to_jax = TRUE
)
```

Arguments

scale	A numeric vector or single number representing the scale parameter of the Pareto distribution. Must be positive.
alpha	A numeric vector or single number representing the shape parameter of the Pareto distribution. Must be positive.
validate_args	Logical: Whether to validate parameter values. Defaults to 'reticulate::py_none()'.
name	A character string representing the name of the random variable within a model. This is used to uniquely identify the variable. Defaults to 'x'.
obs	A numeric vector or array of observed values. If provided, the random variable is conditioned on these values. If 'NULL', the variable is treated as a latent (unobserved) variable. Defaults to 'NULL'.
mask	A logical vector. Optional boolean array to mask observations.
sample	A logical value that controls the function's behavior. If 'TRUE', the function will directly draw samples from the distribution. If 'FALSE', it will create a random variable within a model. Defaults to 'FALSE'.
seed	An integer used to set the random seed for reproducibility when 'sample = TRUE'. This argument has no effect when 'sample = FALSE', as randomness is handled by the model's inference engine. Defaults to 0.
shape	A numeric vector. When `sample=FALSE` (model building), this is used with '.expand(shape)' to set the distribution's batch shape. When `sample=TRUE` (direct sampling), this is used as 'sample_shape' to draw a raw JAX array of the given shape.
event	An integer representing the number of batch dimensions to reinterpret as event dimensions (used in model building).
create_obj	A logical value. If 'TRUE', returns the raw BI distribution object instead of creating a sample site. This is essential for building complex distributions like 'MixtureSameFamily'.
to_jax	Boolean. Indicates whether to return a JAX array or not.

Value

- When sample=FALSE, a BI Pareto distribution object (for model building).

- When sample=TRUE, a JAX array of samples drawn from the Pareto distribution (for direct sampling).

- When create_obj=TRUE, the raw BI distribution object (for advanced use cases).

See Also

https://num.pyro.ai/en/stable/distributions.html#pareto

Examples

```
library(BayesianInference)
m=importBI(platform='cpu')
bi.dist.pareto(scale = c(0.2, 0.5, 0.8), alpha = c(-1.0, 0.5, 1.0), sample = TRUE)
```

---

bi.dist.poisson *Poisson Distribution*

Description

The **Poisson distribution** models the probability of observing a given number of events k occurring in a fixed interval of time or space when these events happen independently and at a constant average rate

$$\lambda > 0$$

. It is widely used for modeling **count data**, such as the number of emails received per hour or mutations in a DNA strand per unit length. Formally,

$$K \sim \text{Poisson}(\lambda)$$

where

$$\lambda$$

is both the **mean** and **variance** of the distribution.

Usage

```
bi.dist.poisson(
  rate,
  is_sparse = FALSE,
  validate_args = py_none(),
  name = "x",
  obs = py_none(),
  mask = py_none(),
  sample = FALSE,
  seed = py_none(),
```

```
  shape = c(),
  event = 0,
  create_obj = FALSE,
  to_jax = TRUE
)
```

Arguments

rate	A numeric vector representing the average number of events.
is_sparse	(bool, optional): Indicates whether the ‘rate‘ parameter is sparse. If ‘True‘, a specialized sparse sampling implementation is used, which can be more efficient for models with many zero-rate components (e.g., zero-inflated models). Defaults to ‘False‘.
validate_args	Logical: Whether to validate parameter values. Defaults to ‘reticulate::py_none()‘.
name	A character string representing the name of the random variable within a model. This is used to uniquely identify the variable. Defaults to ’x’.
obs	A numeric vector or array of observed values. If provided, the random variable is conditioned on these values. If ‘NULL‘, the variable is treated as a latent (unobserved) variable. Defaults to ‘NULL‘.
mask	A logical vector to mask observations.
sample	A logical value that controls the function’s behavior. If ‘TRUE‘, the function will directly draw samples from the distribution. If ‘FALSE‘, it will create a random variable within a model. Defaults to ‘FALSE‘.
seed	An integer used to set the random seed for reproducibility when ‘sample = TRUE‘. This argument has no effect when ‘sample = FALSE‘, as randomness is handled by the model’s inference engine. Defaults to 0.
shape	A numeric vector used for shaping. When ‘sample=False‘ (model building), this is used with ‘.expand(shape)‘ to set the distribution’s batch shape. When ‘sample=True‘ (direct sampling), this is used as ‘sample_shape‘ to draw a raw JAX array of the given shape.
event	An integer representing the number of batch dimensions to reinterpret as event dimensions (used in model building).
create_obj	A logical value. If ‘TRUE‘, returns the raw BI distribution object instead of creating a sample site.
to_jax	Boolean. Indicates whether to return a JAX array or not.

Value

- When `sample=FALSE`, a BI Poisson distribution object (for model building).

- When `sample=TRUE`, a JAX array of samples drawn from the Poisson distribution (for direct sampling).

- When `create_obj=TRUE`, the raw BI distribution object (for advanced use cases).

Examples

```
library(BayesianInference)
m=importBI(platform='cpu')
bi.dist.poisson(rate = c(0.2, 0.5, 0.8), sample = TRUE)
```

`bi.dist.projected_normal`

Samples from a Projected Normal distribution.

Description

The projected normal distribution arises by taking a multivariate normal vector

$$\mathbf{X} \sim \mathcal{N}_n(\mu, \Sigma)$$

in

$$\mathbb{R}^n$$

and projecting it to the unit sphere. This distribution is commonly used in directional statistics (data on circles or spheres) and supports asymmetric and even multimodal behaviours depending on parameters.

Usage

```
bi.dist.projected_normal(
  concentration,
  validate_args = py_none(),
  name = "x",
  obs = py_none(),
  mask = py_none(),
  sample = FALSE,
  seed = py_none(),
  shape = c(),
  event = 0,
  create_obj = FALSE,
  to_jax = TRUE
)
```

Arguments

`concentration`	A numeric vector representing the concentration parameter, representing the direction towards which the samples are concentrated.
`validate_args`	Logical: Whether to validate parameter values. Defaults to 'reticulate::py_none()'.
`name`	A character string representing the name of the random variable within a model. This is used to uniquely identify the variable. Defaults to 'x'.

obs	A numeric vector or array of observed values. If provided, the random variable is conditioned on these values. If 'NULL', the variable is treated as a latent (unobserved) variable. Defaults to 'NULL'.
mask	An optional boolean array to mask observations.
sample	A logical value that controls the function's behavior. If 'TRUE', the function will directly draw samples from the distribution. If 'FALSE', it will create a random variable within a model. Defaults to 'FALSE'.
seed	An integer used to set the random seed for reproducibility when 'sample = TRUE'. This argument has no effect when 'sample = FALSE', as randomness is handled by the model's inference engine. Defaults to 0.
shape	A numeric vector used for shaping. When `sample=FALSE` (model building), this is used with '.expand(shape)' to set the distribution's batch shape. When `sample=TRUE` (direct sampling), this is used as 'sample_shape' to draw a raw JAX array of the given shape.
event	The number of batch dimensions to reinterpret as event dimensions (used in model building).
create_obj	Logical; If 'TRUE', returns the raw BI distribution object instead of creating a sample site. This is essential for building complex distributions like 'MixtureSameFamily'.
to_jax	Boolean. Indicates whether to return a JAX array or not.

Value

- When `sample=FALSE`, a BI Projected Normal distribution object (for model building).

- When `sample=TRUE`, a JAX array of samples drawn from the Projected Normal distribution (for direct sampling).

- When `create_obj=TRUE`, the raw BI distribution object (for advanced use cases).

See Also

https://num.pyro.ai/en/stable/distributions.html#projectednormal

Examples

```
library(BayesianInference)
m=importBI(platform='cpu')
bi.dist.projected_normal(concentration = c(1.0, 3.0, 2.0), sample = TRUE)
```

`bi.dist.relaxed_bernoulli` *Samples from a Relaxed Bernoulli distribution.*

Description

The Relaxed Bernoulli is a continuous distribution on the interval

$$(0, 1)$$

that smoothly approximates the discrete Bernoulli distribution (which has support

$$0, 1$$

). It was introduced to allow for *differentiable* sampling of approximate binary random variables, which is useful in variational inference and other gradient-based optimization settings.

Usage

