

Package ‘twoe’

February 20, 2013

Type Package

Title Software for modelling tropical forest dynamics

Version 1.0

Date 2011-09-09

Depends coda

Author Ghislain Vieilledent <ghislain.vieilledent@cirad.fr>

Maintainer Ghislain Vieilledent <ghislain.vieilledent@cirad.fr>

Description Software for modeling tropical forest dynamics

License GPL (>= 2)

URL <http://twoe.org>

LazyLoad yes

R topics documented:

twoe-package	2
entry_data	2
entry_growth	3
entry_growth_gibbs	4
entry_mortality	8
entry_mortality_gibbs	9
entry_ngrowth	14
entry_ngrowth_gibbs	15
entry_recruitment	19
entry_recruitment_gibbs	20
exit_simu	24
InvGamma	25
InvWishart	27
logit	28
surface_comp_index	29
Wishart	30

Index**31**

twoe-package	<i>Software for modelling tropical forest dynamics</i>
--------------	--

Description

"twoe" is a software which aims first at estimating demographic parameters for tropical tree species from permanent forest plot data (through the "entry" module) and second at simulating forest dynamics (through the "exit" module).

Details

Package:	twoe
Type:	Package
Version:	1.0
Date:	2010-11-08
License:	GPL
LazyLoad:	yes

Author(s)

Ghislain Vieilledent <ghislain.vieilledent@cirad.fr>

entry_data	<i>Function to transform the permanent plot data-set into right format data-sets for the growth, mortality and recruitment model.</i>
------------	---

Description

Function to transform the permanent plot data-set into right format data-sets for the growth, mortality and recruitment model.

Usage

```
entry_data(Data, XY.Plot, R.Comp, L.Cell)
```

Arguments

Data	Data frame of the permanent plot data-set.
XY.Plot	Data frame with the dimensions (m) of each plot: Plot X.Plot Y.Plot
R.Comp	radius (m) for the competition area around each tree
L.Cell	width (m) of the ground cell for recruitment estimation

Value

Return three data-sets: data_growth.txt, data_mortality.txt and data_recruitment.txt

Author(s)

Ghislain Vieilledent <ghislain.vieilledent@cirad.fr>

entry_growth	<i>A function to estimate the parameters of the growth model for each species in the data-set.</i>
--------------	--

Description

A function to estimate the parameters of the growth model for each species in the data-set. The function prints summaries and plots diagnostic graphics.

Usage

```
entry_growth(data_growth, burnin=1000, mcmc=1000, thin=1, th.sign=5)
```

Arguments

data_growth	data frame which must have the following column names: Tree Sp D_t C_t G_tpl
burnin	The number of burnin iterations for the sampler.
mcmc	The number of Gibbs iterations for the sampler. Total number of Gibbs iterations is equal to burnin+mcmc. burnin+mcmc must be divisible by 10 and superior or equal to 100 so that the progress bar can be displayed.
thin	The thinning interval used in the simulation. The number of mcmc iterations must be divisible by this value.
th.sign	Threshold (in %) used to test parameter significance: is zero included in the interval defined by the threshold? Default to 5.

Value

The function prints summaries and plots diagnostic graphics.

Author(s)

Ghislain Vieilledent <ghislain.vieilledent@cirad.fr>

entry_growth_gibbs	<i>Markov Chain Monte Carlo for the Hierarchical Gaussian Linear Regression Model</i>
--------------------	---

Description

entry_growth_gibbs generates a sample from the posterior distribution of a Hierarchical Gaussian Linear Regression Model using Algorithm 2 of Chib and Carlin (1999). This model uses a multi-variate Normal prior for the fixed effects parameters, an Inverse-Wishart prior on the random effects variance matrix, and an Inverse-Gamma prior on the residual error variance. The user supplies data and priors, and a sample from the posterior distribution is returned as an mcmc object, which can be subsequently analyzed with functions provided in the coda package.

Usage

```
entry_growth_gibbs(fixed, random, group, data, burnin=1000,
  mcmc=10000, thin=10, verbose=1, seed=NA, beta.start=NA, sigma2.start=NA,
  Vb.start=NA, mubeta=0, Vbeta=1.0E6, r, R, nu=0.001, delta=0.001, ...)
```

Arguments

fixed	A two-sided linear formula of the form 'y~x1+...+xp' describing the fixed-effects part of the model, with the response on the left of a '~' operator and the p fixed terms, separated by '+' operators, on the right.
random	A one-sided formula of the form '~x1+...+xq' specifying the model for the random effects part of the model, with the q random terms, separated by '+' operators.
group	String indicating the name of the grouping variable in data, defining the hierarchical structure of the model.
data	A data frame containing the variables in the model.
burnin	The number of burnin iterations for the sampler.
mcmc	The number of Gibbs iterations for the sampler. Total number of Gibbs iterations is equal to burnin+mcmc. burnin+mcmc must be divisible by 10 and superior or equal to 100 so that the progress bar can be displayed.
thin	The thinning interval used in the simulation. The number of mcmc iterations must be divisible by this value.
seed	The seed for the random number generator. If NA, the Mersenne Twister generator is used with default seed 12345; if an integer is passed it is used to seed the Mersenne twister.
verbose	A switch (0,1) which determines whether or not the progress of the sampler is printed to the screen. Default is 1: a progress bar is printed, indicating the step (in %) reached by the Gibbs sampler.

beta.start	The starting values for the β vector. This can either be a scalar or a p-length vector. The default value of NA will use the OLS β estimate of the corresponding Gaussian Linear Regression without random effects. If this is a scalar, that value will serve as the starting value mean for all of the betas.
sigma2.start	Scalar for the starting value of the residual error variance. The default value of NA will use the OLS estimates of the corresponding Gaussian Linear Regression without random effects.
Vb.start	The starting value for variance matrix of the random effects. This must be a square q-dimension matrix. Default value of NA uses an identity matrix.
mubeta	The prior mean of β . This can either be a scalar or a p-length vector. If this takes a scalar value, then that value will serve as the prior mean for all of the betas. The default value of 0 will use a vector of zeros for an uninformative prior.
Vbeta	The prior variance of β . This can either be a scalar or a square p-dimension matrix. If this takes a scalar value, then that value times an identity matrix serves as the prior variance of beta. Default value of 1.0E6 will use a diagonal matrix with very large variance for an uninformative flat prior.
r	The shape parameter for the Inverse-Wishart prior on variance matrix for the random effects. r must be superior or equal to q. Set r=q for an uninformative prior. See the NOTE for more details
R	The scale matrix for the Inverse-Wishart prior on variance matrix for the random effects. This must be a square q-dimension matrix. Use plausible variance regarding random effects for the diagonal of R. See the NOTE for more details
nu	The shape parameter for the Inverse-Gamma prior on the residual error variance. Default value is nu=delta=0.001 for uninformative prior.
delta	The rate (1/scale) parameter for the Inverse-Gamma prior on the residual error variance. Default value is nu=delta=0.001 for uninformative prior.
...	further arguments to be passed

Details

entry_growth_gibbs simulates from the posterior distribution sample using the blocked Gibbs sampler of Chib and Carlin (1999), Algorithm 2. The simulation is done in compiled C++ code to maximize efficiency. Please consult the coda documentation for a comprehensive list of functions that can be used to analyze the posterior sample.

