

# The DPpackage Package

May 12, 2006

**Version** 1.0-0

**Date** 2006-03-27

**Title** Semiparametric Bayesian Analysis using Dirichlet process priors

**Author** Alejandro Jara Vallejos <Alejandro.JaraVallejos@med.kuleuven.be>

**Maintainer** Alejandro Jara Vallejos <Alejandro.JaraVallejos@med.kuleuven.be>

**Description** This package contains functions for posterior analysis for a number of semiparametric statistical models. Simulation is done in compiled FORTRAN.

**License** GPL version 2 or newer

**URL** <http://www.student.kuleuven.ac.be/%7Es0166452/software.html>

## R topics documented:

DPbinary . . . . .	2
DPdensity . . . . .	5
DPelicit . . . . .	9
DPglmm . . . . .	10
DPimm . . . . .	14
DPpsBF . . . . .	18
DPrandom . . . . .	19
DPsurvint . . . . .	20
DPsurvpred . . . . .	23
fleabeetles . . . . .	26
galaxy . . . . .	27
indon . . . . .	27
rats . . . . .	28
schoolgirls . . . . .	29
seizures . . . . .	30
<b>Index</b>	<b>31</b>

---

DPbinary	<i>Performs a Bayesian analysis for a semiparametric binary regression model</i>
----------	--

---

### Description

This function generates a posterior density sample for a semiparametric binary regression model.

### Usage

```
DPbinary(formula, prior, mcmc, state, status, misc=NULL,
         data=sys.frame(sys.parent()), na.action=na.fail)
```

### Arguments

formula	a two-sided linear formula object describing the fixed-effects part of the model, with the response on the left of a $\sim$ operator and the terms, separated by + operators, on the right.
prior	a list giving the prior information. The list includes the following parameters: <code>a0</code> and <code>b0</code> giving the hyperparameters for prior distribution of the precision parameter of the Dirichlet process prior, <code>alpha</code> giving the value of the precision parameter (it must be specified if <code>a0</code> and <code>b0</code> are missing, see the details below), and <code>beta0</code> and <code>Sbeta0</code> giving the hyperparameters of the normal prior distribution for the regression coefficients.
mcmc	a list giving the MCMC parameters. The list must include the following integers: <code>nburn</code> giving the number of burn-in scans, <code>nskip</code> giving the thinning interval, <code>nsave</code> giving the total number of scans to be saved, <code>ndisplay</code> giving the number of saved scans to be displayed on the screen (the function reports on the screen when every <code>ndisplay</code> iterations have been carried out), and <code>tune</code> giving the Metropolis tuning parameter.
state	a list giving the current value of the parameters. This list is used if the current analysis is the continuation of a previous analysis.
status	a logical variable indicating whether this run is new (TRUE) or the continuation of a previous analysis (FALSE). In the latter case the current value of the parameters must be specified in the object <code>state</code> .
misc	misclassification information. When used, this list must include two objects, <code>sens</code> and <code>spec</code> , giving the sensitivity and specificity, respectively. Both can be a vector or a scalar. This information is used to correct for misclassification in the conditional bernoulli model.
data	data frame.
na.action	a function that indicates what should happen when the data contain NAs. The default action ( <code>na.fail</code> ) causes <code>DPbinary</code> to print an error message and terminate if there are any incomplete observations.

## Details

This generic function fits a Dirichlet process in a AFT regression model for interval censored data (see, Jara, Garcia-Zattera and Lesaffre, 2006):

$$y_i = I(V_i < X_i\beta), i = 1, \dots, n$$

$$V_i|G \sim G$$

$$G|\alpha, G_0 \sim DP(\alpha G_0)$$

where,  $G_0 = \text{Logistic}(V|0, 1)$ . To complete the model specification, independent hyperpriors are assumed,

$$\alpha|a_0, b_0 \sim \text{Gamma}(a_0, b_0) \text{ (optional)}$$

$$\beta|\beta_0, S_{\beta_0} \sim N(\beta_0, S_{\beta_0})$$

The precision or total mass parameter,  $\alpha$ , of the DP prior can be considered as random, having a gamma distribution,  $\text{Gamma}(a_0, b_0)$ , or fixed at some particular value. When *alpha* is random the method described by Escobar and West (1995) is used. To let *alpha* to be fixed at a particular value, set *a<sub>0</sub>* to NULL in the prior specification.

A Metropolis-Hastings step is used to sample the fully conditional distribution of the regression coefficients and errors (see, Jara, Garcia-Zattera and Lesaffre, 2006). In the computational implementation of the model, G is considered as latent data and sampled partially with sufficient accuracy to be able to generate  $V_1, \dots, V_{n+1}$  which are exactly iid G, as proposed by Doss (1994). Both Ferguson's definition of DP and the Sethuraman-Tiwari (1982) representation of the process are used, as described in Jara, Garcia-Zattera and Lesaffre (2006).

## Value

An object of class `DPbinary` representing the semiparametric logistic regression model fit. Generic functions such as `print`, `plot`, and `summary` have methods to show the results of the fit. The results include `beta`, the precision parameter (`alpha`), the number of clusters (`ncluster`), and the `link` function.

The MCMC samples of the parameters and the errors in the model are stored in the object `thetasave` and `randsave`, respectively. Both objects are included in the list `save.state` and are matrices which can be analyzed directly by functions provided by the `coda` package.

The list `state` in the output object contains the current value of the parameters necessary to restart the analysis. If you want to specify different starting values to run multiple chains set `status=TRUE` and create the list `state` based on this starting values. In this case the list `state` must include the following objects:

`beta` giving the value of the regression coefficients,

`v` giving the value of the errors (it must be consistent with  $y_i = I(V_i < x_i \text{ beta})$ ),

`y` giving the value of the true response binary variable (only if the model considers correction for misclassification), and

`alpha` giving the value of the precision parameter.

## Author(s)

Alejandro Jara Vallejos (<Alejandro.JaraVallejos@med.kuleuven.be>)

## References

Doss, H. (1994). Bayesian nonparametric estimation for incomplete data using mixtures of Dirichlet priors. *The Annals of Statistics*, 22: 1763 - 1786.

Escobar, M.D. and West, M. (1995) Bayesian Density Estimation and Inference Using Mixtures. *Journal of the American Statistical Association*, 90: 577-588.

Jara, A., Garcia-Zattera, M.J., Lesaffre, E. (2006) Semiparametric Bayesian Analysis of Misclassified Binary Data, In preparation.

Sethuraman, J., and Tiwari, R. C. (1982) Convergence of Dirichlet Measures and the Interpretation of their Parameter, in *Statistical Decision Theory and Related Topics III* (vol. 2), eds. S. S. Gupta and J. O. Berger, New York: Academic Press, pp. 305 - 315.