```
bi.dist.relaxed_bernoulli(
  temperature,
  probs = py_none(),
  logits = py_none(),
  validate_args = py_none(),
  name = "x",
  obs = py_none(),
  mask = py_none(),
  sample = FALSE,
  seed = py_none(),
  shape = c(),
  event = 0,
  create_obj = FALSE,
  to_jax = TRUE
)
```

Arguments

`temperature`	A numeric value representing the temperature parameter.
`probs`	(jnp.ndarray, optional): The probability of success. Must be in the interval ‘[0, 1]‘. Only one of ‘probs‘ or ‘logits‘ can be specified.
`logits`	A numeric vector or matrix representing the logits parameter.
`validate_args`	Logical: Whether to validate parameter values. Defaults to ‘reticulate::py_none()‘.
`name`	A character string representing the name of the random variable within a model. This is used to uniquely identify the variable. Defaults to ’x’.
`obs`	A numeric vector or array of observed values. If provided, the random variable is conditioned on these values. If ‘NULL‘, the variable is treated as a latent (unobserved) variable. Defaults to ‘NULL‘.

mask	A logical vector or array to mask observations.
sample	A logical value that controls the function's behavior. If 'TRUE', the function will directly draw samples from the distribution. If 'FALSE', it will create a random variable within a model. Defaults to 'FALSE'.
seed	An integer used to set the random seed for reproducibility when 'sample = TRUE'. This argument has no effect when 'sample = FALSE', as randomness is handled by the model's inference engine. Defaults to 0.
shape	A numeric vector (e.g., 'c(10)') specifying the shape. When 'sample=False' (model building), this is used with '.expand(shape)' to set the distribution's batch shape. When 'sample=True' (direct sampling), this is used as 'sample_shape' to draw a raw JAX array of the given shape.
event	An integer representing the number of batch dimensions to reinterpret as event dimensions (used in model building).
create_obj	A logical value. If 'TRUE', returns the raw BI distribution object instead of creating a sample site. This is essential for building complex distributions like 'MixtureSameFamily'.
to_jax	Boolean. Indicates whether to return a JAX array or not.

Value

- When `sample=FALSE`, a BI Relaxed Bernoulli distribution object (for model building).

- When `sample=TRUE`, a JAX array of samples drawn from the Relaxed Bernoulli distribution (for direct sampling).

- When `create_obj=TRUE`, the raw BI distribution object (for advanced use cases).

See Also

https://num.pyro.ai/en/stable/distributions.html#relaxedbernoulli

Examples

```
library(BayesianInference)
m=importBI(platform='cpu')
bi.dist.relaxed_bernoulli(temperature = c(1,1), logits = 0.0, sample = TRUE)
```

---

bi.dist.relaxed_bernoulli_logits

Relaxed Bernoulli Logits Distribution.

Description

The Relaxed Bernoulli (logits) is a **continuous** relaxation of the standard Bernoulli distribution, parameterised by logits (or probabilities) and a **temperature** parameter. Rather than producing strictly 0 or 1, it produces values in the continuous interval (0, 1). As the temperature $\to 0$, the distribution approximates a Bernoulli; as temperature $\to \infty$, the distribution approximates a uniform distribution.

Usage

```
bi.dist.relaxed_bernoulli_logits(
  temperature,
  logits,
  validate_args = py_none(),
  name = "x",
  obs = py_none(),
  mask = py_none(),
  sample = FALSE,
  seed = py_none(),
  shape = c(),
  event = 0,
  create_obj = FALSE,
  to_jax = TRUE
)
```

Arguments

temperature	A numeric vector or matrix representing the temperature parameter, must be positive.
logits	A numeric vector or matrix representing the logits parameter.
validate_args	Logical: Whether to validate parameter values. Defaults to ‘reticulate::py_none()‘.
name	A character string representing the name of the random variable within a model. This is used to uniquely identify the variable. Defaults to ’x’.
obs	A numeric vector or array of observed values. If provided, the random variable is conditioned on these values. If ‘NULL‘, the variable is treated as a latent (unobserved) variable. Defaults to ‘NULL‘.
mask	A logical vector indicating observations to mask.
sample	A logical value that controls the function’s behavior. If ‘TRUE‘, the function will directly draw samples from the distribution. If ‘FALSE‘, it will create a random variable within a model. Defaults to ‘FALSE‘.
seed	An integer used to set the random seed for reproducibility when ‘sample = TRUE‘. This argument has no effect when ‘sample = FALSE‘, as randomness is handled by the model’s inference engine. Defaults to 0.
shape	A numeric vector specifying the shape of the distribution. When ‘sample=False‘ (model building), this is used with ‘.expand(shape)‘ to set the distribution’s batch shape. When ‘sample=True‘ (direct sampling), this is used as ‘sample_shape‘ to draw a raw JAX array of the given shape.
event	An integer representing the number of batch dimensions to reinterpret as event dimensions (used in model building).
create_obj	A logical value. If ‘TRUE‘, returns the raw BI distribution object instead of creating a sample site. This is essential for building complex distributions like ‘MixtureSameFamily‘.
to_jax	Boolean. Indicates whether to return a JAX array or not.

Value

- When sample=FALSE, a BI Relaxed Bernoulli Logits distribution object (for model building).

- When sample=TRUE, a JAX array of samples drawn from the Relaxed Bernoulli Logits distribution (for direct sampling).

- When create_obj=TRUE, the raw BI distribution object (for advanced use cases).

See Also

https://num.pyro.ai/en/stable/distributions.html#relaxed-bernoulli-logits

Examples

```
library(BayesianInference)
m=importBI(platform='cpu')
bi.dist.relaxed_bernoulli_logits(1, 0.1, sample = TRUE)
```

---

bi.dist.right_truncated_distribution

Samples from a right-truncated distribution.

Description

A right-truncated distribution is a statistical distribution that arises when the possible values of a random variable are restricted to be below a certain specified value ‘high‘. In essence, the right tail of the original distribution is "cut off" at a particular point, and the remaining probability is redistributed among the allowable values. This type of distribution is common in various fields where there are inherent upper limits or observational constraints.

Usage

```
bi.dist.right_truncated_distribution(
  base_dist,
  high = 0,
  validate_args = py_none(),
  name = "x",
  obs = py_none(),
  mask = py_none(),
  sample = FALSE,
  seed = py_none(),
  shape = c(),
  event = 0,
  create_obj = FALSE,
  to_jax = TRUE
)
```

Arguments

base_dist	The base distribution to truncate. Must be a univariate distribution with real support.
high	(float, jnp.ndarray, optional): The upper truncation point. The support of the new distribution is $(-\infty, \text{high})$. Defaults to 0.0.
validate_args	Logical: Whether to validate parameter values. Defaults to 'reticulate::py_none()'.
name	A character string representing the name of the random variable within a model. This is used to uniquely identify the variable. Defaults to 'x'.
obs	A numeric vector or array of observed values. If provided, the random variable is conditioned on these values. If 'NULL', the variable is treated as a latent (unobserved) variable. Defaults to 'NULL'.
mask	An optional boolean vector to mask observations.
sample	A logical value that controls the function's behavior. If 'TRUE', the function will directly draw samples from the distribution. If 'FALSE', it will create a random variable within a model. Defaults to 'FALSE'.
seed	An integer used to set the random seed for reproducibility when 'sample = TRUE'. This argument has no effect when 'sample = FALSE', as randomness is handled by the model's inference engine. Defaults to 0.
shape	A numeric vector. When sample=FALSE (model building), this is used with '.expand(shape)' to set the distribution's batch shape. When sample=TRUE (direct sampling), this is used as 'sample_shape' to draw a raw JAX array of the given shape.
event	The number of batch dimensions to reinterpret as event dimensions (used in model building).
create_obj	Logical. If 'TRUE', returns the raw BI distribution object instead of creating a sample site.
to_jax	Boolean. Indicates whether to return a JAX array or not.

Value

- When sample=FALSE, a BI right-truncated distribution object (for model building).

- When sample=TRUE, a JAX array of samples drawn from the right-truncated distribution (for direct sampling).

- When create_obj=TRUE, the raw BI distribution object (for advanced use cases).

Examples

```
library(BayesianInference)
m=importBI(platform='cpu')
bi.dist.right_truncated_distribution(
base_dist = bi.dist.normal(0,1, create_obj = TRUE),
high = 10,
sample = TRUE)
```

`bi.dist.soft_laplace` *Samples from a Soft Laplace distribution.*

Description

This distribution is a smooth approximation of a Laplace distribution, characterized by its log-convex density. It offers Laplace-like tails while being infinitely differentiable, making it suitable for HMC and Laplace approximation.

Usage