The model takes the following form:

$$y_i = X_i\beta + W_ib_i + \varepsilon_i$$

Where each group i have k_i observations.

Where the random effects:

$$b_i \sim \mathcal{N}_q(0, V_b)$$

And the errors:

$$\varepsilon_i \sim \mathcal{N}(0, \sigma^2 I_{k_i})$$

We assume standard, conjugate priors:

$$\beta \sim \mathcal{N}_p(\mu_\beta, V_\beta)$$

And:

$$\sigma^2 \sim \mathcal{IGamma}(\nu, 1/\delta)$$

And:

$$V_b \sim \mathcal{IWishart}(r, rR)$$

See Chib and Carlin (1999) for more details.

NOTE: We do not provide default parameters for the priors on the precision matrix for the random effects. When fitting one of these models, it is of utmost importance to choose a prior that reflects your prior beliefs about the random effects. Using the `dwish` and `rwish` functions might be useful in choosing these values.

Value

<code>mcmc</code>	An <code>mcmc</code> object that contains the posterior sample. This object can be summarized by functions provided by the <code>coda</code> package. The posterior sample of the deviance D , with $D = -2 \log(\prod_i P(y_i \beta, b_i, \sigma^2))$, is also provided.
<code>Y.pred</code>	Predictive posterior mean for each observation.
<code>pD</code>	Effective number of the parameters of the model.
<code>DIC</code>	Deviance Information Criterion.

Author(s)

Ghislain Vieilledent <ghislain.vieilledent@cirad.fr>

References

- Siddhartha Chib and Bradley P. Carlin. 1999. “On MCMC Sampling in Hierarchical Longitudinal Models.” *Statistics and Computing*. 9: 17-26.
- Daniel Pemstein, Kevin M. Quinn, and Andrew D. Martin. 2007. *Scythe Statistical Library 1.0*. <http://scythe.wustl.edu>.
- Andrew D. Martin and Kyle L. Saunders. 2002. “Bayesian Inference for Political Science Panel Data.” Paper presented at the 2002 Annual Meeting of the American Political Science Association.
- Martyn Plummer, Nicky Best, Kate Cowles, and Karen Vines. 2002. *Output Analysis and Diagnostics for MCMC (CODA)*. <http://www-fis.iarc.fr/coda/>.

See Also

[plot.mcmc](#), [summary.mcmc](#)

Examples

```
## Not run:
#=====
# Hierarchical Gaussian Linear Regression
#=====

#== Generating data
```

```

# Constants
nobs <- 1000
nspecies <- 20
species <- c(1:nspecies,sample(c(1:nspecies),(nobs-nspecies),replace=TRUE))

# Covariates
X1 <- runif(n=nobs,min=0,max=10)
X2 <- runif(n=nobs,min=0,max=10)
X <- cbind(rep(1,nobs),X1,X2)
W <- X

# Target parameters
# beta
beta.target <- matrix(c(0.1,0.3,0.2),ncol=1)
# Vb
Vb.target <- c(0.5,0.2,0.1)
# b
b.target <- cbind(rnorm(nspecies,mean=0,sd=sqrt(Vb.target[1])),
                  rnorm(nspecies,mean=0,sd=sqrt(Vb.target[2])),
                  rnorm(nspecies,mean=0,sd=sqrt(Vb.target[3])))
# sigma2
sigma2.target <- 0.02

# Response
Y <- vector()
for (n in 1:nobs) {
  Y[n] <- rnorm(n=1,
               mean=X[n,]%*%beta.target+W[n,]%*%b.target[species[n],],
               sd=sqrt(sigma2.target))
}

# Data-set
Data <- as.data.frame(cbind(Y,X1,X2,species))
plot(Data$X1,Data$Y)

#== Call to entry_growth_gibbs
model <- entry_growth_gibbs(fixed=Y~X1+X2, random=~X1+X2, group="species",
                           data=Data, burnin=1000, mcmc=1000, thin=1,verbose=1,
                           seed=NA, beta.start=0, sigma2.start=1,
                           Vb.start=1, mubeta=0, Vbeta=1.0E6,
                           r=3, R=diag(c(1,0.1,0.1)), nu=0.001, delta=0.001)

#== MCMC analysis

# Graphics
pdf("Posteriors-entry_growth_gibbs.pdf")
plot(model$mcmc)
dev.off()

# Summary
summary(model$mcmc)

```

```
# Predictive posterior mean for each observation
model$Y.pred

# Predicted-Observed
plot(Data$Y,model$Y.pred)
abline(a=0,b=1)

## End(Not run)
```

entry_mortality	<i>A function to estimate the parameters of the mortality model for each species in the data-set.</i>
-----------------	---

Description

A function to estimate the parameters of the mortality model for each species in the data-set. The function prints summaries and plots diagnostic graphics.

Usage

```
entry_mortality(data_mortality, burnin=5000, mcmc=5000, thin=5, th.sign=5)
```

Arguments

data_mortality	data frame which must have the following column names: Tree Sp D_t C_t status_tp1 interval_tp1
burnin	The number of burnin iterations for the sampler.
mcmc	The number of Gibbs iterations for the sampler. Total number of Gibbs iterations is equal to burnin+mcmc. burnin+mcmc must be divisible by 10 and superior or equal to 100 so that the progress bar can be displayed.
thin	The thinning interval used in the simulation. The number of mcmc iterations must be divisible by this value.
th.sign	Threshold (in %) used to test parameter significance: is zero included in the interval defined by the threshold? Default to 5.

Value

The function prints summaries and plots diagnostic graphics.

Author(s)

Ghislain Vieilledent <ghislain.vieilledent@cirad.fr>

entry_mortality_gibbs *Markov Chain Monte Carlo for the Hierarchical Binomial Linear Regression Model using the logit link function*

Description

entry_mortality_gibbs generates a sample from the posterior distribution of a Hierarchical Binomial Linear Regression Model using the logit link function and Algorithm 2 of Chib and Carlin (1999). This model uses a multivariate Normal prior for the fixed effects parameters, an Inverse-Wishart prior on the random effects variance matrix, and an Inverse-Gamma prior on the variance modelling over-dispersion. The user supplies data and priors, and a sample from the posterior distribution is returned as an mcmc object, which can be subsequently analyzed with functions provided in the coda package.