## Examples

```
## Not run:
# Bioassay Data Example
# Cox, D.R. and Snell, E.J. (1989). Analysis of Binary Data. 2nd ed.
# Chapman and Hall. p. 7
# In this example there are 150 subjects at 5 different stimulus levels,
# 30 at each level.

y<-c(rep(0,30-2),rep(1,2),
      rep(0,30-8),rep(1,8),
      rep(0,30-15),rep(1,15),
      rep(0,30-23),rep(1,23),
      rep(0,30-27),rep(1,27))

x<-c(rep(0,30),
      rep(1,30),
      rep(2,30),
      rep(3,30),
      rep(4,30))

# Initial state
state <- NULL

# MCMC parameters
nburn<-50
nsave<-200
nskip<-5
ndisplay<-100
mcmc <- list(nburn=nburn,nsave=nsave,nskip=nskip,ndisplay=ndisplay,
             tune=0.125)

# Prior distribution
prior <- list(a0=2,b0=1,beta0=rep(0,2), Sbeta0=diag(10000,2))

# Fit the model
fit1 <- DPbinary(y~x,prior=prior,mcmc=mcmc,state=state,status=TRUE)
fit1

# Summary with HPD and Credibility intervals
summary(fit1)
summary(fit1,hpd=FALSE)

# Plot model parameters (to see the plots gradually set ask=TRUE)
```

```

plot(fit1)
plot(fit1,nfigr=2,nfigc=2)

# Plot an specific model parameter (to see the plots gradually set ask=TRUE)
plot(fit1,ask=FALSE,nfigr=1,nfigc=2,param="x")
plot(fit1,ask=FALSE,nfigr=1,nfigc=2,param="ncluster")
plot(fit1,ask=FALSE,param="link",nfigc=1,nfigr=1)
## End(Not run)

```

---

DPdensity

*Performs a Bayesian density estimation*


---

## Description

This function generates a posterior density sample for a Dirichlet process mixture of normals model.

## Usage

```

DPdensity(y,prior,mcmc,state,status,method="neal",
          data=sys.frame(sys.parent()),na.action=na.fail)

```

## Arguments

- |                    |  |
|--------------------|--|
| <code>y</code>     | a vector or matrix giving the data from which the density estimate is to be computed.  |
| <code>prior</code> | a list giving the prior information. The list includes the following parameter: <code>a0</code> and <code>b0</code> giving the hyperparameters for prior distribution of the precision parameter of the Dirichlet process prior, <code>alpha</code> giving the value of the precision parameter (it must be specified if <code>a0</code> is missing, see details below), <code>nu2</code> and <code>psiinv2</code> giving the hyperparameters of the inverted Wishart prior distribution for the scale matrix, <code>Psi1</code> , of the inverted Wishart part of the baseline distribution, <code>tau1</code> and <code>tau2</code> giving the hyperparameters for the gamma prior distribution of the scale parameter <code>k0</code> of the normal part of the baseline distribution, <code>m2</code> and <code>s2</code> giving the mean and the covariance of the normal prior for the mean, <code>m1</code> , of the normal component of the baseline distribution, respectively, <code>nu1</code> and <code>psiinv1</code> (it must be specified if <code>nu2</code> is missing, see details below) giving the hyperparameters of the inverted Wishart part of the baseline distribution and, <code>m1</code> giving the mean of the normal part of the baseline distribution (it must be specified if <code>m2</code> is missing, see details below) and, <code>k0</code> giving the scale parameter of the normal part of the baseline distribution (it must be specified if <code>tau1</code> is missing, see details below). |
| <code>mcmc</code>  | a list giving the MCMC parameters. The list must include the following integers: <code>nburn</code> giving the number of burn-in scans, <code>nskip</code> giving the thinning interval, <code>nsave</code> giving the total number of scans to be saved, and <code>ndisplay</code> giving the number of saved scans to be displayed on screen (the function reports on the screen when every <code>ndisplay</code> iterations have been carried out).   |
| <code>state</code> | a list giving the current value of the parameters. This list is used if the current analysis is the continuation of a previous analysis.   |

<code>status</code>	a logical variable indicating whether this run is new (TRUE) or the continuation of a previous analysis (FALSE). In the latter case the current value of the parameters must be specified in the object <code>state</code> .
<code>method</code>	the method to be used. See <code>Details</code> .
<code>data</code>	data frame.
<code>na.action</code>	a function that indicates what should happen when the data contain NAs. The default action ( <code>na.fail</code> ) causes <code>DPdensity</code> to print an error message and terminate if there are any incomplete observations.

## Details

This generic function fits a Dirichlet process mixture of normal model for density estimation (Escobar and West, 1995):

$$y_i | \mu_i, \sigma_i \sim N(\mu_i, \sigma_i), i = 1, \dots, n$$

$$(\mu_i, \sigma_i) | G \sim G$$

$$G | \alpha, G_0 \sim DP(\alpha G_0)$$

where, the baseline distribution is the conjugate normal-inverted-Wishart,

$$G_0 = N(\mu | m_1, (1/k_0)\sigma) IW(\sigma | \nu_1, \psi_1).$$

To complete the model specification, independent hyperpriors are assumed (optional),

$$\alpha | a_0, b_0 \sim Gamma(a_0, b_0)$$

$$m_1 | m_2, s_2 \sim N(m_2, s_2)$$

$$k_0 | \tau_1, \tau_2 \sim Gamma(\tau_1/2, \tau_2/2)$$

$$\psi_1 | \nu_2, \psi_2 \sim IW(\nu_2, \psi_2)$$

Note that the inverted-Wishart prior is parametrized such that if  $A \sim IW_q(\nu, \psi)$  then  $E(A) = \psi^{-1}/(\nu - q - 1)$ .

To let part of the baseline distribution fixed at a particular value, set the corresponding hyperparameters of the prior distributions to NULL in the hyperprior specification of the model.

Although the baseline distribution,  $G_0$ , is a conjugate prior in this model specification, the algorithms with auxiliary parameters described in MacEachern and Muller (1998) and Neal (2000) are adopted. Specifically, the no-gaps algorithm of MacEachern and Muller (1998), "no-gaps" and the algorithm 8 with  $m=1$  of Neal (2000), "neal", are considered in the `DPdensity` function. The default method is the algorithm 8 of Neal.

## Value

An object of class `DPdensity` representing the DP mixture of normals model fit. Generic functions such as `print` and `plot` have methods to show the results of the fit. The results include `alpha`, and the number of clusters.

The function `DPrandom` can be used to extract the posterior mean of the subject-specific means and covariance matrices.

The MCMC samples of the parameters and the errors in the model are stored in the object `thetasave` and `randsave`, respectively. Both objects are included in the list `save.state` and are matrices which can be analyzed directly by functions provided by the `coda` package.

The list `state` in the output object contains the current value of the parameters necessary to restart the analysis. If you want to specify different starting values to run multiple chains set `status=TRUE` and create the list `state` based on this starting values. In this case the list `state` must include the following objects:

`ncluster` an integer giving the number of clusters,  
`muclus` a matrix of dimension  $(\text{nobservations}+2) \times (\text{nvariables})$  giving the means of the clusters (only the first `ncluster` are considered to start the chain),  
`sigmaclus` a matrix of dimension  $(\text{nobservations}+2) \times ((\text{nvariables}) \times (\text{nvariables})+1)/2$  giving the lower matrix of the covariance matrix of the clusters (only the first `ncluster` are considered to start the chain),  
`ss` an integer vector defining to which of the `ncluster` clusters each observation belongs,  
`alpha` giving the value of the precision parameter,  
`m1` giving the mean of the normal components of the baseline distribution,  
`k0` giving the scale parameter of the normal part of the baseline distribution, and  
`psil` giving the scale matrix of the inverted-Wishart part of the baseline distribution.