```
bi.dist.soft_laplace(
  loc,
  scale,
  validate_args = py_none(),
  name = "x",
  obs = py_none(),
  mask = py_none(),
  sample = FALSE,
  seed = py_none(),
  shape = c(),
  event = 0,
  create_obj = FALSE,
  to_jax = TRUE
)
```

Arguments

`loc`	Location parameter.
`scale`	Scale parameter.
`validate_args`	Logical: Whether to validate parameter values. Defaults to 'reticulate::py_none()'.
`name`	A character string representing the name of the random variable within a model. This is used to uniquely identify the variable. Defaults to 'x'.
`obs`	A numeric vector or array of observed values. If provided, the random variable is conditioned on these values. If 'NULL', the variable is treated as a latent (unobserved) variable. Defaults to 'NULL'.
`mask`	An optional boolean vector to mask observations.
`sample`	A logical value that controls the function's behavior. If 'TRUE', the function will directly draw samples from the distribution. If 'FALSE', it will create a random variable within a model. Defaults to 'FALSE'.
`seed`	An integer used to set the random seed for reproducibility when 'sample = TRUE'. This argument has no effect when 'sample = FALSE', as randomness is handled by the model's inference engine. Defaults to 0.

shape	A numeric vector specifying the shape. When sample=FALSE (model building), this is used with '.expand(shape)' to set the distribution's batch shape. When sample=TRUE (direct sampling), this is used as 'sample_shape' to draw a raw JAX array of the given shape.
event	The number of batch dimensions to reinterpret as event dimensions (used in model building).
create_obj	Logical; If 'TRUE', returns the raw BI distribution object instead of creating a sample site.
to_jax	Boolean. Indicates whether to return a JAX array or not.

Value

- When sample=FALSE, a BI Soft Laplace distribution object (for model building).

- When sample=TRUE, a JAX array of samples drawn from the Soft Laplace distribution (for direct sampling).

- When create_obj=TRUE, the raw BI distribution object (for advanced use cases).

Examples

```
library(BayesianInference)
m=importBI(platform='cpu')
bi.dist.soft_laplace(loc = 0, scale = 2, sample = TRUE)
```

---

bi.dist.student_t *Student's t-distribution.*

Description

The Student's t-distribution is a probability distribution that arises in hypothesis testing involving the mean of a normally distributed population when the population standard deviation is unknown. It is similar to the normal distribution, but has heavier tails, making it more robust to outliers. For large

$$\nu$$

, it converges to the Normal distribution.

$$X \sim t_\nu(\mu, \sigma)$$

where:

*

$$\mu$$

is the **location (mean)** parameter *

$$\sigma > 0$$

is the **scale** parameter *

$$\nu > 0$$

is the **degrees of freedom** controlling the tail heaviness

Usage

```
bi.dist.student_t(
  df,
  loc = 0,
  scale = 1,
  validate_args = py_none(),
  name = "x",
  obs = py_none(),
  mask = py_none(),
  sample = FALSE,
  seed = py_none(),
  shape = c(),
  event = 0,
  create_obj = FALSE,
  to_jax = TRUE
)
```

Arguments

df	A numeric vector representing degrees of freedom, must be positive.
loc	A numeric vector representing the location parameter, defaults to 0.0.
scale	A numeric vector representing the scale parameter, defaults to 1.0.
validate_args	Logical: Whether to validate parameter values. Defaults to 'reticulate::py_none()'.
name	A character string representing the name of the random variable within a model. This is used to uniquely identify the variable. Defaults to 'x'.
obs	A numeric vector or array of observed values. If provided, the random variable is conditioned on these values. If 'NULL', the variable is treated as a latent (unobserved) variable. Defaults to 'NULL'.
mask	A logical vector to mask observations.
sample	A logical value that controls the function's behavior. If 'TRUE', the function will directly draw samples from the distribution. If 'FALSE', it will create a random variable within a model. Defaults to 'FALSE'.
seed	An integer used to set the random seed for reproducibility when 'sample = TRUE'. This argument has no effect when 'sample = FALSE', as randomness is handled by the model's inference engine. Defaults to 0.
shape	A numeric vector. When 'sample=False' (model building), this is used with '.expand(shape)' to set the distribution's batch shape. When 'sample=True' (direct sampling), this is used as 'sample_shape' to draw a raw JAX array of the given shape.
event	Integer representing the number of batch dimensions to reinterpret as event dimensions (used in model building).
create_obj	Logical. If 'TRUE', returns the raw BI distribution object instead of creating a sample site. This is essential for building complex distributions like 'MixtureSameFamily'.
to_jax	Boolean. Indicates whether to return a JAX array or not.

Value

- When sample=FALSE, a BI Student's t-distribution object (for model building).

- When sample=TRUE, a JAX array of samples drawn from the Student's t-distribution (for direct sampling).

- When create_obj=TRUE, the raw BI distribution object (for advanced use cases).

See Also

https://num.pyro.ai/en/stable/distributions.html#studentt

Examples

```
library(BayesianInference)
m=importBI(platform='cpu')
bi.dist.student_t(df = 2, loc = 0, scale = 2, sample = TRUE)
```

---

bi.dist.truncated_cauchy

Truncated Cauchy Distribution

Description

The Cauchy distribution, also known as the Lorentz distribution, is a continuous probability distribution that appears frequently in various areas of mathematics and physics. It is characterized by its heavy tails, which extend to infinity. The truncated version limits the support of the Cauchy distribution to a specified interval.

Usage

```
bi.dist.truncated_cauchy(
  loc = 0,
  scale = 1,
  low = py_none(),
  high = py_none(),
  validate_args = py_none(),
  name = "x",
  obs = py_none(),
  mask = py_none(),
  sample = FALSE,
  seed = py_none(),
  shape = c(),
  event = 0,
  create_obj = FALSE,
  to_jax = TRUE
)
```

Arguments

loc	Location parameter of the Cauchy distribution.
scale	Scale parameter of the Cauchy distribution.
low	(float, jnp.ndarray, optional): The lower truncation point. If 'None', the distribution is only truncated on the right. Defaults to 'None'.
high	(float, jnp.ndarray, optional): The upper truncation point. If 'None', the distribution is only truncated on the left. Defaults to 'None'. validate_args (bool, optional): Whether to enable validation of distribution parameters. Defaults to 'None'.
validate_args	Logical: Whether to validate parameter values. Defaults to 'reticulate::py_none()'.
name	A character string representing the name of the random variable within a model. This is used to uniquely identify the variable. Defaults to 'x'.
obs	A numeric vector or array of observed values. If provided, the random variable is conditioned on these values. If 'NULL', the variable is treated as a latent (unobserved) variable. Defaults to 'NULL'.
mask	An optional boolean array to mask observations.
sample	A logical value that controls the function's behavior. If 'TRUE', the function will directly draw samples from the distribution. If 'FALSE', it will create a random variable within a model. Defaults to 'FALSE'.
seed	An integer used to set the random seed for reproducibility when 'sample = TRUE'. This argument has no effect when 'sample = FALSE', as randomness is handled by the model's inference engine. Defaults to 0.
shape	A numeric vector. When 'sample=False' (model building), this is used with '.expand(shape)' to set the distribution's batch shape. When 'sample=True' (direct sampling), this is used as 'sample_shape' to draw a raw JAX array of the given shape.
event	The number of batch dimensions to reinterpret as event dimensions (used in model building).
create_obj	Logical; If 'TRUE', returns the raw BI distribution object instead of creating a sample site. This is essential for building complex distributions like 'MixtureSameFamily'.
to_jax	Boolean. Indicates whether to return a JAX array or not.

Value

- When `sample=FALSE`, a BI Truncated Cauchy distribution object (for model building).

- When `sample=TRUE`, a JAX array of samples drawn from the Truncated Cauchy distribution (for direct sampling).

- When `create_obj=TRUE`, the raw BI distribution object (for advanced use cases).

Examples