Usage

```
entry_mortality_gibbs(fixed, random, group, interval=1, data,
  burnin=5000, mcmc=10000, thin=10, verbose=1, seed=NA, beta.start=NA,
  sigma2.start=NA, Vb.start=NA, mubeta=0, Vbeta=1.0E6, r, R, nu=0.001,
  delta=0.001, FixOD=0, ...)
```

Arguments

fixed	A two-sided linear formula of the form 'y~x1+...+xp' describing the fixed-effects part of the model, with the response on the left of a '~' operator and the p fixed terms, separated by '+' operators, on the right. Response variable y must be 0 or 1 (Binomial process).
random	A one-sided formula of the form '~x1+...+xq' specifying the model for the random effects part of the model, with the q random terms, separated by '+' operators.
group	String indicating the name of the grouping variable in data, defining the hierarchical structure of the model.
interval	A numeric scalar or vector of length equal to the number of observations. interval specifies the time interval between censuses for the observation of mortality. Default to 1.
data	A data frame containing the variables in the model.
burnin	The number of burnin iterations for the sampler.
mcmc	The number of Gibbs iterations for the sampler. Total number of Gibbs iterations is equal to burnin+mcmc. burnin+mcmc must be divisible by 10 and superior or equal to 100 so that the progress bar can be displayed.
thin	The thinning interval used in the simulation. The number of mcmc iterations must be divisible by this value.
seed	The seed for the random number generator. If NA, the Mersenne Twister generator is used with default seed 12345; if an integer is passed it is used to seed the Mersenne twister.

verbose	A switch (0,1) which determines whether or not the progress of the sampler is printed to the screen. Default is 1: a progress bar is printed, indicating the step (in %) reached by the Gibbs sampler.
beta.start	The starting values for the β vector. This can either be a scalar or a p-length vector. The default value of NA will use the OLS β estimate of the corresponding Gaussian Linear Regression without random effects. If this is a scalar, that value will serve as the starting value mean for all of the betas.
sigma2.start	Scalar for the starting value of the residual error variance. The default value of NA will use the OLS estimates of the corresponding Gaussian Linear Regression without random effects.
Vb.start	The starting value for variance matrix of the random effects. This must be a square q-dimension matrix. Default value of NA uses an identity matrix.
mubeta	The prior mean of β . This can either be a scalar or a p-length vector. If this takes a scalar value, then that value will serve as the prior mean for all of the betas. The default value of 0 will use a vector of zeros for an uninformative prior.
Vbeta	The prior variance of β . This can either be a scalar or a square p-dimension matrix. If this takes a scalar value, then that value times an identity matrix serves as the prior variance of beta. Default value of 1.0E6 will use a diagonal matrix with very large variance for an uninformative flat prior.
r	The shape parameter for the Inverse-Wishart prior on variance matrix for the random effects. r must be superior or equal to q. Set r=q for an uninformative prior. See the NOTE for more details
R	The scale matrix for the Inverse-Wishart prior on variance matrix for the random effects. This must be a square q-dimension matrix. Use plausible variance regarding random effects for the diagonal of R. See the NOTE for more details
nu	The shape parameter for the Inverse-Gamma prior on the residual error variance. Default value is nu=delta=0.001 for uninformative prior.
delta	The rate (1/scale) parameter for the Inverse-Gamma prior on the residual error variance. Default value is nu=delta=0.001 for uninformative prior.
FixOD	A switch (0,1) which determines whether or not the variance for over-dispersion (sigma2) should be fixed (1) or not (0). Default is 0, parameter sigma2 is estimated. If FixOD=1, sigma2 is fixed to the value provided for sigma2.start.
...	further arguments to be passed

Details

entry_mortality_gibbs simulates from the posterior distribution sample using the blocked Gibbs sampler of Chib and Carlin (1999), Algorithm 2. The simulation is done in compiled C++ code to maximize efficiency. Please consult the coda documentation for a comprehensive list of functions that can be used to analyze the posterior sample.

The model takes the following form:

$$y_i \sim \text{Bernoulli}(\theta_i)$$

With $\theta'_i = 1 - (1 - \theta_i)^{I_i}$, where I_i stands for the time interval between censuses for the observation of the mortality for observation i . Thus, θ_i is the "mortality rate": the probability to die by unit of time.

With latent variables $\phi(\theta_i)$, ϕ being the logit link function:

$$\phi(\theta_i) = X_i\beta + W_ib_i + \varepsilon_i$$

Where each group i have k_i observations.

Where the random effects:

$$b_i \sim \mathcal{N}(0, V_b)$$

And the over-dispersion terms:

$$\varepsilon_i \sim \mathcal{N}(0, \sigma^2 I_{k_i})$$

We assume standard, conjugate priors:

$$\beta \sim \mathcal{N}_p(\mu_\beta, V_\beta)$$

And:

$$\sigma^2 \sim \mathcal{IGamma}(\nu, 1/\delta)$$

And:

$$V_b \sim \mathcal{IWishart}(r, rR)$$

See Chib and Carlin (1999) for more details.

NOTE: We do not provide default parameters for the priors on the precision matrix for the random effects. When fitting one of these models, it is of utmost importance to choose a prior that reflects your prior beliefs about the random effects. Using the `dwish` and `rwish` functions might be useful in choosing these values.

Value

mcmc	An mcmc object that contains the posterior sample. This object can be summarized by functions provided by the coda package. The posterior sample of the deviance D , with $D = -2 \log(\prod_i P(y_i \theta_i))$, is also provided.
theta.pred	Predictive posterior mean for the inverse-logit of the latent variables. The approximation of Diggle et al. (2004) is used to marginalized with respect to over-dispersion terms:

$$E[\theta_i|\beta, b_i, \sigma^2] = \phi^{-1}((X_i\beta + W_ib_i)/\sqrt{(16\sqrt{3}/15\pi)^2\sigma^2 + 1})$$

Author(s)

Ghislain Vieilledent <ghislain.vieilledent@cirad.fr>

References

- Siddhartha Chib and Bradley P. Carlin. 1999. “On MCMC Sampling in Hierarchical Longitudinal Models.” *Statistics and Computing*. 9: 17-26.
- Daniel Pemstein, Kevin M. Quinn, and Andrew D. Martin. 2007. *Scythe Statistical Library 1.0*. <http://scythe.wustl.edu>.
- Andrew D. Martin and Kyle L. Saunders. 2002. “Bayesian Inference for Political Science Panel Data.” Paper presented at the 2002 Annual Meeting of the American Political Science Association.
- Martyn Plummer, Nicky Best, Kate Cowles, and Karen Vines. 2002. *Output Analysis and Diagnostics for MCMC (CODA)*. <http://www-fis.iarc.fr/coda/>.
- Diggle P., Heagerty P., Liang K., and Zeger S. 2004. “Analysis of Longitudinal Data.” *Oxford University Press*, 2nd Edition.