### Author(s)

Alejandro Jara (<Alejandro.JaraVallejos@med.kuleuven.be>)

### References

Escobar, M.D. and West, M. (1995) Bayesian Density Estimation and Inference Using Mixtures. *Journal of the American Statistical Association*, 90: 577-588.

MacEachern, S. N. and Muller, P. (1998) Estimating mixture of Dirichlet Process Models. *Journal of Computational and Graphical Statistics*, 7 (2): 223-338.

Neal, R. M. (2000). Markov Chain sampling methods for Dirichlet process mixture models. *Journal of Computational and Graphical Statistics*, 9: 249-265.

### See Also

[DPrandom](#)

### Examples

```
## Not run:
#####
# Univariate example
#####

# Data
data(galaxy)
galaxy<-data.frame(galaxy, speeds=galaxy$speed/1000)
attach(galaxy)

# Initial state
state <- NULL

# MCMC parameters

nburn<-10
nsave<-100
nskip<-10
ndisplay<-50
mcmc <- list(nburn=nburn,nsave=nsave,nskip=nskip,ndisplay=ndisplay)
```

```

# Example of Prior information 1
# Fixing alpha, m1, and Psi1

prior1<-list(alpha=1,m1=rep(0,1),psiinv1=diag(0.5,1),nu1=4,tau1=1,tau2=100)

# Example of Prior information 2
# Fixing alpha and m1

prior2<-list(alpha=1,m1=rep(0,1),psiinv2=diag(0.5,1),nu1=4,nu2=4,tau1=1,
             tau2=100)

# Example of Prior information 3
# Fixing only alpha

prior3<-list(alpha=1,m2=rep(0,1),s2=diag(100000,1),psiinv2=diag(0.5,1),
             nu1=4,nu2=4,tau1=1,tau2=100)

# Example of Prior information 4
# Everything is random

prior4<-list(a0=2,b0=1,m2=rep(0,1),s2=diag(100000,1),psiinv2=diag(0.5,1),
             u1=4,nu2=4,tau1=1,tau2=100)

# Fit the models

fit1.1<-DPdensity(y=speeds,prior=prior1,mcmc=mcmc,state=state,status=TRUE)
fit1.2<-DPdensity(y=speeds,prior=prior2,mcmc=mcmc,state=state,status=TRUE)
fit1.3<-DPdensity(y=speeds,prior=prior3,mcmc=mcmc,state=state,status=TRUE)
fit1.4<-DPdensity(y=speeds,prior=prior4,mcmc=mcmc,state=state,status=TRUE)

# Posterior means
fit1.1
fit1.2
fit1.3
fit1.4

# Plot the estimated density
plot(fit1.1,ask=FALSE)
plot(fit1.2,ask=FALSE)
plot(fit1.3,ask=FALSE)
plot(fit1.4,ask=FALSE)

# Plot the parameters (only prior 2 for illustration)
# (to see the plots gradually set ask=TRUE)
plot(fit1.2,ask=FALSE,output="param")

# Plot the a specific parameters
# (to see the plots gradually set ask=TRUE)
plot(fit1.2,ask=FALSE,output="param",param="psi1-speeds",nfigr=1,nfigc=2)

# Extracting the posterior mean of the specific means and covariance matrices
# (only prior 2 for illustration)
DPrandom(fit1.2)

# Plotting predictive information about the specific means and covariance matrices
# with HPD and Credibility intervals
# (only prior 2 for illustration)

```

```

# (to see the plots gradually set ask=TRUE)
plot(DPrandom(fit1.2,predictive=TRUE),ask=FALSE)
plot(DPrandom(fit1.2,predictive=TRUE),ask=FALSE,hpd=FALSE)

# Plotting information about all the specific means and covariance matrices
# with HPD and Credibility intervals
# (only prior 2 for illustration)
# (to see the plots gradually set ask=TRUE)
plot(DPrandom(fit1.2),ask=FALSE,hpd=FALSE)
## End(Not run)

```

---

DPelicit

*Performs a prior elicitation for the precision parameter of a DP prior*


---

## Description

This function performs a prior elicitation for the precision parameter of a DP prior. The function calculates:

- 1) the expected value and the standard deviation of the number of clusters, given the values of the parameters of the gamma prior for the precision parameter,  $a_0$  and  $b_0$ , or
- 2) the value of the parameters  $a_0$  and  $b_0$  of the gamma prior distribution for the precision parameter,  $\alpha$ , given the prior expected number and the standard deviation of the number of clusters.

## Usage

```
DPelicit(n,method='JGL',a0=NULL,b0=NULL,mean=NULL,std=NULL)
```

## Arguments

<code>n</code>	number of observations which distribution follows a DP prior.
<code>method</code>	the method to be used. See details.
<code>a0</code>	hyperparameter for the Gamma prior distribution of the precision parameter of the Dirichlet process prior, $\alpha \sim \text{Gamma}(a_0, b_0)$ .
<code>b0</code>	hyperparameter for the Gamma prior distribution of the precision parameter of the Dirichlet process prior, $\alpha \sim \text{Gamma}(a_0, b_0)$ .
<code>mean</code>	prior expected number of clusters when $\alpha \sim \text{Gamma}(a_0, b_0)$ .
<code>std</code>	prior standard deviation for the number of clusters when $\alpha \sim \text{Gamma}(a_0, b_0)$ .

## Details

The methods supported by this function are based on the fact that a priori  $E(\alpha) = a_0/b_0$  and  $\text{Var}(\alpha) = a_0/b_0^2$ , and an additional approximation based on Taylor series expansion.

The default method, "JGL", is based on the exact value of the expected value of the number of clusters given the precision parameter  $\alpha$  (see, Jara, Garcia-Zatera and Lesaffre, 2006) and an approximation to the variance of the number of clusters given the precision parameter (Liu, 1996).

The Method "KMQ" is based on the Liu (1996) approximation to both the expected value and the variance of the number of clusters given the precision parameter  $\alpha$  (see, Kottas, Muller and Quintana, 2005).

Given the prior judgement for the mean and variance of the number of clusters, the equations are numerically solve for  $a_0$  and  $b_0$ . With this objective, the Newton-Raphson algorithm and the forward-difference approximation to Jacobian are used.

### Author(s)

Alejandro Jara (<Alejandro.JaraVallejos@med.kuleuven.be>)

### References

Jara, A., Garcia-Zattera, M.J., Lesaffre, E. (2006) A Dirichlet Process mixture model for the analysis of correlated binary responses, Submitted.

Kottas, A., Muller, P., Quintana, F. (2005) Nonparametric Bayesian modeling for multivariate ordinal data, Journal of Computational and Graphical Statistics 14: 610-625.

Liu, J.S. (1996) Nonparametric Hierarchical Bayes via Sequential Imputations, The Annals of Statistics, 24: 911-930.