```
library(BayesianInference)
m=importBI(platform='cpu')
bi.dist.truncated_cauchy(loc = 0, scale = 2, low = 0, high = 1.5, sample = TRUE)
```

`bi.dist.truncated_distribution` *Truncated Distribution*

Description

A **truncated distribution** arises when you take a random variable

$$X$$

that originally has some distribution (with PDF

$$f_X(x)$$

and CDF

$$F_X(x)$$

) and you restrict attention only to those values of

$$X$$

that are *above* a given truncation point

$$a$$

. In other words you only observe X when $X > a$. All the "mass" below (or equal to)

$$a$$

is **excluded** (not just unobserved, but removed from the sample/analysis). This differs from *censoring*, where values below a threshold might be known (for example "< a"), but here they are entirely excluded from the domain. Left truncation is common in many applied fields.

Usage

```
bi.dist.truncated_distribution(
  base_dist,
  low = py_none(),
  high = py_none(),
  validate_args = py_none(),
  name = "x",
  obs = py_none(),
  mask = py_none(),
  sample = FALSE,
  seed = py_none(),
  shape = c(),
  event = 0,
  create_obj = FALSE,
  to_jax = TRUE
)
```

Arguments

base_dist	The base distribution to be truncated. This should be a univariate distribution. Currently, only the following distributions are supported: Cauchy, Laplace, Logistic, Normal, and StudentT.
low	(float, jnp.ndarray, optional): The lower truncation point. If 'None', the distribution is only truncated on the right. Defaults to 'None'.
high	(float, jnp.ndarray, optional): The upper truncation point. If 'None', the distribution is only truncated on the left. Defaults to 'None'.
validate_args	Logical: Whether to validate parameter values. Defaults to 'reticulate::py_none()'.
name	A character string representing the name of the random variable within a model. This is used to uniquely identify the variable. Defaults to 'x'.
obs	A numeric vector or array of observed values. If provided, the random variable is conditioned on these values. If 'NULL', the variable is treated as a latent (unobserved) variable. Defaults to 'NULL'.
mask	An optional boolean array to mask observations.
sample	A logical value that controls the function's behavior. If 'TRUE', the function will directly draw samples from the distribution. If 'FALSE', it will create a random variable within a model. Defaults to 'FALSE'.
seed	An integer used to set the random seed for reproducibility when 'sample = TRUE'. This argument has no effect when 'sample = FALSE', as randomness is handled by the model's inference engine. Defaults to 0.
shape	A numeric vector (e.g., 'c(10)') specifying the shape. When sample=FALSE (model building), this is used with '.expand(shape)' to set the distribution's batch shape. When sample=TRUE (direct sampling), this is used as 'sample_shape' to draw a raw JAX array of the given shape.
event	The number of batch dimensions to reinterpret as event dimensions (used in model building).
create_obj	Logical; If 'TRUE', returns the raw BI distribution object instead of creating a sample site. This is essential for building complex distributions like 'MixtureSameFamily'.
to_jax	Boolean. Indicates whether to return a JAX array or not.

Value

- When sample=FALSE, a BI Truncated distribution object (for model building).

- When sample=TRUE, a JAX array of samples drawn from the Truncated distribution (for direct sampling).

- When create_obj=TRUE, the raw BI distribution object (for advanced use cases).

See Also

https://num.pyro.ai/en/stable/distributions.html#truncateddistribution

Examples

```
library(BayesianInference)
m=importBI(platform='cpu')
bi.dist.truncated_distribution(
base_dist = bi.dist.normal(0,1, create_obj = TRUE),
high = 0.7,
low = 0.1,
sample = TRUE)
```

`bi.dist.truncated_normal`

Truncated Normal Distribution

Description

A truncated normal distribution is derived from a normal (Gaussian) random variable by restricting (truncating) its domain to an interval

$$[a, b]$$

(which could be one-sided, e.g., (a) only or (b) only). It is defined by its location ('loc'), scale ('scale'), lower bound

$$a$$

('low'), and upper bound

$$b$$

('high'). In effect: if

$$X \sim \mathcal{N}(\mu, \sigma^2)$$

, then the truncated version

$$Y = X|(a \leq X \leq b)$$

has the same "shape" but only supports values in

$$[a, b]$$

. This is used when you know that values outside a range are impossible or not observed (e.g., measurement limits, natural bounds).

Usage

```
bi.dist.truncated_normal(
  loc = 0,
  scale = 1,
  low = py_none(),
  high = py_none(),
  validate_args = py_none(),
  name = "x",
  obs = py_none(),
```

```
  mask = py_none(),
  sample = FALSE,
  seed = py_none(),
  shape = c(),
  event = 0,
  create_obj = FALSE,
  to_jax = TRUE
)
```

Arguments

loc	The location parameter of the normal distribution.
scale	The scale parameter of the normal distribution.
low	(float, jnp.ndarray, optional): The lower truncation point. If 'None', the distribution is only truncated on the right. Defaults to 'None'.
high	(float, jnp.ndarray, optional): The upper truncation point. If 'None', the distribution is only truncated on the left. Defaults to 'None'.
validate_args	Logical: Whether to validate parameter values. Defaults to 'reticulate::py_none()'.
name	A character string representing the name of the random variable within a model. This is used to uniquely identify the variable. Defaults to 'x'.
obs	A numeric vector or array of observed values. If provided, the random variable is conditioned on these values. If 'NULL', the variable is treated as a latent (unobserved) variable. Defaults to 'NULL'.
mask	An optional boolean array to mask observations.
sample	A logical value that controls the function's behavior. If 'TRUE', the function will directly draw samples from the distribution. If 'FALSE', it will create a random variable within a model. Defaults to 'FALSE'.
seed	An integer used to set the random seed for reproducibility when 'sample = TRUE'. This argument has no effect when 'sample = FALSE', as randomness is handled by the model's inference engine. Defaults to 0.
shape	A numeric vector (e.g., 'c(10)') used to shape the distribution. When 'sample=False' (model building), this is used with '.expand(shape)' to set the distribution's batch shape. When 'sample=True' (direct sampling), this is used as 'sample_shape' to draw a raw JAX array of the given shape.
event	The number of batch dimensions to reinterpret as event dimensions (used in model building).
create_obj	Logical; If 'TRUE', returns the raw BI distribution object instead of creating a sample site.
to_jax	Boolean. Indicates whether to return a JAX array or not.

Value

- When `sample=FALSE`, a BI Truncated Normal distribution object (for model building).

- When `sample=TRUE`, a JAX array of samples drawn from the Truncated Normal distribution (for direct sampling).

- When `create_obj=TRUE`, the raw BI distribution object (for advanced use cases).

Examples

```
library(BayesianInference)
m=importBI(platform='cpu')
bi.dist.truncated_normal(loc = 0, scale = 2, low = 0, high = 1.5, sample = TRUE)
```

`bi.dist.truncated_polya_gamma`

Truncated PolyaGamma Distribution

Description

This distribution is a truncated version of the PolyaGamma distribution, defined over the interval [0, truncation_point]. It is often used in Bayesian non-parametric models.

$$p(x) = \frac{1}{Z} \exp\left(\sum_{n=0}^{N}\left(\log(2n+1) - 1.5\log(x) - \frac{(2n+1)^2}{4x}\right)\right)$$

Usage

```
bi.dist.truncated_polya_gamma(
  batch_shape = c(),
  validate_args = py_none(),
  name = "x",
  obs = py_none(),
  mask = py_none(),
  sample = FALSE,
  seed = py_none(),
  shape = c(),
  event = 0,
  create_obj = FALSE,
  to_jax = TRUE
)
```

Arguments

`batch_shape`	A numeric vector specifying the shape of the batch dimension.
`validate_args`	Logical: Whether to validate parameter values. Defaults to ‘reticulate::py_none()‘.
`name`	A character string representing the name of the random variable within a model. This is used to uniquely identify the variable. Defaults to ’x’.
`obs`	A numeric vector or array of observed values. If provided, the random variable is conditioned on these values. If ‘NULL‘, the variable is treated as a latent (unobserved) variable. Defaults to ‘NULL‘.
`mask`	A numeric vector, matrix, or array (e.g., a JAX array) of boolean values to mask observations.

sample	A logical value that controls the function's behavior. If 'TRUE', the function will directly draw samples from the distribution. If 'FALSE', it will create a random variable within a model. Defaults to 'FALSE'.
seed	An integer used to set the random seed for reproducibility when 'sample = TRUE'. This argument has no effect when 'sample = FALSE', as randomness is handled by the model's inference engine. Defaults to 0.
shape	A numeric vector (e.g., 'c(10)') used to shape the distribution. When 'sample=False' (model building), this is used with '.expand(shape)' to set the distribution's batch shape. When 'sample=True' (direct sampling), this is used as 'sample_shape' to draw a raw JAX array of the given shape.
event	An integer representing the number of batch dimensions to reinterpret as event dimensions.
create_obj	A logical value. If 'TRUE', returns the raw BI distribution object instead of creating a sample site.
to_jax	Boolean. Indicates whether to return a JAX array or not.

Value

- When `sample=FALSE`, a BI Truncated PolyaGamma distribution object (for model building).

- When `sample=TRUE`, a JAX array of samples drawn from the Truncated PolyaGamma distribution (for direct sampling).

- When `create_obj=TRUE`, the raw BI distribution object (for advanced use cases).

Examples