See Also

[plot.mcmc](#), [summary.mcmc](#)

Examples

```
## Not run:
#####
# Hierarchical Binomial Linear Regression
#####

#== inv.logit function
inv.logit <- function(x, min=0, max=1) {
  p <- exp(x)/(1+exp(x))
  p <- ifelse( is.na(p) & !is.na(x), 1, p ) # fix problems with +Inf
  return(p*(max-min)+min)
}

#== Generating data

# Constants
nobs <- 1000
nspecies <- 20
species <- c(1:nspecies,sample(c(1:nspecies),(nobs-nspecies),replace=TRUE))

# Covariates
X1 <- runif(n=nobs,min=-10,max=10)
X2 <- runif(n=nobs,min=-10,max=10)
X <- cbind(rep(1,nobs),X1,X2)
W <- X

# Target parameters
# beta
beta.target <- matrix(c(0.3,0.2,0.1),ncol=1)
# Vb
Vb.target <- c(0.5,0.05,0.05)
# b
```

```

b.target <- cbind(rnorm(nspecies,mean=0,sd=sqrt(Vb.target[1])),
                 rnorm(nspecies,mean=0,sd=sqrt(Vb.target[2])),
                 rnorm(nspecies,mean=0,sd=sqrt(Vb.target[3])))

# Response
theta <- vector()
Y <- vector()
for (n in 1:nobs) {
  theta[n] <- inv.logit(X[n,]*%b.target+W[n,]*%b.target[species[n],])
  Y[n] <- rbinom(n=1,size=1,prob=theta[n])
}

# Data-set
Data <- as.data.frame(cbind(Y,theta,X1,X2,species))
plot(Data$X1,Data$theta)

#== Call to entry_mortality_gibbs
model <- entry_mortality_gibbs(fixed=Y~X1+X2, random=~X1+X2, group="species",
                              interval=1, data=Data, burnin=5000, mcmc=1000, thin=1,verbose=1,
                              seed=NA, beta.start=0, sigma2.start=1,
                              Vb.start=1, mubeta=0, Vbeta=1.0E6,
                              r=3, R=diag(c(1,0.1,0.1)), nu=0.001, delta=0.001, FixOD=1)

#== MCMC analysis

# Graphics
pdf("Posteriors-entry_mortality_gibbs.pdf")
plot(model$mcmc)
dev.off()

# Summary
summary(model$mcmc)

# Predictive posterior mean for each observation
model$theta.pred

# Predicted-Observed
plot(Data$theta,model$theta.pred)
abline(a=0,b=1)

## #Not run
## #You can also compare with lme4 results
## #== lme4 resolution
## library(lme4)
## model.lme4 <- lmer(Y~X1+X2+(1+X1+X2|species),data=Data,family="binomial")
## summary(model.lme4)
## plot(fitted(model.lme4),model$theta.pred,main="entry_mortality_gibbs/lme4")
## abline(a=0,b=1)

## End(Not run)

```

entry_ngrowth	<i>A function to estimate the parameters of the growth model for each species in the data-set.</i>
---------------	--

Description

A function to estimate the parameters of the growth model for each species in the data-set. The function prints summaries and plots diagnostic graphics.

Usage

```
entry_ngrowth(data_growth, a.sd=0.927, b.sd=0.0038, burnin=1000, mcmc=1000, thin=1, th.sign=5)
```

Arguments

data_growth	data frame which must have the following column names: Tree Sp D_t C_t G_tpl
a.sd	A positive scalar for the intercept of the function describing the standard deviation for the observation error process. Default to 0.927 (see Ruger et al. 2005).
b.sd	A positive scalar for the slope of the function describing the standard deviation for the observation error process. Default to 0.0038 (see Ruger et al. 2005).
burnin	The number of burnin iterations for the sampler.
mcmc	The number of Gibbs iterations for the sampler. Total number of Gibbs iterations is equal to burnin+mcmc. burnin+mcmc must be divisible by 10 and superior or equal to 100 so that the progress bar can be displayed.
thin	The thinning interval used in the simulation. The number of mcmc iterations must be divisible by this value.
th.sign	Threshold (in %) used to test parameter significance: is zero included in the interval defined by the threshold? Default to 5.

Value

The function prints summaries and plots diagnostic graphics.

Author(s)

Ghislain Vieilledent <ghislain.vieilledent@cirad.fr>

entry_ngrowth_gibbs	<i>Markov Chain Monte Carlo for the Hierarchical Gaussian Linear Regression Model with observation errors (negative growth)</i>
---------------------	---

Description

entry_ngrowth_gibbs generates a sample from the posterior distribution of a Hierarchical Gaussian Linear Regression Model with observation errors using Algorithm 2 of Chib and Carlin (1999). This model uses a multivariate Normal prior for the fixed effects parameters, an Inverse-Wishart prior on the random effects variance matrix, and an Inverse-Gamma prior on the residual error variance. The user supplies data and priors, and a sample from the posterior distribution is returned as an mcmc object, which can be subsequently analyzed with functions provided in the coda package.

Usage

```
entry_ngrowth_gibbs(fixed, random, group, diameter, data,
  burnin=1000, mcmc=10000, thin=10, verbose=1, seed=NA, a.sd=0.927,
  b.sd=0.0038, beta.start=NA, sigma2.start=NA, Vb.start=NA, mubeta=0,
  Vbeta=1.0E6, r, R, nu=0.001, delta=0.001, ...)
```

Arguments

fixed	A two-sided linear formula of the form 'y~x1+...+xp' describing the fixed-effects part of the model, with the response on the left of a '~' operator and the p fixed terms, separated by '+' operators, on the right.
random	A one-sided formula of the form '~x1+...+xq' specifying the model for the random effects part of the model, with the q random terms, separated by '+' operators.
group	String indicating the name of the grouping variable in data, defining the hierarchical structure of the model.
diameter	A numeric vector of length equal to the number of observations. diameter specifies the diameter of each observation for the observation error process.
data	A data frame containing the variables in the model.
burnin	The number of burnin iterations for the sampler.
mcmc	The number of Gibbs iterations for the sampler. Total number of Gibbs iterations is equal to burnin+mcmc. burnin+mcmc must be divisible by 10 and superior or equal to 100 so that the progress bar can be displayed.
thin	The thinning interval used in the simulation. The number of mcmc iterations must be divisible by this value.
seed	The seed for the random number generator. If NA, the Mersenne Twister generator is used with default seed 12345; if an integer is passed it is used to seed the Mersenne twister.
a.sd	A positive scalar for the intercept of the function describing the standard deviation for the observation error process. Default to 0.927 (see Ruger et al. 2005).