### Examples

```
# Calculate the expected value and the standard deviation
# for the number of cluster given alpha ~ Gamma(a0,b0).

DPelicit(200,a0=2.01,b0=2.01,method="JGL")
DPelicit(200,a0=2.01,b0=2.01,method="KMQ")

# Calculate the values of a0 and b0, given the expected value
# and the standard deviation of the number of clusters

DPelicit(200,mean=0.01633,std=1.90136,method="JGL")
DPelicit(200,mean=4.61500,std=4.09300,method="KMQ")
```

---

DPglmm

*Performs a Bayesian analysis for a semiparametric generalized linear mixed model*

---

### Description

This function generates a posterior density sample for a semiparametric generalized linear mixed model.

### Usage

```
DPglmm(fixed, random, family, offset, n, prior, mcmc, state, status,
       data=sys.frame(sys.parent()), na.action=na.fail)
```

**Arguments**

<code>fixed</code>	a two-sided linear formula object describing the fixed-effects part of the model, with the response on the left of a <code>~</code> operator and the terms, separated by <code>+</code> operators, on the right.
<code>random</code>	a one-sided formula of the form <code>~z1+...+zn   g</code> , with <code>z1+...+zn</code> specifying the model for the random effects and <code>g</code> the grouping variable. The random effects formula will be repeated for all levels of grouping.
<code>family</code>	a description of the error distribution and link function to be used in the model. This can be a character string naming a family function, a family function or the result of a call to a family function. The families(links) considered by <code>DPglmm</code> so far are <code>binomial(logit)</code> , <code>binomial(probit)</code> , and <code>poisson(log)</code> . The <code>gaussian(identity)</code> case is implemented separately in the function <code>DP1mm</code> .
<code>offset</code>	this can be used to specify an a priori known component to be included in the linear predictor during the fitting (only for poisson models).
<code>n</code>	this can be used to indicate the total number of cases in a binomial model (only implemented for the logistic link). If it is not specified the response variable must be binary.
<code>prior</code>	a list giving the prior information. The list includes the following parameter: <code>a0</code> and <code>b0</code> giving the hyperparameters for prior distribution of the precision parameter of the Dirichlet process prior, <code>alpha</code> giving the value of the precision parameter (it must be specified if <code>a0</code> is missing, see details below), <code>nu0</code> and <code>Tinv</code> giving the hyperparameters of the inverted Wishart prior distribution for the scale matrix of the normal baseline distribution, <code>mub</code> and <code>Sb</code> giving the hyperparameters of the normal prior distribution for the mean of the normal baseline distribution and, <code>beta0</code> and <code>Sbeta0</code> giving the hyperparameters of the normal prior distribution for the fixed effects (must be specified only if fixed effects are considered in the model).
<code>mcmc</code>	a list giving the MCMC parameters. The list must include the following integers: <code>nburn</code> giving the number of burn-in scans, <code>nskip</code> giving the thinning interval, <code>nsave</code> giving the total number of scans to be saved, and <code>ndisplay</code> giving the number of saved scans to be displayed on the screen (the function reports on the screen when every <code>ndisplay</code> iterations have been carried out).
<code>state</code>	a list giving the current value of the parameters. This list is used if the current analysis is the continuation of a previous analysis.
<code>status</code>	a logical variable indicating whether this run is new ( <code>TRUE</code> ) or the continuation of a previous analysis ( <code>FALSE</code> ). In the latter case the current value of the parameters must be specified in the object <code>state</code> .
<code>data</code>	data frame.
<code>na.action</code>	a function that indicates what should happen when the data contain NAs. The default action ( <code>na.fail</code> ) causes <code>DPglmm</code> to print an error message and terminate if there are any incomplete observations.

**Details**

This generic function fits a generalized linear mixed-effects model, where the linear predictor is modeled as follows:

$$\eta_i = X_i\beta_F + Z_i\beta_R + Z_ib_i, i = 1, \dots, n$$

$$\theta_i|G \sim G$$

$$G|\alpha, G_0 \sim DP(\alpha G_0)$$

where,  $\theta_i = \beta_R + b_i$ ,  $\beta = \beta_F$ , and  $G_0 = N(\theta|\mu, \Sigma)$ . To complete the model specification, independent hyperpriors are assumed,

$$\alpha|a_0, b_0 \sim \text{Gamma}(a_0, b_0) \text{ (optional)}$$

$$\beta|\beta_0, S_{\beta_0} \sim N(\beta_0, S_{\beta_0})$$

$$\mu|\mu_b, S_b \sim N(\mu_b, S_b)$$

$$\Sigma|\nu_0, T \sim IW(\nu_0, T)$$

Note that the inverted-Wishart prior is parametrized such that  $E(\Sigma) = T^{-1}/(\nu_0 - q - 1)$ .

The precision or total mass parameter,  $\alpha$ , of the DP prior can be considered as random, having a gamma distribution,  $\text{Gamma}(a_0, b_0)$ , or fixed at some particular value. When  $\alpha$  is random the method described by Escobar and West (1995) is used. To let  $\alpha$  to be fixed at a particular value set,  $a_0$  to NULL in the prior specification.

The computational implementation of the model is based on the marginalization of the DP and the MCMC is model-specific.

For the `poisson` and `binomial` (when the total number of cases,  $n$ , is specified) MCMC methods for nonconjugate priors (see, MacEachern and Muller, 1998; Neal, 2000) are used. Specifically, the algorithm 8 with `m=1` of Neal (2000), is considered in the `DPglm` function. In this case, the fully conditional distributions for fixed and in the resampling step of random effects are generated through the Metropolis-Hastings algorithm with a IWLS proposal (see, West, 1985 and Gamerman, 1997).

For conditional bernoulli models, `binomial(probit)` and `binomial(logit)`, (i.e., the total number of cases is not specified) the following latent variable representation is used:

$$y_{ij} = I(w_{ij} > 0), j = 1, \dots, n_i$$

$$w_{ij}|\beta, \theta_i, \lambda_i \sim N(X_{ij}\beta + Z_{ij}\theta_i, \lambda_{ij})$$

In the probit case,  $\lambda_{ij} = 1$  while in the logit case (see, Andrews and Mallows, 1974),

$$\lambda_{ij} = (2\phi_{ij})^2$$

$$\phi_{ij} \sim KS$$

where *KS* refers to the Kolmogorov Smirnov distribution. In this case, the computational implementation of the model is based on the marginalization of the DP and on the use of MCMC methods for conjugate priors (Escobar, 1994; Escobar and West, 1998). For the logistic model, the full conditional distribution of the scale parameters, `lambdaij` are sampled via the rejection sampling algorithm as proposed by Holmes and Held (2006).

## Value

An object of class `DPglm` representing the generalized linear mixed-effects model fit. Generic functions such as `print`, `plot`, and `summary` have methods to show the results of the fit. The results include `betaR`, `betaF`, `mu`, the elements of `Sigma`, the precision parameter `alpha`, and the number of clusters.

The function `DPrandom` can be used to extract the posterior mean of the random effects.

The list `state` in the output object contains the current value of the parameters necessary to restart the analysis. If you want to specify different starting values to run multiple chains set `status=TRUE` and create the list `state` based on this starting values. In this case the list `state` must include the following objects:

`ncluster` an integer giving the number of clusters,

`alpha` giving the value of the precision parameter,

**b** a matrix of dimension (nsubjects)\*(nrandom effects) giving the value of the random effects for each subject,  
**bclus** a matrix of dimension (nsubjects)\*(nrandom effects) giving the value of the random effects for each clusters (only the first ncluster are considered to start the chain),  
**ss** an integer vector defining to which of the ncluster clusters each subject belongs,  
**beta** giving the value of the fixed effects,  
**betar** giving the mean value of the random effects,  
**mu** giving the mean of the normal baseline distributions, and  
**sigma** giving the variance matrix of the normal baseline distributions.