```
library(BayesianInference)
m=importBI(platform='cpu')
bi.dist.truncated_polya_gamma(batch_shape = c(), sample = TRUE)
```

`bi.dist.two_sided_truncated_distribution`

Two-Sided Truncated Distribution

Description

A "two-sided truncated distribution" is a general concept: you take a base continuous distribution and **restrict it** to an interval (['low', 'high']), discarding all mass outside, then **renormalize** so the inner portion integrates to 1. I'll spell out the general formulas, caveats, sampling strategies, and special cases (e.g. truncated normal) to illustrate.

Usage

```
bi.dist.two_sided_truncated_distribution(
  base_dist,
  low = 0,
  high = 1,
  validate_args = py_none(),
  name = "x",
  obs = py_none(),
  mask = py_none(),
  sample = FALSE,
  seed = py_none(),
  shape = c(),
  event = 0,
  create_obj = FALSE,
  to_jax = TRUE
)
```

Arguments

base_dist	The base distribution to truncate.
low	The lower bound for truncation.
high	The upper bound for truncation.
validate_args	Logical: Whether to validate parameter values. Defaults to 'reticulate::py_none()'.
name	A character string representing the name of the random variable within a model. This is used to uniquely identify the variable. Defaults to 'x'.
obs	A numeric vector or array of observed values. If provided, the random variable is conditioned on these values. If 'NULL', the variable is treated as a latent (unobserved) variable. Defaults to 'NULL'.
mask	A logical vector to mask observations.
sample	A logical value that controls the function's behavior. If 'TRUE', the function will directly draw samples from the distribution. If 'FALSE', it will create a random variable within a model. Defaults to 'FALSE'.
seed	An integer used to set the random seed for reproducibility when 'sample = TRUE'. This argument has no effect when 'sample = FALSE', as randomness is handled by the model's inference engine. Defaults to 0.
shape	A numeric vector. When 'sample=False' (model building), this is used with '.expand(shape)' to set the distribution's batch shape. When 'sample=True' (direct sampling), this is used as 'sample_shape' to draw a raw JAX array of the given shape.
event	Integer representing the number of batch dimensions to reinterpret as event dimensions (used in model building).
create_obj	Logical; if 'TRUE', returns the raw BI distribution object. Defaults to 'FALSE'.
to_jax	Boolean. Indicates whether to return a JAX array or not.

Value

- When sample=FALSE, a BI Two-Sided Truncated distribution object (for model building).

- When sample=TRUE, a JAX array of samples drawn from the Two-Sided Truncated distribution (for direct sampling).

- When create_obj=TRUE, the raw BI distribution object (for advanced use cases).

See Also

https://num.pyro.ai/en/stable/distributions.html#twosidedtruncateddistribution

Examples

```
library(BayesianInference)
m=importBI(platform='cpu')
bi.dist.two_sided_truncated_distribution(
base_dist = bi.dist.normal(0,1, create_obj = TRUE),
high = 0.5, low = 0.1, sample = TRUE)
```

---

bi.dist.uniform *Uniform Distribution*

Description

The Uniform distribution is the simplest continuous distribution: every value in the interval ([a, b]) is **equally likely**. It is widely used for modeling complete randomness within a fixed range, random sampling, and as a building block for other distributions.

Usage

```
bi.dist.uniform(
  low = 0,
  high = 1,
  validate_args = py_none(),
  name = "x",
  obs = py_none(),
  mask = py_none(),
  sample = FALSE,
  seed = py_none(),
  shape = c(),
  event = 0,
  create_obj = FALSE,
  to_jax = TRUE
)
```

Arguments

low	A numeric vector, matrix, or array representing the lower bound of the uniform interval.
high	A numeric vector, matrix, or array representing the upper bound of the uniform interval.
validate_args	Logical: Whether to validate parameter values. Defaults to 'reticulate::py_none()'.
name	A character string representing the name of the random variable within a model. This is used to uniquely identify the variable. Defaults to 'x'.
obs	A numeric vector or array of observed values. If provided, the random variable is conditioned on these values. If 'NULL', the variable is treated as a latent (unobserved) variable. Defaults to 'NULL'.
mask	A logical vector, matrix, or array (optional) to mask observations.
sample	A logical value that controls the function's behavior. If 'TRUE', the function will directly draw samples from the distribution. If 'FALSE', it will create a random variable within a model. Defaults to 'FALSE'.
seed	An integer used to set the random seed for reproducibility when 'sample = TRUE'. This argument has no effect when 'sample = FALSE', as randomness is handled by the model's inference engine. Defaults to 0.
shape	A numeric vector specifying the shape of the output. When sample=FALSE (model building), this is used with '.expand(shape)' to set the distribution's batch shape. When sample=TRUE (direct sampling), this is used as 'sample_shape' to draw a raw JAX array of the given shape.
event	An integer representing the number of batch dimensions to reinterpret as event dimensions (used in model building).
create_obj	A logical value. If 'TRUE', returns the raw BI distribution object instead of creating a sample site.
to_jax	Boolean. Indicates whether to return a JAX array or not.

Value

- When sample=FALSE, a BI Uniform distribution object (for model building).

- When sample=TRUE, a JAX array of samples drawn from the Uniform distribution (for direct sampling).

- When create_obj=TRUE, the raw BI distribution object (for advanced use cases).

Examples

```
library(BayesianInference)
m=importBI(platform='cpu')
bi.dist.uniform(low = 0, high = 1.5, sample = TRUE)
```

`bi.dist.weibull` *Samples from a Weibull distribution.*

Description

The Weibull distribution is widely used for modeling **lifetime or reliability data**. Its shape parameter (k) controls the hazard function: * (k < 1): decreasing hazard (infant mortality) * (k = 1): constant hazard ??? reduces to **Exponential distribution** * (k > 1): increasing hazard (aging/failure over time)

Usage

```
bi.dist.weibull(
  scale,
  concentration,
  validate_args = py_none(),
  name = "x",
  obs = py_none(),
  mask = py_none(),
  sample = FALSE,
  seed = py_none(),
  shape = c(),
  event = 0,
  create_obj = FALSE,
  to_jax = TRUE
)
```

Arguments

`scale`	A numeric vector, matrix, or array representing the scale parameter of the Weibull distribution. Must be positive.
`concentration`	A numeric vector, matrix, or array representing the shape parameter of the Weibull distribution. Must be positive.
`validate_args`	Logical: Whether to validate parameter values. Defaults to 'reticulate::py_none()'.
`name`	A character string representing the name of the random variable within a model. This is used to uniquely identify the variable. Defaults to 'x'.
`obs`	A numeric vector or array of observed values. If provided, the random variable is conditioned on these values. If 'NULL', the variable is treated as a latent (unobserved) variable. Defaults to 'NULL'.
`mask`	An optional boolean vector to mask observations.
`sample`	A logical value that controls the function's behavior. If 'TRUE', the function will directly draw samples from the distribution. If 'FALSE', it will create a random variable within a model. Defaults to 'FALSE'.
`seed`	An integer used to set the random seed for reproducibility when 'sample = TRUE'. This argument has no effect when 'sample = FALSE', as randomness is handled by the model's inference engine. Defaults to 0.

shape	A numeric vector. This is used with '.expand(shape)' when 'sample=False' (model building) to set the distribution's batch shape. When 'sample=True' (direct sampling), this is used as 'sample_shape' to draw a raw JAX array of the given shape.
event	An integer representing the number of batch dimensions to reinterpret as event dimensions (used in model building).
create_obj	A logical value. If 'TRUE', returns the raw BI distribution object instead of creating a sample site.
to_jax	Boolean. Indicates whether to return a JAX array or not.

Value

- When sample=FALSE, a BI Weibull distribution object (for model building).

- When sample=TRUE, a JAX array of samples drawn from the Weibull distribution (for direct sampling).

- When create_obj=TRUE, the raw BI distribution object (for advanced use cases).

See Also

https://num.pyro.ai/en/stable/distributions.html#weibull

Examples