b.sd	A positive scalar for the slope of the function describing the standard deviation for the observation error process. Default to 0.0038 (see Ruger et al. 2005).
verbose	A switch (0,1) which determines whether or not the progress of the sampler is printed to the screen. Default is 1: a progress bar is printed, indicating the step (in %) reached by the Gibbs sampler.
beta.start	The starting values for the β vector. This can either be a scalar or a p-length vector. The default value of NA will use the OLS β estimate of the corresponding Gaussian Linear Regression without random effects. If this is a scalar, that value will serve as the starting value mean for all of the betas.
sigma2.start	Scalar for the starting value of the residual error variance. The default value of NA will use the OLS estimates of the corresponding Gaussian Linear Regression without random effects.
Vb.start	The starting value for variance matrix of the random effects. This must be a square q-dimension matrix. Default value of NA uses an identity matrix.
mubeta	The prior mean of β . This can either be a scalar or a p-length vector. If this takes a scalar value, then that value will serve as the prior mean for all of the betas. The default value of 0 will use a vector of zeros for an uninformative prior.
Vbeta	The prior variance of β . This can either be a scalar or a square p-dimension matrix. If this takes a scalar value, then that value times an identity matrix serves as the prior variance of beta. Default value of 1.0E6 will use a diagonal matrix with very large variance for an uninformative flat prior.
r	The shape parameter for the Inverse-Wishart prior on variance matrix for the random effects. r must be superior or equal to q. Set r=q for an uninformative prior. See the NOTE for more details
R	The scale matrix for the Inverse-Wishart prior on variance matrix for the random effects. This must be a square q-dimension matrix. Use plausible variance regarding random effects for the diagonal of R. See the NOTE for more details
nu	The shape parameter for the Inverse-Gamma prior on the residual error variance. Default value is nu=delta=0.001 for uninformative prior.
delta	The rate (1/scale) parameter for the Inverse-Gamma prior on the residual error variance. Default value is nu=delta=0.001 for uninformative prior.
...	further arguments to be passed

Details

entry_ngrowth_gibbs simulates from the posterior distribution sample using the blocked Gibbs sampler of Chib and Carlin (1999), Algorithm 2. The simulation is done in compiled C++ code to maximize efficiency. Please consult the coda documentation for a comprehensive list of functions that can be used to analyze the posterior sample.

The model takes the following form:

Observation error process:

$$y_i^{obs} \sim \mathcal{N}(y_i^{true}, (a + bD_i)^2)$$

Growth process:

$$\log(y_i^{true}) = X_i\beta + W_ib_i + \varepsilon_i$$

Where each group i have k_i observations.

Where the random effects:

$$b_i \sim \mathcal{N}_q(0, V_b)$$

And the errors:

$$\varepsilon_i \sim \mathcal{N}(0, \sigma^2 I_{k_i})$$

We assume standard, conjugate priors:

$$\beta \sim \mathcal{N}_p(\mu_\beta, V_\beta)$$

And:

$$\sigma^2 \sim \mathcal{IGamma}(\nu, 1/\delta)$$

And:

$$V_b \sim \mathcal{IWishart}(r, rR)$$

See Chib and Carlin (1999) for more details.

NOTE: We do not provide default parameters for the priors on the precision matrix for the random effects. When fitting one of these models, it is of utmost importance to choose a prior that reflects your prior beliefs about the random effects. Using the `dwish` and `rwish` functions might be useful in choosing these values.

Value

mcmc	An mcmc object that contains the posterior sample. This object can be summarized by functions provided by the coda package. The posterior sample of the deviance D , with $D = -2 \log(\prod_i P(y_i^{obs} y_i^{true})P(y_i^{true} \theta_i))$, is also provided.
Y.true	"True" observations of the latente variable.
Y.pred	Predictions after back-transforming the predictive posterior mean of the log-linear model.
pD	Effective number of the parameters of the model.
DIC	Deviance Information Criterion.

Author(s)

Ghislain Vieilledent <ghislain.vieilledent@cirad.fr>

References

- Nadja Ruger, Uta Berger, Stephen P. Hubbell, Ghislain Vieilledent and Richard Condit. 2011. "Growth strategies of tropical tree species: disentangling light and size effects." *PLoS ONE*. 6(9): e25330. doi:10.1371/journal.pone.0025330
- Siddhartha Chib and Bradley P. Carlin. 1999. "On MCMC Sampling in Hierarchical Longitudinal Models." *Statistics and Computing*. 9: 17-26.
- Daniel Pemstein, Kevin M. Quinn, and Andrew D. Martin. 2007. *Scythe Statistical Library 1.0*. <http://scythe.wustl.edu>.
- Andrew D. Martin and Kyle L. Saunders. 2002. "Bayesian Inference for Political Science Panel Data." Paper presented at the 2002 Annual Meeting of the American Political Science Association.
- Martyn Plummer, Nicky Best, Kate Cowles, and Karen Vines. 2002. *Output Analysis and Diagnostics for MCMC (CODA)*. <http://www-fis.iarc.fr/coda/>.

See Also

[plot.mcmc](#), [summary.mcmc](#)

Examples

```
## Not run:
#=====
# Hierarchical Gaussian Linear Regression
#=====

#== Generating data

# Constants
nobs <- 1000
nspecies <- 20
species <- c(1:nspecies,sample(c(1:nspecies),(nobs-nspecies),replace=TRUE))

# Covariates
X1 <- runif(n=nobs,min=0,max=10)
X2 <- runif(n=nobs,min=0,max=10)
X <- cbind(rep(1,nobs),X1,X2)
W <- X

# Target parameters
# beta
beta.target <- matrix(c(0.1,0.3,0.2),ncol=1)
# Vb
Vb.target <- c(0.5,0.2,0.1)
# b
b.target <- cbind(rnorm(nspecies,mean=0,sd=sqrt(Vb.target[1])),
                  rnorm(nspecies,mean=0,sd=sqrt(Vb.target[2])),
                  rnorm(nspecies,mean=0,sd=sqrt(Vb.target[3])))
# sigma2
sigma2.target <- 0.02

# Response
Y <- vector()
for (n in 1:nobs) {
  Y[n] <- rnorm(n=1,
               mean=X[n,]%*%beta.target+W[n,]%*%b.target[species[n],],
               sd=sqrt(sigma2.target))
}

# Data-set
Data <- as.data.frame(cbind(Y,X1,X2,species))
plot(Data$X1,Data$Y)

#== Call to entry_growth_gibbs
model <- entry_growth_gibbs(fixed=Y~X1+X2, random=~X1+X2, group="species",
                           data=Data, burnin=1000, mcmc=1000, thin=1,verbose=1,
                           seed=NA, beta.start=0, sigma2.start=1,
```

```

        Vb.start=1, mubeta=0, Vbeta=1.0E6,
        r=3, R=diag(c(1,0.1,0.1)), nu=0.001, delta=0.001)

##== MCMC analysis

# Graphics
pdf("Posteriors-entry_growth_gibbs.pdf")
plot(model$mcmc)
dev.off()

# Summary
summary(model$mcmc)

# Predictive posterior mean for each observation
model$Y.pred

# Predicted-Observed
plot(Data$Y,model$Y.pred)
abline(a=0,b=1)

## End(Not run)

```

entry_recruitment	<i>A function to estimate the parameters of the recruitment model for each species in the data-set.</i>
-------------------	---

Description

A function to estimate the parameters of the recruitment model for each species in the data-set. The function prints summaries and plots diagnostic graphics.