### Author(s)

Alejandro Jara (<Alejandro.JaraVallejos@med.kuleuven.be>)

### References

- Andrews, D. and Mallows, C. (1974) Scale mixture of normal distributions. *Journal of the Royal Statistical Society, B*, 36: 99-102.
- Escobar, M.D. and West, M. (1995) Bayesian Density Estimation and Inference Using Mixtures. *Journal of the American Statistical Association*, 90: 577-588.
- Escobar, M.D. and West, M. (1998) Computing Bayesian Nonparametric Hierarchical Models, in *Practical Nonparametric and Semiparametric Bayesian Statistics*, eds: D. Dey, P. Muller, D. Sinha, New York: Springer-Verlag, pp. 1-22.
- Gamerman, D. (1997) Sampling from the posterior distribution in generalized linear mixed models. *Statistics and Computing*, 7: 57-68.
- Holmes, C.C. and Held, L. (2006) Bayesian auxiliary variable models for binary and multinomial regression. *Bayesian Analysis*, 1: 145-168.
- MacEachern, S. N. and Muller, P. (1998) Estimating mixture of Dirichlet Process Models. *Journal of Computational and Graphical Statistics*, 7 (2): 223-338.
- Neal, R. M. (2000) Markov Chain sampling methods for Dirichlet process mixture models. *Journal of Computational and Graphical Statistics*, 9:249-265.
- West, M. (1985) Generalized linear models: outlier accomodation, scale parameter and prior distributions. In *Bayesian Statistics 2* (eds Bernardo et al.), 531-558, Amsterdam: North Holland.

### See Also

[DPrandom](#), [DPlmm](#)

### Examples

```
## Not run:
# Respiratory Data Example

data(indon)
attach(indon)

baseage2<-baseage**2
follow<-age-baseage
follow2<-follow**2
```

```

# Prior information

beta0<-rep(0,9)
Sbeta0<-diag(1000,9)
tinv<-diag(1,1)
prior<-list(a0=2,b0=0.1,nu0=4,tinv=tinv,mub=rep(0,1),Sb=diag(1000,1),
            beta0=beta0,Sbeta0=Sbeta0)

# Initial state
state <- NULL

# MCMC parameters

nburn<-5
nsave<-100
nskip<-5
ndisplay<-100
mcmc <- list(nburn=nburn,nsave=nsave,nskip=nskip,ndisplay=ndisplay)

# Fit the Probit model
fit1<-DPglm(fixed=infect~gender+height+cosv+sinv+xero+baseage+baseage2+
            follow+follow2,random=~1|id,family=binomial(probit),
            prior=prior,mcmc=mcmc,state=state,status=TRUE)

# Fit the Logit model
fit2<-DPglm(fixed=infect~gender+height+cosv+sinv+xero+baseage+baseage2+
            follow+follow2,random=~1|id,family=binomial(logit),
            prior=prior,mcmc=mcmc,state=state,status=TRUE)

# Summary with HPD and Credibility intervals
summary(fit1)
summary(fit1,hpd=FALSE)

summary(fit2)
summary(fit2,hpd=FALSE)

# Plot model parameters (to see the plots gradually set ask=TRUE)
plot(fit1,ask=FALSE)
plot(fit1,ask=FALSE,nfigr=2,nfigc=2)

# Plot an specific model parameter (to see the plots gradually set ask=TRUE)
plot(fit1,ask=FALSE,nfigr=1,nfigc=2,param="baseage")
plot(fit1,ask=FALSE,nfigr=1,nfigc=2,param="ncluster")
## End(Not run)

```

---

DPlmm

*Performs a Bayesian analysis for a semiparametric linear mixed model*


---

## Description

This function generates a posterior density sample for a semiparametric linear mixed model.

**Usage**

```
DPlmm(fixed, random, prior, mcmc, state, status, data=sys.frame(sys.parent()),
      na.action=na.fail)
```

**Arguments**

<code>fixed</code>	a two-sided linear formula object describing the fixed-effects part of the model, with the response on the left of a $\sim$ operator and the terms, separated by + operators, on the right.
<code>random</code>	a one-sided formula of the form $\sim z_1 + \dots + z_n \mid g$ , with $z_1 + \dots + z_n$ specifying the model for the random effects and $g$ the grouping variable. The random effects formula will be repeated for all levels of grouping.
<code>prior</code>	a list giving the prior information. The list include the following parameter: <code>a0</code> and <code>b0</code> giving the hyperparameters for prior distribution of the precision parameter of the Dirichlet process prior, <code>alpha</code> giving the value of the precision parameter (it must be specified if <code>a0</code> and <code>b0</code> are missing, see details below), <code>nu0</code> and <code>Tinv</code> giving the hyperparameters of the inverted Wishart prior distribution for the scale matrix of the normal baseline distribution, <code>mub</code> and <code>Sb</code> giving the hyperparameters of the normal prior distribution for the mean of the normal baseline distribution, <code>beta0</code> and <code>Sbeta0</code> giving the hyperparameters of the normal prior distribution for the fixed effects (must be specified only if fixed effects are considered in the model) and, <code>tau1</code> and <code>tau2</code> giving the hyperparameters for the prior distribution of the error variance.
<code>mcmc</code>	a list giving the MCMC parameters. The list must include the following integers: <code>nburn</code> giving the number of burn-in scans, <code>nskip</code> giving the thinning interval, <code>nsave</code> giving the total number of scans to be saved, and <code>ndisplay</code> giving the number of saved scans to be displayed on screen (the function reports on the screen when every <code>ndisplay</code> iterations have been carried out).
<code>state</code>	a list giving the current value of the parameters. This list is used if the current analysis is the continuation of a previous analysis.
<code>status</code>	a logical variable indicating whether this run is new (TRUE) or the continuation of a previous analysis (FALSE). In the latter case the current value of the parameters must be specified in the object <code>state</code> .
<code>data</code>	data frame.
<code>na.action</code>	a function that indicates what should happen when the data contain NAs. The default action ( <code>na.fail</code> ) causes <code>DPlmm</code> to print an error message and terminate if there are any incomplete observations.

**Details**

This generic function fits a linear mixed-effects model (Verbeke and Molenberghs, 2000):

$$y_i \sim N(X_i \beta_F + Z_i \beta_R + Z_i b_i, \sigma_e^2 I_{n_i}), i = 1, \dots, n$$

$$\theta_i | G \sim G$$

$$G | \alpha, G_0 \sim DP(\alpha G_0)$$

$$\sigma_e^{-2} | \tau_1, \tau_2 \sim \text{Gamma}(\tau_1/2, \tau_2/2)$$

where,  $\theta_i = \beta_R + b_i$ ,  $\beta = \beta_F$ , and  $G_0 = N(\theta | \mu, \Sigma)$ . To complete the model specification, independent hyperpriors are assumed,

$$\alpha|a_0, b_0 \sim \text{Gamma}(a_0, b_0) \text{ (optional)}$$

$$\beta|\beta_0, S_{\beta_0} \sim N(\beta_0, S_{\beta_0})$$

$$\mu|\mu_b, S_b \sim N(\mu_b, S_b)$$

$$\Sigma|\nu_0, T \sim IW(\nu_0, T)$$

Note that the inverted-Wishart prior is parametrized such that  $E(\Sigma) = T^{-1}/(\nu_0 - q - 1)$ .

