

Package ‘rcppbugs’

February 20, 2015

Title R binding for cppbugs

Version 0.1.4.1

Maintainer Whit Armstrong <armstrong.whit@gmail.com>

Author Whit Armstrong <armstrong.whit@gmail.com>

Description R binding for cppbugs

License GPL-3

URL <http://github.com/armstrtw/rcppbugs>

BugReports <http://github.com/armstrtw/rcppbugs/issues>

Depends Rcpp (>= 0.11)

LinkingTo Rcpp, RcppArmadillo, BH

NeedsCompilation yes

Repository CRAN

Date/Publication 2014-05-02 09:39:27

R topics documented:

deterministic	2
logp	3
mcmc.distributions	4
run.model	5
Index	7

deterministic *create deterministic rcppbugs objects*

Description

Deterministic objects determine their values from other nodes in the mcmc model. They have no contribution to the overall likelihood of the model, but are frequently used as hyperparameters of stochastic objects.

Usage

```
deterministic(f, ...)  
linear(X,b)  
linear.grouped(X,b,group)
```

Arguments

f	a user provided function that updates the value of the variable
X	the data matrix X to be used in estimation
b	the coefficient vector b to be used in estimation
group	an integer vector equal in length to the rows of X indicating the group membership of the corresponding row
...	arguments to function f

Details

The number of arguments must match the arity of function f.

Value

an rcppbugs object representing the deterministic object.

Author(s)

rcppbugs was written by Whit Armstrong.

References

<https://github.com/armstrtw/CppBugs>

See Also

[logp](#), [run.model](#)

Examples

```
NR <- 1e2L
NC <- 2L
y <- matrix(rnorm(NR,1),nr=NR,nc=1L)
X <- matrix(nr=NR,nc=NC)

b <- mcmc.normal(rnorm(NC),mu=0,tau=0.0001)
y.hat.user <- deterministic(function(X,b) { X %*% b }, X, b)
y.hat.cpp <- linear(X,b)
```

logp

Calculate the log likelihood of an rcppbugs object.

Description

Calculate the log likelihood of an rcppbugs object based on the current state of the dependant(upstream) objects in the chain.

Usage

```
logp(x)
```

Arguments

x any stochastic rcppbugs object

Value

a double precision value indicating the total log likelihood of the object.

Author(s)

rcppbugs was written by Whit Armstrong.

References

<https://github.com/armstrtw/CppBugs>

See Also

[mcmc.normal](#)

Examples

```

NR <- 100
NC <- 2

b <- mcmc.normal(rnorm(NC),mu=0,tau=0.0001)
tau.y <- mcmc.gamma(runif(1),alpha=0.1,beta=0.1)

logp(b)
logp(tau.y)

```

mcmc.distributions *Create a stochastic rcppbugs object.*

Description

Create stochastic objects in the spirit of PyMC.

Usage

```

mcmc.normal(x, mu, tau, observed = FALSE)
mcmc.uniform(x, lower, upper, observed = FALSE)
mcmc.gamma(x, alpha, beta, observed = FALSE)
mcmc.beta(x, alpha, beta, observed = FALSE)
mcmc.bernoulli(x, p, observed = FALSE)
mcmc.binomial(x, n, p, observed = FALSE)

```

Arguments

x	the initial value of the object
mu	the mean for normally distributed objects
tau	the precision of normally distributed objects
lower	the lower limit of the uniform distribution
upper	the upper limit of the uniform distribution
alpha	the shape parameter of the gamma distribution
beta	the scale parameter of the gamma distribution
n	the sample size of a binomial distribution
p	the success rate in bernoulli or binomial distributions
observed	whether the object should be treated as constant data or simulated over the MCMC chain

Value

an rcppbugs object corresponding to the particular distribution requested

Author(s)

rcppbugs was written by Whit Armstrong.

References

<https://github.com/armstrtw/CppBugs>

See Also

[logp](#)

Examples

```
b <- mcmc.normal(rnorm(10),mu=0,tau=0.0001)
tau <- mcmc.gamma(runif(1),alpha=0.1,beta=0.1)
b.unif <- mcmc.uniform(runif(1),lower=0,upper=100)
```

run.model

Create and run rcppbugs models.

Description

The create.model function creates a model object. The run.model function simulates an mcmc chain of the nodes in the model. After the completion of run.model, the history of the run is returned as a named list.

Usage

```
create.model(...)
run.model(m, iterations, burn, adapt, thin)
get.ar(x)
```

Arguments

m	the rcppbugs model object.
iterations	how many iterations to sample.
burn	how many iterations to use for burnin.
adapt	how many iterations to use for the adaptive period.
thin	how frequently to record traces of the model nodes.
...	rcppbugs objects to use as the nodes of the model.
x	the result of an rcppbugs run.

Value

create.model returns a mcmc.model model object. run.model returns a named list containing the historical traces of the model run. get.ar returns the acceptance ratio of an MCMC run

Author(s)

rcppbugs was written by Whit Armstrong.

References

<https://github.com/armstrtw/CppBugs>

See Also

[logp](#)

Examples

```
library(rcppbugs)

NR <- 1e2L
NC <- 2L
y <- matrix(rnorm(NR,1) + 10,nr=NR,nc=1L)
X <- matrix(nr=NR,nc=NC)
X[,1] <- 1
X[,2] <- y + rnorm(NR)/2 - 10

## RCppBugs Model
b <- mcmc.normal(rnorm(NC),mu=0,tau=0.0001)
tau.y <- mcmc.gamma(sd(as.vector(y)),alpha=0.1,beta=0.1)
y.hat <- linear(X,b)
y.lik <- mcmc.normal(y,mu=y.hat,tau=tau.y,observed=TRUE)
m <- create.model(b, tau.y, y.hat, y.lik)

runtime <- system.time(ans <- run.model(m, iterations=1e2L, burn=1e2L, adapt=1e3L, thin=10L))
print(get.ar(ans))

print(apply(ans[["b"]],2,mean))
```

Index

*Topic **models**

- deterministic, 2
- logp, 3
- mcmc.distributions, 4
- run.model, 5

create.model (run.model), 5

deterministic, 2

get.ar (run.model), 5

linear (deterministic), 2

logistic (deterministic), 2

logp, 2, 3, 5, 6

mcmc.bernoulli (mcmc.distributions), 4

mcmc.beta (mcmc.distributions), 4

mcmc.binomial (mcmc.distributions), 4

mcmc.distributions, 4

mcmc.gamma (mcmc.distributions), 4

mcmc.normal, 3

mcmc.normal (mcmc.distributions), 4

mcmc.uniform (mcmc.distributions), 4

run.model, 2, 5