```
library(BayesianInference)
m=importBI(platform='cpu')
bi.dist.weibull(scale = c(10, 10), concentration = c(1,1), sample = TRUE)
```

---

bi.dist.wishart *Wishart distribution for covariance matrices.*

Description

The Wishart distribution is a multivariate distribution used to model positive definite matrices, often representing covariance matrices. It's commonly used in Bayesian statistics and machine learning, particularly in models involving covariance estimation.

Usage

```
bi.dist.wishart(
  concentration,
  scale_matrix = py_none(),
  rate_matrix = py_none(),
  scale_tril = py_none(),
  validate_args = py_none(),
  name = "x",
```

```
  obs = py_none(),
  mask = py_none(),
  sample = FALSE,
  seed = py_none(),
  shape = c(),
  event = 0,
  create_obj = FALSE,
  to_jax = TRUE
)
```

Arguments

concentration	A positive concentration parameter analogous to the concentration of a Gamma distribution. The concentration must be larger than the dimensionality of the scale matrix.
scale_matrix	A scale matrix analogous to the inverse rate of a Gamma distribution.
rate_matrix	A rate matrix anaologous to the rate of a Gamma distribution.
scale_tril	Cholesky decomposition of the 'scale_matrix'.
validate_args	Logical: Whether to validate parameter values. Defaults to 'reticulate::py_none()'.
name	A character string representing the name of the random variable within a model. This is used to uniquely identify the variable. Defaults to 'x'.
obs	A numeric vector or array of observed values. If provided, the random variable is conditioned on these values. If 'NULL', the variable is treated as a latent (unobserved) variable. Defaults to 'NULL'.
mask	An optional boolean array to mask observations.
sample	A logical value that controls the function's behavior. If 'TRUE', the function will directly draw samples from the distribution. If 'FALSE', it will create a random variable within a model. Defaults to 'FALSE'.
seed	An integer used to set the random seed for reproducibility when 'sample = TRUE'. This argument has no effect when 'sample = FALSE', as randomness is handled by the model's inference engine. Defaults to 0.
shape	A numeric vector specifying the shape. When 'sample=False' (model building), this is used with '.expand(shape)' to set the distribution's batch shape. When 'sample=True' (direct sampling), this is used as 'sample_shape' to draw a raw JAX array of the given shape.
event	The number of batch dimensions to reinterpret as event dimensions (used in model building).
create_obj	If 'TRUE', returns the raw BI distribution object instead of creating a sample site. This is essential for building complex distributions like 'MixtureSameFamily'.
to_jax	Boolean. Indicates whether to return a JAX array or not.

Value

- When `sample=FALSE`, a BI Wishart distribution object (for model building).

- When `sample=TRUE`, a JAX array of samples drawn from the Wishart distribution (for direct sampling).

- When `create_obj=TRUE`, the raw BI distribution object (for advanced use cases).

See Also

https://num.pyro.ai/en/stable/distributions.html#wishart

Examples

```
library(BayesianInference)
m=importBI(platform='cpu')
bi.dist.wishart(concentration = 5, scale_matrix = matrix(c(1,0,0,1), nrow = 2), sample = TRUE)
```

`bi.dist.wishart_cholesky`

Wishart Cholesky Distribution

Description

* The **Wishart** distribution is a distribution over positive definite matrices, often used as a prior for covariance or precision matrices in multivariate normal models. * The **Cholesky parameterization** of the Wishart (called "wishart_cholesky" in Stan, for example) reparameterizes the Wishart over its **lower (or upper) triangular Cholesky factor**. This is useful for numerical stability and unconstrained parameterization in Bayesian sampling frameworks. * In this parameterization, one works with a lower-triangular matrix L_W such that

$$\Sigma = L_W L_W^\top$$

and imposes a density over L_W corresponding to the induced Wishart density on

$$\Sigma$$

.

The Wishart distribution is a multivariate distribution used as a prior distribution for covariance matrices. This implementation represents the distribution in terms of its Cholesky decomposition.

Usage

```
bi.dist.wishart_cholesky(
  concentration,
  scale_matrix = py_none(),
  rate_matrix = py_none(),
  scale_tril = py_none(),
```

```
  validate_args = py_none(),
  name = "x",
  obs = py_none(),
  mask = py_none(),
  sample = FALSE,
  seed = py_none(),
  shape = c(),
  event = 0,
  create_obj = FALSE,
  to_jax = TRUE
)
```

Arguments

concentration	(numeric or vector) Positive concentration parameter analogous to the concentration of a 'Gamma' distribution. The concentration must be larger than the dimensionality of the scale matrix.
scale_matrix	(numeric vector, matrix, or array, optional) Scale matrix analogous to the inverse rate of a 'Gamma' distribution. If not provided, 'rate_matrix' or 'scale_tril' must be.
rate_matrix	(numeric vector, matrix, or array, optional) Rate matrix anaologous to the rate of a 'Gamma' distribution. If not provided, 'scale_matrix' or 'scale_tril' must be.
scale_tril	(numeric vector, matrix, or array, optional) Cholesky decomposition of the 'scale_matrix'. If not provided, 'scale_matrix' or 'rate_matrix' must be.
validate_args	Logical: Whether to validate parameter values. Defaults to 'reticulate::py_none()'.
name	A character string representing the name of the random variable within a model. This is used to uniquely identify the variable. Defaults to 'x'.
obs	A numeric vector or array of observed values. If provided, the random variable is conditioned on these values. If 'NULL', the variable is treated as a latent (unobserved) variable. Defaults to 'NULL'.
mask	An optional boolean vector to mask observations.
sample	A logical value that controls the function's behavior. If 'TRUE', the function will directly draw samples from the distribution. If 'FALSE', it will create a random variable within a model. Defaults to 'FALSE'.
seed	An integer used to set the random seed for reproducibility when 'sample = TRUE'. This argument has no effect when 'sample = FALSE', as randomness is handled by the model's inference engine. Defaults to 0.
shape	A numeric vector. This is used with '.expand(shape)' when 'sample=False' (model building) to set the distribution's batch shape. When 'sample=True' (direct sampling), this is used as 'sample_shape' to draw a raw JAX array of the given shape.
event	An integer representing the number of batch dimensions to reinterpret as event dimensions (used in model building).
create_obj	A logical value. If 'TRUE', returns the raw BI distribution object instead of creating a sample site.

`to_jax` Boolean. Indicates whether to return a JAX array or not.

Value

- When `sample=FALSE`, a BI Wishart Cholesky distribution object (for model building).

- When `sample=TRUE`, a JAX array of samples drawn from the Wishart Cholesky distribution (for direct sampling).

- When `create_obj=TRUE`, the raw BI distribution object (for advanced use cases).

Examples

```
library(BayesianInference)
m=importBI(platform='cpu')
bi.dist.wishart_cholesky(
concentration = 5,
scale_matrix = matrix(c(1,0,0,1),
nrow = 2),
sample = TRUE)
```

---

bi.dist.zero_inflated_distribution

Generic Zero Inflated distribution.

Description

The Zero-Inflated Poisson distribution is a discrete count-distribution designed for data with *more zeros* than would be expected under a standard Poisson. Essentially, it assumes two underlying processes: * With probability

$$\pi$$

you are in a "structural zero" state (i.e., you automatically get a zero count). * With probability

$$1 - \pi$$

you draw from a standard Poisson distribution with parameter

$$\lambda$$

.

This results in a mixture distribution that places more mass at zero than a Poisson alone would. It's widely used in, for instance, ecology (species counts with many zeros), insurance/claims problems, and any count-data setting with excess zeros.

Usage