The precision or total mass parameter,  $\alpha$ , of the DP prior can be considered as random, having a gamma distribution,  $\text{Gamma}(a_0, b_0)$ , or fixed at some particular value. When `alpha` is random the method described by Escobar and West (1995) is used. To let  $\alpha$  to be fixed at a particular value, set `a0` to NULL in the prior specification.

The computational implementation of the model is based on the marginalization of the DP and on the use of MCMC methods for conjugate priors (Escobar, 1994; Escobar and West, 1998).

## Value

An object of class `DPlmm` representing the linear mixed-effects model fit. Generic functions such as `print`, `plot`, and `summary` have methods to show the results of the fit. The results include `betaR`, `betaF`, `sigma2e`, `mu`, the elements of `Sigma`, `alpha`, and the number of clusters.

The function `DPrandom` can be used to extract the posterior mean of the random effects.

The list `state` in the output object contains the current value of the parameters necessary to restart the analysis. If you want to specify different starting values to run multiple chains set `status=TRUE` and create the list `state` based on this starting values. In this case the list `state` must include the following objects:

`ncluster` an integer giving the number of clusters,

`alpha` giving the value of the precision parameter,

`b` a matrix of dimension `(nsubjects)*(nrandom effects)` giving the value of the random effects for each subject,

`bclus` a matrix of dimension `(nsubjects)*(nrandom effects)` giving the value of the random effects for each clusters (only the first `ncluster` are considered to start the chain),

`ss` an integer vector defining to which of the `ncluster` clusters each subject belongs,

`beta` giving the value of the fixed effects,

`betaR` giving the mean value of the random effects,

`mu` giving the mean of the normal baseline distributions,

`sigma` giving the variance matrix of the normal baseline distributions, and

`sigma2e` giving the error variance.

## Author(s)

Alejandro Jara <Alejandro.JaraVallejos@med.kuleuven.be>

## References

Escobar, M.D. (1994) Estimating Normal Means with a Dirichlet Process Prior, *Journal of the American Statistical Association*, 89: 268-277.

Escobar, M.D. and West, M. (1995) Bayesian Density Estimation and Inference Using Mixtures. *Journal of the American Statistical Association*, 90: 577-588.

Escobar, M.D. and West, M. (1998) Computing Bayesian Nonparametric Hierarchical Models, in Practical Nonparametric and Semiparametric Bayesian Statistics, eds: D. Dey, P. Muller, D. Sinha, New York: Springer-Verlag, pp. 1-22.

Verbeke, G. and Molenberghs, G. (2000). Linear mixed models for longitudinal data, New York: Springer-Verlag.

### See Also

[DPrandom](#), [DPglmm](#)

### Examples

```
## Not run:
# School Girls Data Example

data(schoolgirls)
attach(schoolgirls)

# Prior information

tinv<-diag(100,2)
prior<-list(a0=0.01,b0=0.01,nu0=2,tau1=0.001,tau2=0.001,
tinv=tinv,mub=rep(0,2),Sb=diag(10000,2))

# Initial state
state <- NULL

# MCMC parameters

nburn<-100
nsave<-50
nskip<-10
ndisplay<-10
mcmc <- list(nburn=nburn,nsave=nsave,nskip=nskip,ndisplay=ndisplay)

# Fit the model: First run

fit1<-DPlmm(fixed=height~1,random=~age|child,prior=prior,mcmc=mcmc,
state=state,status=TRUE)

fit1

# Fit the model: Continuation
state<-fit1$state

fit2<-DPlmm(fixed=height~1,random=~age|child,prior=prior,mcmc=mcmc,
state=state,status=FALSE)

fit2

# Summary with HPD and Credibility intervals
summary(fit2)
summary(fit2,hpd=FALSE)

# Plot model parameters (to see the plots gradually set ask=TRUE)
plot(fit2,ask=FALSE)
plot(fit2,ask=FALSE,nfigr=2,nfigc=2)
```

```

# Plot an specific model parameter (to see the plots gradually set ask=TRUE)
plot (fit2,ask=FALSE,nfigr=1,nfigc=2,param="sigma-(Intercept)")
plot (fit2,ask=FALSE,nfigr=1,nfigc=2,param="ncluster")
## End(Not run)

```

---

DPpsBF

*Computes Pseudo Bayes Factors from DPpackage output*


---

### Description

This function computes Pseudo Bayes Factors from DPpackage output.

### Usage

```
DPpsBF(...)
```

### Arguments

... DPpackage output objects. These have to be of the same class.

### Examples

```

## Not run:
# Respiratory Data Example

data(indon)
attach(indon)

baseage2<-baseage**2
follow<-age-baseage
follow2<-follow**2

# Prior information

beta0<-rep(0,9)
Sbeta0<-diag(1000,9)
tinv<-diag(1,1)
prior<-list(a0=2,b0=0.1,nu0=4,tinv=tinv,mub=rep(0,1),Sb=diag(1000,1),
            beta0=beta0,Sbeta0=Sbeta0)

# Initial state
state <- NULL

# MCMC parameters

nburn<-5
nsave<-100
nskip<-5
ndisplay<-100
mcmc <- list(nburn=nburn,nsave=nsave,nskip=nskip,ndisplay=ndisplay)

# Fit the Probit model
fit1<-DPglm(fixed=infect~gender+height+cosv+sinv+xero+baseage+baseage2+
            follow+follow2,random=~1|id,family=binomial(probit),

```

```

prior=prior,mcmc=mcmc,state=state,status=TRUE)

# Fit the Logit model
fit2<-DPglm(fixed=infect~gender+height+cosv+sinv+xero+baseage+baseage2+
follow+follow2,random=~1|id,family=binomial(logit),
prior=prior,mcmc=mcmc,state=state,status=TRUE)

# Model comparison
DPpsBF(fit1,fit2)

## End(Not run)

```

---

DPrandom

*Extracts Random Effects*


---

## Description

## Usage

```
DPrandom(object, centered=FALSE, predictive=FALSE)
```

## Arguments

object	DP fitted model object from which random effects estimates can be extracted.
centered	logical variable indicating whether the random effects should be extracted centered, bi, or uncentered thetai.
predictive	logical variable indicating whether actual or predictive information of the random effects should be extracted.