Usage

```
entry_recruitment(data_recruitment, burnin=5000, mcmc=5000, thin=5, th.sign=5)
```

Arguments

data_recruitment	data frame which must have the following column names: Cell Sp BAsp_t C_t R_tp1 interval_tp1 SCell
burnin	The number of burnin iterations for the sampler.
mcmc	The number of Gibbs iterations for the sampler. Total number of Gibbs iterations is equal to burnin+mcmc. burnin+mcmc must be divisible by 10 and superior or equal to 100 so that the progress bar can be displayed.
thin	The thinning interval used in the simulation. The number of mcmc iterations must be divisible by this value.
th.sign	Threshold (in %) used to test parameter significance: is zero included in the interval defined by the threshold? Default to 5.

Value

The function prints summaries and plots diagnostic graphics.

Author(s)

Ghislain Vieilledent <ghislain.vieilledent@cirad.fr>

entry_recruitment_gibbs

Markov Chain Monte Carlo for the Hierarchical Poisson Linear Regression Model using the log link function

Description

entry_recruitment_gibbs generates a sample from the posterior distribution of a Hierarchical Poisson Linear Regression Model using the log link function and Algorithm 2 of Chib and Carlin (1999). This model uses a multivariate Normal prior for the fixed effects parameters, an Inverse-Wishart prior on the random effects variance matrix, and an Inverse-Gamma prior on the variance modelling over-dispersion. The user supplies data and priors, and a sample from the posterior distribution is returned as an mcmc object, which can be subsequently analyzed with functions provided in the coda package.

Usage

```
entry_recruitment_gibbs(fixed, random, group, interval=1, area=1,
  data, burnin=5000, mcmc=10000, thin=10, verbose=1, seed=NA,
  beta.start=NA, sigma2.start=NA, Vb.start=NA, mubeta=0, Vbeta=1.0E6, r,
  R, nu=0.001, delta=0.001, FixOD=0, ...)
```

Arguments

fixed	A two-sided linear formula of the form 'y~x1+...+xp' describing the fixed-effects part of the model, with the response on the left of a '~' operator and the p fixed terms, separated by '+' operators, on the right. Response variable y must be 0 or 1 (Binomial process).
random	A one-sided formula of the form '~x1+...+xq' specifying the model for the random effects part of the model, with the q random terms, separated by '+' operators.
group	String indicating the name of the grouping variable in data, defining the hierarchical structure of the model.
interval	A numeric scalar or vector of length equal to the number of observations. interval specifies the time interval between censuses for the observation of recruitment. Default to 1.
area	A numeric scalar or vector of length equal to the number of observations. area specifies the ground cell surface on which the recruitment is observed. Default to 1.

data	A data frame containing the variables in the model.
burnin	The number of burnin iterations for the sampler.
mcmc	The number of Gibbs iterations for the sampler. Total number of Gibbs iterations is equal to burnin+mcmc. burnin+mcmc must be divisible by 10 and superior or equal to 100 so that the progress bar can be displayed.
thin	The thinning interval used in the simulation. The number of mcmc iterations must be divisible by this value.
seed	The seed for the random number generator. If NA, the Mersenne Twister generator is used with default seed 12345; if an integer is passed it is used to seed the Mersenne twister.
verbose	A switch (0,1) which determines whether or not the progress of the sampler is printed to the screen. Default is 1: a progress bar is printed, indicating the step (in %) reached by the Gibbs sampler.
beta.start	The starting values for the β vector. This can either be a scalar or a p-length vector. The default value of NA will use the OLS β estimate of the corresponding Gaussian Linear Regression without random effects. If this is a scalar, that value will serve as the starting value mean for all of the betas.
sigma2.start	Scalar for the starting value of the residual error variance. The default value of NA will use the OLS estimates of the corresponding Gaussian Linear Regression without random effects.
Vb.start	The starting value for variance matrix of the random effects. This must be a square q-dimension matrix. Default value of NA uses an identity matrix.
mubeta	The prior mean of β . This can either be a scalar or a p-length vector. If this takes a scalar value, then that value will serve as the prior mean for all of the betas. The default value of 0 will use a vector of zeros for an uninformative prior.
Vbeta	The prior variance of β . This can either be a scalar or a square p-dimension matrix. If this takes a scalar value, then that value times an identity matrix serves as the prior variance of beta. Default value of 1.0E6 will use a diagonal matrix with very large variance for an uninformative flat prior.
r	The shape parameter for the Inverse-Wishart prior on variance matrix for the random effects. r must be superior or equal to q. Set r=q for an uninformative prior. See the NOTE for more details
R	The scale matrix for the Inverse-Wishart prior on variance matrix for the random effects. This must be a square q-dimension matrix. Use plausible variance regarding random effects for the diagonal of R. See the NOTE for more details
nu	The shape parameter for the Inverse-Gamma prior on the residual error variance. Default value is nu=delta=0.001 for uninformative prior.
delta	The rate (1/scale) parameter for the Inverse-Gamma prior on the residual error variance. Default value is nu=delta=0.001 for uninformative prior.
FixOD	A switch (0,1) which determines whether or not the variance for over-dispersion (sigma2) should be fixed (1) or not (0). Default is 0, parameter sigma2 is estimated. If FixOD=1, sigma2 is fixed to the value provided for sigma2.start.
...	further arguments to be passed

Details

`entry_recruitment_gibbs` simulates from the posterior distribution sample using the blocked Gibbs sampler of Chib and Carlin (1999), Algorithm 2. The simulation is done in compiled C++ code to maximize efficiency. Please consult the coda documentation for a comprehensive list of functions that can be used to analyze the posterior sample.

The model takes the following form:

$$y_i \sim \text{Poisson}(\lambda_i)$$

With $\lambda'_i = \lambda_i I_i S_i$, where I_i stands for the time interval between censuses for the observation of the recruitment on ground cell i . S_i stands for the surface of ground cell i on which the recruitment is observed. Thus, λ_i is the "recruitment rate": the number of recruits by unit of time and space.

With latent variables $\phi(\lambda_i)$, ϕ being the log link function:

$$\phi(\lambda_i) = X_i \beta + W_i b_i + \varepsilon_i$$

Where each group i have k_i observations.

Where the random effects:

$$b_i \sim \mathcal{N}_q(0, V_b)$$

And the over-dispersion terms:

$$\varepsilon_i \sim \mathcal{N}(0, \sigma^2 I_{k_i})$$

We assume standard, conjugate priors:

$$\beta \sim \mathcal{N}_p(\mu_\beta, V_\beta)$$

And:

$$\sigma^2 \sim \text{IGamma}(\nu, 1/\delta)$$

And:

$$V_b \sim \text{IWishart}(r, rR)$$

See Chib and Carlin (1999) for more details.

NOTE: We do not provide default parameters for the priors on the precision matrix for the random effects. When fitting one of these models, it is of utmost importance to choose a prior that reflects your prior beliefs about the random effects. Using the `dwish` and `rwish` functions might be useful in choosing these values.