```
bi.dist.zero_inflated_distribution(
  base_dist,
  gate = py_none(),
  gate_logits = py_none(),
  validate_args = py_none(),
  name = "x",
  obs = py_none(),
  mask = py_none(),
  sample = FALSE,
  seed = py_none(),
  shape = c(),
  event = 0,
  create_obj = FALSE,
  to_jax = TRUE
)
```

Arguments

base_dist	Distribution: The base distribution to be zero-inflated (e.g., Poisson, Negative-Binomial).
gate	numeric(1): Probability of extra zeros (between 0 and 1).
gate_logits	numeric(1): Log-odds of extra zeros.
validate_args	Logical: Whether to validate parameter values. Defaults to 'reticulate::py_none()'.
name	A character string representing the name of the random variable within a model. This is used to uniquely identify the variable. Defaults to 'x'.
obs	A numeric vector or array of observed values. If provided, the random variable is conditioned on these values. If 'NULL', the variable is treated as a latent (unobserved) variable. Defaults to 'NULL'.
mask	logical(1): Optional boolean array to mask observations.
sample	A logical value that controls the function's behavior. If 'TRUE', the function will directly draw samples from the distribution. If 'FALSE', it will create a random variable within a model. Defaults to 'FALSE'.
seed	An integer used to set the random seed for reproducibility when 'sample = TRUE'. This argument has no effect when 'sample = FALSE', as randomness is handled by the model's inference engine. Defaults to 0.
shape	numeric(1): A multi-purpose argument for shaping. When 'sample=False' (model building), this is used with '.expand(shape)' to set the distribution's batch shape. When 'sample=True' (direct sampling), this is used as 'sample_shape' to draw a raw JAX array of the given shape. Provide as a numeric vector (e.g., 'c(10)').
event	int(1): The number of batch dimensions to reinterpret as event dimensions (used in model building).
create_obj	Logical: If True, returns the raw BI distribution object instead of creating a sample site. This is essential for building complex distributions like 'MixtureSameFamily'.
to_jax	Boolean. Indicates whether to return a JAX array or not.

Value

- When `sample=FALSE`, a BI Zero Inflated distribution object (for model building).

- When `sample=TRUE`, a JAX array of samples drawn from the Zero Inflated distribution (for direct sampling).

- When `create_obj=TRUE`, the raw BI distribution object (for advanced use cases).

See Also

https://num.pyro.ai/en/stable/distributions.html#zeroinflateddistribution

Examples

```
library(BayesianInference)
m=importBI(platform='cpu')
bi.dist.zero_inflated_distribution(
base_dist = bi.dist.poisson(5, create_obj = TRUE),
gate=0.3, sample = TRUE)
```

---

bi.dist.zero_inflated_negative_binomial

Zero-Inflated Negative Binomial Distribution

Description

A Zero-Inflated Negative Binomial distribution is used for count data that exhibit **both** (a) over-dispersion relative to a Poisson (i.e., variance > mean) *and* (b) an excess of zero counts beyond what a standard Negative Binomial would predict. It assumes two latent processes: 1. With probability

$$\pi$$

(sometimes denoted

$$\psi$$

or "zero-inflation probability") you are in a "structural zero" state ??? you observe a zero. 2. With probability

$$1 - \pi$$

, you come from a regular Negative Binomial distribution (with parameters e.g. mean

$$\mu$$

and dispersion parameter

$$\alpha$$

or size/r parameter) and then you might observe zero or a positive count. Thus the model is a mixture of a point-mass at zero + a Negative Binomial for counts. This distribution combines a Negative Binomial distribution with a binary gate variable. Observations are either drawn from the Negative Binomial distribution with probability (1 - gate) or are treated as zero with probability 'gate'.

This models data with excess zeros compared to what a standard Negative Binomial distribution would predict.

Usage

```
bi.dist.zero_inflated_negative_binomial(
  mean,
  concentration,
  gate = py_none(),
  gate_logits = py_none(),
  validate_args = py_none(),
  name = "x",
  obs = py_none(),
  mask = py_none(),
  sample = FALSE,
  seed = py_none(),
  shape = c(),
  event = 0,
  create_obj = FALSE,
  to_jax = TRUE
)
```

Arguments

mean	Numeric or a numeric vector. The mean of the Negative Binomial 2 distribution.
concentration	Numeric or a numeric vector. The concentration parameter of the Negative Binomial 2 distribution.
gate	numeric(1): Probability of extra zeros (between 0 and 1).
gate_logits	numeric(1): Log-odds of extra zeros.
validate_args	Logical: Whether to validate parameter values. Defaults to ‘reticulate::py_none()‘.
name	A character string representing the name of the random variable within a model. This is used to uniquely identify the variable. Defaults to ’x’.
obs	A numeric vector or array of observed values. If provided, the random variable is conditioned on these values. If ‘NULL‘, the variable is treated as a latent (unobserved) variable. Defaults to ‘NULL‘.
mask	Logical vector. Optional boolean array to mask observations.
sample	A logical value that controls the function’s behavior. If ‘TRUE‘, the function will directly draw samples from the distribution. If ‘FALSE‘, it will create a random variable within a model. Defaults to ‘FALSE‘.
seed	An integer used to set the random seed for reproducibility when ‘sample = TRUE‘. This argument has no effect when ‘sample = FALSE‘, as randomness is handled by the model’s inference engine. Defaults to 0.
shape	A numeric vector. A multi-purpose argument for shaping. When ‘sample=False‘ (model building), this is used with ‘.expand(shape)‘ to set the distribution’s batch shape. When ‘sample=True‘ (direct sampling), this is used as ‘sample_shape‘ to draw a raw JAX array of the given shape.
event	Integer. The number of batch dimensions to reinterpret as event dimensions (used in model building).

create_obj	Logical. If 'TRUE', returns the raw NumPyro distribution object instead of creating a sample site. This is essential for building complex distributions like 'MixtureSameFamily'.
to_jax	Boolean. Indicates whether to return a JAX array or not.

Value

- When sample=FALSE, a BI Zero-Inflated Negative Binomial distribution object (for model building).

- When sample=TRUE, a JAX array of samples drawn from the Zero-Inflated Negative Binomial distribution (for direct sampling).

- When create_obj=TRUE, the raw BI distribution object (for advanced use cases).

Examples

```
library(BayesianInference)
m <- importBI(platform = "cpu")
bi.dist.zero_inflated_negative_binomial(mean = 2, concentration = 1, gate = 0.3, sample = TRUE)
```

---

bi.dist.zero_inflated_poisson

A Zero Inflated Poisson distribution.

Description

The Zero-Inflated Poisson distribution is a discrete count-distribution designed for data with *more zeros* than would be expected under a standard Poisson. Essentially, it assumes two underlying processes: * With probability

$$\pi$$

you are in a "structural zero" state (i.e., you automatically get a zero count). * With probability

$$1 - \pi$$

you draw from a standard Poisson distribution with parameter

$$\lambda$$

.

This results in a mixture distribution that places more mass at zero than a Poisson alone would. It's widely used in, for instance, ecology (species counts with many zeros), insurance/claims problems, and any count-data setting with excess zeros.

Usage

```
bi.dist.zero_inflated_poisson(
  gate,
  rate = 1,
  validate_args = py_none(),
  name = "x",
  obs = py_none(),
  mask = py_none(),
  sample = FALSE,
  seed = py_none(),
  shape = c(),
  event = 0,
  create_obj = FALSE,
  to_jax = TRUE
)
```

Arguments

gate	The gate parameter.
rate	A numeric vector, matrix, or array representing the rate parameter of the underlying Poisson distribution.
validate_args	Logical: Whether to validate parameter values. Defaults to 'reticulate::py_none()'.
name	A character string representing the name of the random variable within a model. This is used to uniquely identify the variable. Defaults to 'x'.
obs	A numeric vector or array of observed values. If provided, the random variable is conditioned on these values. If 'NULL', the variable is treated as a latent (unobserved) variable. Defaults to 'NULL'.
mask	An optional boolean vector, matrix, or array to mask observations.
sample	A logical value that controls the function's behavior. If 'TRUE', the function will directly draw samples from the distribution. If 'FALSE', it will create a random variable within a model. Defaults to 'FALSE'.
seed	An integer used to set the random seed for reproducibility when 'sample = TRUE'. This argument has no effect when 'sample = FALSE', as randomness is handled by the model's inference engine. Defaults to 0.
shape	A numeric vector used to shape the distribution. When 'sample=False' (model building), this is used with '.expand(shape)' to set the distribution's batch shape. When 'sample=True' (direct sampling), this is used as 'sample_shape' to draw a raw JAX array of the given shape.
event	The number of batch dimensions to reinterpret as event dimensions (used in model building).
create_obj	Logical; If 'TRUE', returns the raw BI distribution object instead of creating a sample site.
to_jax	Logical. Defaults to TRUE.

Value

- When `sample=FALSE`, a BI Zero Inflated Poisson distribution object (for model building).
- When `sample=TRUE`, a JAX array of samples drawn from the Zero Inflated Poisson distribution (for direct sampling).
- When `create_obj=TRUE`, the raw BI distribution object (for advanced use cases).

Examples