## Examples

```

## Not run:
# School Girls Data Example

data(schoolgirls)
attach(schoolgirls)

# Prior information

tinv<-diag(100,2)
prior<-list(a0=0.01,b0=0.01,nu0=2,tau1=0.001,tau2=0.001,
tinv=tinv,mub=rep(0,2),Sb=diag(10000,2))

# Initial state
state <- NULL

# MCMC parameters

nburn<-100
nsave<-50
nskip<-10

```

```

ndisplay<-10
mcmc <- list(nburn=nburn, nsave=nsave, nskip=nskip, ndisplay=ndisplay)

# Fit the model

fit1<-DPlmm(fixed=height~1, random=~age|child, prior=prior, mcmc=mcmc,
            state=state, status=TRUE)

fit1

# Extract random effects

DPrandom(fit1)
DPrandom(fit1, centered=TRUE)

plot(DPrandom(fit1))
plot(DPrandom(fit1, centered=TRUE))

# Extract predictive information of random effects

DPrandom(fit1, predictive=TRUE)
DPrandom(fit1, centered=TRUE, predictive=TRUE)

plot(DPrandom(fit1, predictive=TRUE))
plot(DPrandom(fit1, centered=TRUE, predictive=TRUE))
## End(Not run)

```

---

DPsurvint

*Performs a Bayesian analysis for a semiparametric AFT regression model*


---

## Description

This function generates a posterior density sample from a semiparametric AFT regression model for interval-censored data.

## Usage

```

DPsurvint(formula, prior, mcmc, state, status,
          data=sys.frame(sys.parent()), na.action=na.fail)

```

## Arguments

formula	a two-sided linear formula object describing the fixed-effects part of the model, with the response on the left of a <code>~</code> operator and the terms, separated by <code>+</code> operators, on the right. In the response vector, the unknown limits should be <code>-999</code> .
prior	a list giving the prior information. The list includes the following parameter: <code>a0</code> and <code>b0</code> giving the hyperparameters for prior distribution of the precision parameter of the Dirichlet process prior, <code>alpha</code> giving the value of the precision parameter (it must be specified if <code>a0</code> and <code>b0</code> are missing, see details below), <code>m0</code> and <code>s0</code> giving the mean and variance of the normal prior distribution for the mean of the log normal baseline distribution, and, <code>tau1</code> and <code>tau2</code> giving

	the hyperparameters for the prior distribution of the variance of the log normal baseline distribution, and <code>beta0</code> and <code>Sbeta0</code> giving the hyperparameters of the normal prior distribution for the regression coefficients.
<code>mcmc</code>	a list giving the MCMC parameters. The list must include the following integers: <code>nburn</code> giving the number of burn-in scans, <code>nskip</code> giving the thinning interval, <code>nsave</code> giving the total number of scans to be saved, <code>ndisplay</code> giving the number of saved scans to be displayed on the screen (the function reports on the screen when every <code>ndisplay</code> iterations have been carried out), and <code>tune</code> giving the Metropolis tuning parameter.
<code>state</code>	a list giving the current value of the parameters. This list is used if the current analysis is the continuation of a previous analysis.
<code>status</code>	a logical variable indicating whether this run is new ( <code>TRUE</code> ) or the continuation of a previous analysis ( <code>FALSE</code> ). In the latter case the current value of the parameters must be specified in the object <code>state</code> .
<code>data</code>	data frame.
<code>na.action</code>	a function that indicates what should happen when the data contain NAs. The default action ( <code>na.fail</code> ) causes <code>DPsurvint</code> to print an error message and terminate if there are any incomplete observations.

## Details

This generic function fits a Mixture of Dirichlet process in a AFT regression model for interval censored data (Hanson and Johnson, 2004):

$$T_i = \exp(-X_i\beta)V_i$$

$$V_i|G \sim G$$

$$G|\alpha, G_0 \sim DP(\alpha G_0)$$

where,  $G_0 = \text{LogNormal}(V|\mu, \sigma)$ . To complete the model specification, independent hyperpriors are assumed,

$$\alpha|a_0, b_0 \sim \text{Gamma}(a_0, b_0) \text{ (optional)}$$

$$\mu|m_0, s_0 \sim N(m_0, s_0)$$

$$\sigma^{-1}|\tau_1, \tau_2 \sim \text{Gamma}(\tau_1/2, \tau_2/2)$$

The precision or total mass parameter,  $\alpha$ , of the DP prior can be considered as random, having a gamma distribution,  $\text{Gamma}(a_0, b_0)$ , or fixed at some particular value. When `alpha` is random the method described by Escobar and West (1995) is used. To let `alpha` to be fixed at a particular value, set `a0` to `NULL` in the prior specification.

In the computational implementation of the model, `G` is considered as latent data and sampled partially with sufficient accuracy to be able to generate  $V_1, \dots, V_{n+1}$  which are exactly iid `G`, as proposed by Doss (1994). Both Ferguson's definition of DP and the Sethuraman-Tiwari (1982) representation of the process are used, as described in Hanson and Johnson (2004) to allow the inclusion of covariates.

A Metropolis-Hastings step is used to sample the fully conditional distribution of the regression coefficients and errors (see, Hanson and Johnson, 2004).

## Value

An object of class `DPsurvint` representing the semiparametric AFT regression model fit. Generic functions such as `print`, `plot`, and `summary` have methods to show the results of the fit. The

results include `beta`, `mu`, `sigma`, the precision parameter `alpha`, and the number of clusters.

The function `DPsurvpred` can be used to extract posterior information of the survival curve.

The list `state` in the output object contains the current value of the parameters necessary to restart the analysis. If you want to specify different starting values to run multiple chains set `status=TRUE` and create the list `state` based on this starting values. In this case the list `state` must include the following objects:

`beta` giving the value of the regression coefficients,

`v` giving the value of the errors (it must be consistent with  $y_i = I(V_i < x_i \text{ beta})$ ),

`mu` giving the mean of the lognormal baseline distribution,

`sigma` giving the variance of the lognormal baseline distribution, and

`alpha` giving the value of the precision parameter.

### Author(s)

Alejandro Jara Vallejos (<Alejandro.JaraVallejos@med.kuleuven.be>)

### References

Doss, H. (1994). Bayesian nonparametric estimation for incomplete data using mixtures of Dirichlet priors. *The Annals of Statistics*, 22: 1763 - 1786.

Escobar, M.D. and West, M. (1995) Bayesian Density Estimation and Inference Using Mixtures. *Journal of the American Statistical Association*, 90: 577-588.

Hanson, T., and Johnson, W. (2004) A Bayesian Semiparametric AFT Model for Interval-Censored Data. *Journal of Computational and Graphical Statistics*, 13: 341-361.

Sethuraman, J., and Tiwari, R. C. (1982) Convergence of Dirichlet Measures and the Interpretation of their Parameter, in *Statistical Decision Theory and Related Topics III* (vol. 2), eds. S. S. Gupta and J. O. Berger, New York: Academic Press, pp. 305 - 315.