Value

<code>mcmc</code>	An <code>mcmc</code> object that contains the posterior sample. This object can be summarized by functions provided by the coda package. The posterior sample of the deviance D , with $D = -2 \log(\prod_i P(y_i \lambda_i))$, is also provided.
<code>lambda.pred</code>	Predictive posterior mean for the exponential of the latent variables. The approximation of Diggle et al. (2004) is used to marginalized with respect to over-dispersion terms:

$$E[\lambda_i | \beta, b_i, \sigma^2] = \phi^{-1}((X_i \beta + W_i b_i) + 0.5 \sigma^2)$$

Author(s)

Ghislain Vieilledent <ghislain.vieilledent@cirad.fr>

References

Siddhartha Chib and Bradley P. Carlin. 1999. “On MCMC Sampling in Hierarchical Longitudinal Models.” *Statistics and Computing*. 9: 17-26.

Daniel Pemstein, Kevin M. Quinn, and Andrew D. Martin. 2007. *Scythe Statistical Library 1.0*. <http://scythe.wustl.edu>.

Andrew D. Martin and Kyle L. Saunders. 2002. “Bayesian Inference for Political Science Panel Data.” Paper presented at the 2002 Annual Meeting of the American Political Science Association.

Martyn Plummer, Nicky Best, Kate Cowles, and Karen Vines. 2002. *Output Analysis and Diagnostics for MCMC (CODA)*. <http://www-fis.iarc.fr/coda/>.

See Also

[plot.mcmc](#), [summary.mcmc](#)

Examples

```
## Not run:
#####
# Hierarchical Poisson Linear Regression
#####

#== Generating data

# Constants
nobs <- 1000
nspecies <- 20
species <- c(1:nspecies,sample(c(1:nspecies),(nobs-nspecies),replace=TRUE))

# Covariates
X1 <- runif(n=nobs,min=-1,max=1)
X2 <- runif(n=nobs,min=-1,max=1)
X <- cbind(rep(1,nobs),X1,X2)
W <- X

# Target parameters
# beta
beta.target <- matrix(c(0.1,0.1,0.1),ncol=1)
# Vb
Vb.target <- c(0.05,0.05,0.05)
# b
b.target <- cbind(rnorm(nspecies,mean=0,sd=sqrt(Vb.target[1])),
                  rnorm(nspecies,mean=0,sd=sqrt(Vb.target[2])),
                  rnorm(nspecies,mean=0,sd=sqrt(Vb.target[3])))

# Response
```

```

lambda <- vector()
Y <- vector()
for (n in 1:nobs) {
  lambda[n] <- exp(X[n,]*%*%beta.target+W[n,]*%*%b.target[species[n],])
  Y[n] <- rpois(1,lambda[n])
}

# Data-set
Data <- as.data.frame(cbind(Y,lambda,X1,X2,species))
plot(Data$X1,Data$lambda)

#== Call to entry_recruitment_gibbs
model <- entry_recruitment_gibbs(fixed=Y~X1+X2, random=~X1+X2, group="species",
  interval=1, area=1, data=Data, burnin=5000, mcmc=1000, thin=1,verbose=1,
  seed=NA, beta.start=0, sigma2.start=1,
  Vb.start=1, mubeta=0, Vbeta=1.0E6,
  r=3, R=diag(c(0.1,0.1,0.1)), nu=0.001, delta=0.001, FixOD=1)

#== MCMC analysis

# Graphics
pdf("Posteriors-entry_recruitment_gibbs.pdf")
plot(model$mcmc)
dev.off()

# Summary
summary(model$mcmc)

# Predictive posterior mean for each observation
model$lambda.pred

# Predicted-Observed
plot(Data$lambda,model$lambda.pred)
abline(a=0,b=1)

## #Not run
## #You can also compare with lme4 results
## #== lme4 resolution
## library(lme4)
## model.lme4 <- lmer(Y~X1+X2+(1+X1+X2|species),data=Data,family="poisson")
## summary(model.lme4)
## plot(fitted(model.lme4),model$lambda.pred,main="entry_recruitment_gibbs/lme4")
## abline(a=0,b=1)

## End(Not run)

```


Description

Function to simulate tropical forest dynamics. It uses the species parameters obtain by the "exit_..." functions. Function also plots the evolution of the basal area of each selected plot for simulations (exit_BA.pdf) and the evolution of the basal area of each species on each plot (exit_biodiversity.pdf).

Usage

```
exit_simu(Data, XY.Plot, D.Recruitment, R.Comp, L.Cell, Plot.Sim,
Year.Sim, Step.Sim)
```

Arguments

Data	Data frame of the permanent plot data-set.
XY.Plot	Data frame with the dimensions (m) of each plot: lPlotlX.PlotlY.Plotl
D.Recruitment	recruitment diameter (cm)
R.Comp	radius (m) for the competition area around each tree
L.Cell	width (m) of the ground cell for recruitment estimation
Plot.Sim	vector of plots name choosen for simulations
Year.Sim	number of years of simulation from the first census
Step.Sim	time step for which the simulation are kept in memory

Value

Return a data-set with the monitoring of each tree from first census and plot the evolution of the basal area of the plot with time.

Author(s)

Ghislain Vieilledent <ghislain.vieilledent@cirad.fr>

InvGamma

The Inverse Gamma Distribution

Description

Density function and random generation from the inverse gamma distribution.

Usage

```
rinvgamma(n, shape, scale = 1)
dinvgamma(x, shape, scale = 1)
```

Arguments

x	Scalar location to evaluate density.
n	Number of draws from the distribution.
shape	Scalar shape parameter.
scale	Scalar scale parameter (default value one).

Details

An inverse gamma random variable with shape a and scale b has mean $\frac{b}{a-1}$ (assuming $a > 1$) and variance $\frac{b^2}{(a-1)^2(a-2)}$ (assuming $a > 2$).

Value

dinvgamma evaluates the density at x. rinvgamma takes n draws from the inverse Gamma distribution. The parameterization is consistent with the Gamma Distribution in the stats package.

Author(s)

Andrew D. Martin <admartin@wustl.edu>, Kevin M. Quinn <kquinn@law.berkeley.edu>, and Jong Hee Park <jhp@uchicago.edu>

References

Andrew Gelman, John B. Carlin, Hal S. Stern, and Donald B. Rubin. 2004. *Bayesian Data Analysis*. 2nd Edition. Boca Raton: Chapman & Hall.

See Also

[GammaDist](#)

Examples

```
## Not run:
density <- dinvgamma(4.2, 1.1)
draws <- rinvgamma(10, 3.2)

## End(Not run)
```

Description

Density function and random generation from the Inverse Wishart distribution.

Usage

```
diwish(W, v, S)
riwish(v, S)
```

Arguments

W	Positive definite matrix W ($p \times p$).
v	Degrees of freedom (scalar).
S	Scale matrix ($p \times p$).

Details

The mean of an inverse Wishart random variable with v degrees of freedom and scale matrix S is $(v - p - 1)^{-1}S$.

Value

`diwish` evaluates the density at positive definite matrix W . `riwish` generates one random draw from the distribution.