```
library(BayesianInference)
m <- importBI(platform = "cpu")
bi.dist.zero_inflated_poisson(gate = 0.3, rate = 5, sample = TRUE)
```

---

bi.dist.zero_sum_normal

zero_sum_normal

Description

A **zero-sum normal** is a variant of a multivariate normal in which one (or more) linear constraint(s) force certain components to **sum to zero**. In practice, it's used to model vectors of random effects (e.g. in hierarchical models) where the effects are constrained to sum to zero (to avoid overparameterization or enforce identifiability).

Usage

```
bi.dist.zero_sum_normal(
  scale,
  event_shape,
  validate_args = py_none(),
  name = "x",
  obs = py_none(),
  mask = py_none(),
  sample = FALSE,
  seed = py_none(),
  shape = c(),
  event = 0,
  create_obj = FALSE,
  to_jax = TRUE
)
```

Arguments

`scale`	A numeric vector or array representing the standard deviation of the underlying normal distribution before the zerosum constraint is enforced.
`event_shape`	(numeric vector): The shape of the event, defining the dimensions of the vector that will be constrained to sum to zero.

validate_args	Logical: Whether to validate parameter values. Defaults to 'reticulate::py_none()'.
name	A character string representing the name of the random variable within a model. This is used to uniquely identify the variable. Defaults to 'x'.
obs	A numeric vector or array of observed values. If provided, the random variable is conditioned on these values. If 'NULL', the variable is treated as a latent (unobserved) variable. Defaults to 'NULL'.
mask	A logical vector or array. Optional boolean array to mask observations.
sample	A logical value that controls the function's behavior. If 'TRUE', the function will directly draw samples from the distribution. If 'FALSE', it will create a random variable within a model. Defaults to 'FALSE'.
seed	An integer used to set the random seed for reproducibility when 'sample = TRUE'. This argument has no effect when 'sample = FALSE', as randomness is handled by the model's inference engine. Defaults to 0.
shape	A numeric vector specifying the shape of the distribution. When 'sample=False' (model building), this is used with '.expand(shape)' to set the distribution's batch shape. When 'sample=True' (direct sampling), this is used as 'sample_shape' to draw a raw JAX array of the given shape.
event	An integer representing the number of batch dimensions to reinterpret as event dimensions (used in model building).
create_obj	A logical value. If 'TRUE', returns the raw BI distribution object instead of creating a sample site. This is essential for building complex distributions like 'MixtureSameFamily'.
to_jax	Boolean. Indicates whether to return a JAX array or not.

Value

- When `sample=FALSE`, a BI zero_sum_normal distribution object (for model building).

- When `sample=TRUE`, a JAX array of samples drawn from the zero_sum_normal distribution (for direct sampling).

- When `create_obj=TRUE`, the raw BI distribution object (for advanced use cases).

See Also

https://num.pyro.ai/en/stable/distributions.html#zerosumnormal

Examples

```
library(BayesianInference)
m=importBI(platform='cpu')
bi.dist.zero_sum_normal(scale=0.3, event_shape = c(), sample = TRUE)
```

`bi.doc` *title Open the BI Documentation Website*

Description

This function opens the BI documentation webpage in the user's default web browser.

Usage

```
bi.doc()
```

Value

Invisibly returns 'NULL'. The function is called for its side effect of opening a URL.

`importBI` *Import the BI Python Module*

Description

This function initializes the BI Python module through **reticulate**, sets up the environment, and loads the necessary 'jax' and 'jax.numpy' modules. The BI module is stored in the hidden object '.bi' for internal use, but the initialized BI object is also returned for convenience.

Usage

```
importBI(
  platform = "cpu",
  cores = NULL,
  rand_seed = TRUE,
  deallocate = FALSE,
  print_devices_found = TRUE,
  backend = "numpyro"
)
```

Arguments

`platform`	Character string, the computational platform to use (e.g. '"cpu"' or '"gpu"'). Defaults to '"cpu"'.
`cores`	Integer or 'NULL'. Number of CPU cores to use. Defaults to 'NULL'.
`rand_seed`	(Boolean): Random seed. Defaults to TRUE.
`deallocate`	Logical. Whether memory should be deallocated when not in use. Defaults to 'FALSE'.
`print_devices_found`	(bool, optional): Whether to print devices found. Defaults to TRUE.
`backend`	(str, optional): Backend to use (numpyro or tfp). Defaults to 'numpyro'.

Details

- Internally, this function imports the ‘BI‘ Python package and assigns it to the hidden variable ‘.bi‘. - It also imports ‘jax‘ and ‘jax.numpy‘, assigning them to ‘jax‘ and ‘jnp‘ respectively. - Startup messages inform the user about the imports.

Value

An initialized BI module object (Python object via **reticulate**).

Examples

```
library(BayesianInference)
m <- importBI()
```

---

setup_env *Create a Python virtual environment*

Description

This function creates a Python virtual environment using reticulate.

Usage

```
setup_env(env_name = "BayesInference", backend = "cpu")
```

Arguments

env_name	A character string for the name of the virtual environment. Defaults to "BayesInference-cpu" or "BayesInference-gpu".
backend	A character string specifying the backend, either "cpu" or "gpu".

Value

The path to the created virtual environment.

Index

bi.dist.asymmetric_laplace, 3
bi.dist.asymmetric_laplace_quantile, 5
bi.dist.bernoulli, 7
bi.dist.beta, 9
bi.dist.beta_binomial, 11
bi.dist.beta_proportion, 12
bi.dist.binomial, 14
bi.dist.car, 16
bi.dist.categorical, 18
bi.dist.cauchy, 19
bi.dist.chi2, 21
bi.dist.delta, 23
bi.dist.dirichlet, 24
bi.dist.dirichlet_multinomial, 26
bi.dist.discrete_uniform, 27
bi.dist.euler_maruyama, 29
bi.dist.exponential, 31
bi.dist.gamma, 32
bi.dist.gamma_poisson, 34
bi.dist.gaussian_copula, 36
bi.dist.gaussian_copula_beta, 37
bi.dist.gaussian_random_walk, 39
bi.dist.gaussian_state_space, 41
bi.dist.geometric, 43
bi.dist.gompertz, 44
bi.dist.gumbel, 46
bi.dist.half_cauchy, 47
bi.dist.half_normal, 49
bi.dist.inverse_gamma, 50
bi.dist.kumaraswamy, 52
bi.dist.laplace, 54
bi.dist.left_truncated_distribution, 55
bi.dist.levy, 57
bi.dist.lkj, 59
bi.dist.lkj_cholesky, 61
bi.dist.log_normal, 63
bi.dist.log_uniform, 64
bi.dist.logistic, 66
bi.dist.low_rank_multivariate_normal, 68
bi.dist.lower_truncated_power_law, 70
bi.dist.matrix_normal, 71
bi.dist.mixture, 73
bi.dist.mixture_general, 75
bi.dist.mixture_same_family, 77
bi.dist.multinomial, 79
bi.dist.multinomial_logits, 80
bi.dist.multinomial_probs, 82
bi.dist.multivariate_normal, 84
bi.dist.multivariate_student_t, 86
bi.dist.negative_binomial2, 91
bi.dist.negative_binomial_logits, 88
bi.dist.negative_binomial_probs, 89
bi.dist.normal, 93
bi.dist.ordered_logistic, 94
bi.dist.pareto, 96
bi.dist.poisson, 98
bi.dist.projected_normal, 100
bi.dist.relaxed_bernoulli, 102
bi.dist.relaxed_bernoulli_logits, 103
bi.dist.right_truncated_distribution, 105
bi.dist.soft_laplace, 107
bi.dist.student_t, 108
bi.dist.truncated_cauchy, 110
bi.dist.truncated_distribution, 112
bi.dist.truncated_normal, 114
bi.dist.truncated_polya_gamma, 116
bi.dist.two_sided_truncated_distribution, 117
bi.dist.uniform, 119
bi.dist.weibull, 121
bi.dist.wishart, 122
bi.dist.wishart_cholesky, 124
bi.dist.zero_inflated_distribution, 126
bi.dist.zero_inflated_negative_binomial,

128
bi.dist.zero_inflated_poisson, 130
bi.dist.zero_sum_normal, 132
bi.doc, 134

importBI, 134

setup_env, 135