Author(s)

Andrew D. Martin <admartin@wustl.edu>, Kevin M. Quinn <kquinn@law.berkeley.edu>, and Jong Hee Park <jhp@uchicago.edu>

Examples

```
## Not run:
density <- diwish(matrix(c(2,-.3,-.3,4),2,2), 3, matrix(c(1,.3,.3,1),2,2))
draw <- riwish(3, matrix(c(1,.3,.3,1),2,2))

## End(Not run)
```

logit

*Generalized logit and inverse logit function***Description**

Compute generalized logit and generalized inverse logit functions.

Usage

```
logit(x, min = 0, max = 1)
inv.logit(x, min = 0, max = 1)
```

Arguments

x	value(s) to be transformed
min	Lower end of logit interval
max	Upper end of logit interval

Details

The generalized logit function takes values on [min, max] and transforms them to span [-Inf,Inf] it is defined as:

$$y = \log\left(\frac{p}{(1-p)}\right)$$

where

$$p = \frac{(x - \text{min})}{(\text{max} - \text{min})}$$

The generalized inverse logit function provides the inverse transformation:

$$x = p'(\text{max} - \text{min}) + \text{min}$$

where

$$p' = \frac{\exp(y)}{(1 + \exp(y))}$$

Value

Transformed value(s).

Author(s)

Gregory R. Warnes <greg@warnes.net>

Examples

```
## Not run:
x <- seq(0,10, by=0.25)
xt <- logit(x, min=0, max=10)
cbind(x,xt)

y <- inv.logit(xt, min=0, max=10)
cbind(x,xt,y)

## End(Not run)
```

surface_comp_index	<i>Function to compute the competition area (m²) around each target tree.</i>
--------------------	--

Description

Function to compute the competition area (m²) around each target tree. As some trees are close to the plot edges, the competition area is changing from tree to tree. The competition area is equal to the intersection between a circle having the target tree as center and the rectangle delimited by the plot edges.

Usage

```
surface_comp_index(X.trees, Y.trees, X.Plot, Y.Plot, R.Comp)
```

Arguments

X.trees	vector of X coordinates (m) of each tree
Y.trees	vector of Y coordinates (m) of each tree
X.Plot	size of the plot (m) on the X axis
Y.Plot	size of the plot (m) on the Y axis
R.Comp	radius (m) for the competition area around each tree

Value

Return a vector of competition area (m²) for each tree

Author(s)

Ghislain Vieilledent <ghislain.vieilledent@cirad.fr>

Examples

```
## Not run:
X.trees <- runif(50,0,100) # X coordinates of the trees
Y.trees <- runif(50,0,100) # Y coordinates of the trees
X.Plot <- 100 # Plot is a 1ha plot with edge=100m
Y.Plot <- 100
R.Comp <- 10 # Competition on a 10m radius around each tree
S_vect <- surface_comp_index(X.trees, Y.trees, X.Plot, Y.Plot, R.Comp)

## End(Not run)
```

Wishart

*The Wishart Distribution***Description**

Density function and random generation from the Wishart distribution.

Usage

```
dwish(W, v, S)
rwish(v, S)
```

Arguments

W	Positive definite matrix W ($p \times p$).
v	Degrees of freedom (scalar).
S	Inverse scale matrix ($p \times p$).

Details

The mean of a Wishart random variable with v degrees of freedom and inverse scale matrix S is vS .

Value

`dwish` evaluates the density at positive definite matrix W . `rwish` generates one random draw from the distribution.

Author(s)

Andrew D. Martin <admartin@wustl.edu>, Kevin M. Quinn <kquinn@law.berkeley.edu>, and Jong Hee Park <jhp@uchicago.edu>

Examples

```
## Not run:
density <- dwish(matrix(c(2,-.3,-.3,4),2,2), 3, matrix(c(1,.3,.3,1),2,2))
draw <- rwish(3, matrix(c(1,.3,.3,1),2,2))

## End(Not run)
```

Index

- *Topic **Binomial**
 - entry_mortality_gibbs, 9
 - *Topic **Gaussian**
 - entry_growth_gibbs, 4
 - entry_ngrowth_gibbs, 15
 - *Topic **Gibbs sampler**
 - entry_growth, 3
 - entry_mortality, 8
 - entry_ngrowth, 14
 - entry_recruitment, 19
 - *Topic **MCMC**
 - entry_growth_gibbs, 4
 - entry_mortality_gibbs, 9
 - entry_ngrowth_gibbs, 15
 - entry_recruitment_gibbs, 20
 - *Topic **Permanent plot data-set**
 - entry_data, 2
 - *Topic **Poisson**
 - entry_recruitment_gibbs, 20
 - *Topic **bayesian**
 - entry_growth_gibbs, 4
 - entry_mortality_gibbs, 9
 - entry_ngrowth_gibbs, 15
 - entry_recruitment_gibbs, 20
 - *Topic **competition index**
 - surface_comp_index, 29
 - *Topic **distribution**
 - InvGamma, 25
 - InvWishart, 27
 - Wishart, 30
 - *Topic **forest dynamics simulation**
 - exit_simu, 24
 - *Topic **glmm**
 - entry_mortality_gibbs, 9
 - entry_recruitment_gibbs, 20
 - *Topic **hierarchical models**
 - entry_growth_gibbs, 4
 - entry_mortality_gibbs, 9
 - entry_ngrowth_gibbs, 15
 - entry_recruitment_gibbs, 20
 - *Topic **logit**
 - entry_mortality_gibbs, 9
 - *Topic **math**
 - logit, 28
 - *Topic **mixed models**
 - entry_growth_gibbs, 4
 - entry_mortality_gibbs, 9
 - entry_ngrowth_gibbs, 15
 - entry_recruitment_gibbs, 20
 - *Topic **models**
 - entry_growth_gibbs, 4
 - entry_mortality_gibbs, 9
 - entry_ngrowth_gibbs, 15
 - entry_recruitment_gibbs, 20
 - *Topic **tropical forest dynamics**
 - twoe-package, 2
- dinvgamma (InvGamma), 25
- diwish (InvWishart), 27
- dwish (Wishart), 30
- entry_data, 2
- entry_growth, 3
- entry_growth_gibbs, 4
- entry_mortality, 8
- entry_mortality_gibbs, 9
- entry_ngrowth, 14
- entry_ngrowth_gibbs, 15
- entry_recruitment, 19
- entry_recruitment_gibbs, 20
- exit_simu, 24
- GammaDist, 26
- inv.logit (logit), 28
- InvGamma, 25
- InvWishart, 27
- logit, 28

`plot.mcmc`, [6](#), [12](#), [18](#), [23](#)

`rinvgamma` (`InvGamma`), [25](#)

`riwish` (`InvWishart`), [27](#)

`rwish` (`Wishart`), [30](#)

`summary.mcmc`, [6](#), [12](#), [18](#), [23](#)

`surface_comp_index`, [29](#)

`twoe` (`twoe-package`), [2](#)

`twoe-package`, [2](#)

`Wishart`, [30](#)