### See Also

[DPsurvpred](#)

### Examples

```
## Not run:
#####
# A simulated Data Set
#####
ind<-rbinom(100,1,0.5)
vsim<-ind*rnorm(100,1,0.25)+(1-ind)*rnorm(100,3,0.25)
x1<-rep(c(0,1),50)
x2<-rnorm(100,0,1)
etasim<-x1+-1*x2
time<-vsim*exp(-etasim)
y<-matrix(-999,nrow=100,ncol=2)
for(i in 1:100){
  for(j in 1:15){
    if((j-1)<time[i] & time[i]<=j){
      y[i,1]<-j-1
      y[i,2]<-j
    }
  }
}
```

```

    }
    if (time[i]>15) y[i,1]<-15
  }

  # Initial state
  state <- NULL

  # MCMC parameters

  nburn<-50
  nsave<-100
  nskip<-1
  ndisplay<-50
  mcmc <- list (nburn=nburn,nsave=nsave,nskip=nskip,ndisplay=ndisplay,
               tune=0.125)

  # Prior information
  prior <- list (alpha=10,beta0=rep(0,2),Sbeta0=diag(100000,2),m0=0,s0=1,
               tau1=0.01,tau2=0.01)

  # Fit the model

  fit1 <- DPsurvint (y~x1+x2,prior=prior,mcmc=mcmc,state=state,status=TRUE)
  fit1

  # Summary with HPD and Credibility intervals
  summary (fit1)
  summary (fit1,hpd=FALSE)

  # Plot model parameters (to see the plots gradually set ask=TRUE)
  plot (fit1,ask=FALSE)
  plot (fit1,ask=FALSE,nfigr=2,nfigc=2)

  # Plot an specific model parameter (to see the plots gradually set ask=TRUE)
  plot (fit1,ask=FALSE,nfigr=1,nfigc=2,param="x1")
  plot (fit1,ask=FALSE,nfigr=1,nfigc=2,param="mu")
## End(Not run)

```

---

DPsurvpred

*Computes the Survival Curve in a Bayesian analysis for a semiparametric AFT regression model*

---

## Description

This function generates a posterior density sample of the Survival curve from a semiparametric AFT regression model for interval-censored data.

## Usage

```
DPsurvpred (object, grid, xnew=NULL, hpd=TRUE)
```

**Arguments**

<code>object</code>	DPsurvint fitted model object.
<code>grid</code>	a vector of grid points where the survival curve should be evaluated.
<code>xnew</code>	an optional matrix containing the value of the covariables with which to predict. If omitted, the baseline survival information is calculated.
<code>hpd</code>	a logical variable indicating whether a 95HPD interval is calculated, TRUE, or a 95Credibility interval is calculated, FALSE, for the survival curve at each grid point. The default value is TRUE.

**Details**

This function computes the survival curve based on the fit of a Mixture of Dirichlet process in a AFT regression model for interval censored data (Hanson and Johnson, 2004).

A sample of the predictive survival curve for  $X$  is calculated as follows:

$$S(t|X, data) = (1/(\alpha + n))\{\alpha G_0((t * \exp(X\beta), \infty)) + \sum_{i=1}^n \delta_{V_i}((t * \exp(X\beta), \infty))\}$$

**Value**

An object of class `DPsurvpred` representing the survival information arising from a `DPsurvint` model fit. The results include the posterior mean (`pmean`), the posterior median (`pmedian`), the posterior standard deviation (`psd`), the naive standard error (`pstd`) and the limits of the HPD or credibility intervals, `plinf` and `plsup`.

**Author(s)**

Alejandro Jara Vallejos (<Alejandro.JaraVallejos@med.kuleuven.be>)

**References**

Doss, H. (1994). Bayesian nonparametric estimation for incomplete data using mixtures of Dirichlet priors. *The Annals of Statistics*, 22: 1763 - 1786.

Hanson, T., and Johnson, W. (2004) A Bayesian Semiparametric AFT Model for Interval-Censored Data. *Journal of Computational and Graphical Statistics*, 13: 341-361.

**See Also**

[DPsurvint](#)

**Examples**

```
## Not run:
#####
# A simulated Data Set
#####

ind<-rbinom(100,1,0.5)
vsim<-ind*rnorm(100,1,0.25)+(1-ind)*rnorm(100,3,0.25)

x1<-rep(c(0,1),50)
x2<-rnorm(100,0,1)

etasim<-x1+1*x2
```

```

time<-vsim*exp(-etasim)

y<-matrix(-999,nrow=100,ncol=2)

for(i in 1:100){
  for(j in 1:15){
    if((j-1)<time[i] & time[i]<=j){
      y[i,1]<-j-1
      y[i,2]<-j
    }
  }
  if(time[i]>15)y[i,1]<-15
}

# Initial state
state <- NULL

# MCMC parameters

nburn<-50
nsave<-100
nskip<-1
ndisplay<-50
mcmc <- list(nburn=nburn,nsave=nsave,nskip=nskip,ndisplay=ndisplay,
             tune=0.125)

# Prior information
prior <- list(alpha=10,beta0=rep(0,2),Sbeta0=diag(100000,2),m0=0,s0=1,
             tau1=0.01,tau2=0.01)

# Fit the model

fit1 <- DPsurvint(y~x1+x2,prior=prior,mcmc=mcmc,state=state,status=TRUE)
fit1

# Summary with HPD and Credibility intervals
summary(fit1)
summary(fit1,hpd=FALSE)

# Plot model parameters
plot(fit1)
plot(fit1,nfigr=2,nfigc=2)

# Plot an specific model parameter
plot(fit1,ask=FALSE,nfigr=1,nfigc=2,param="x1")
plot(fit1,ask=FALSE,nfigr=1,nfigc=2,param="mu")

# Predictive information for baseline survival
grid<-seq(0.00001,14,0.5)
pred<-DPsurvpred(fit1,grid=grid)

# Plot Baseline information with and without Credibility band
plot(pred)
plot(pred,band=TRUE)

# Predictive information with covariates
npred<-10

```

```

xnew<-cbind(rep(1,npred),seq(-1.5,1.5,length=npred))
xnew<-rbind(xnew,cbind(rep(0,npred),seq(-1.5,1.5,length=npred)))
grid<-seq(0.00001,14,0.5)
pred<-DPsurvpred(fit1,xnew=xnew,grid=grid)

# Plot Baseline information with and without Credibility band
plot(pred,band=TRUE)
## End(Not run)

```

---

fleabeetles

*Flea-beetles*


---

### Description

This data set consider physical information of 74 male flea-beetles reported by Lubischew (1962). Information of three species (*Ch. concinna*, *Ch. heptapotamica*, and *Ch. heikertingeri*) is considered and 6 measurements on each flea-beetles.

### Usage

```
data(fleabeetles)
```

### Format

A data frame with 74 observations on the following 7 variables.

**fjft** a numeric vector giving the width of the first joint of the first tarsus in microns (the sum of measurements for both tarsi)

**sjft** a numeric vector giving the width of the second joint of the first tarsus in microns (the sum of measurements for both tarsi)

**mwhbee** a numeric vector giving the maximal width of the head between the external edges of the eyes in 0.01 mm

**mwafp** a numeric vector giving the maximal width of the aedeagus in the fore-part in microns

**faa** a numeric vector giving the front angle of the aedeagus (1 unit = 7.5 degrees)

**awfs** a numeric vector giving the aedeagus width from the side in microns

**species** a numeric vector giving the species: 1=*Ch. concinna*, 2= *Ch. heptapotamica*, and 3=*Ch. heikertingeri*

### Source

Lubischew, A. A. (1962) On the Use of Discriminant Functions in Taxonomy, *Biometrics*, 18: 455-477.

### References

MacEachern, S.N., and Muller, P. (1998) Estimating Mixture of Dirichlet Process Models, *Journal of Computational and Graphical Statistics*, 7: 223-238.

### Examples

```

data(fleabeetles)
## maybe str(fleabeetles) ; plot(fleabeetles) ...

```

---

galaxy

*Galaxy velocities*

---

### Description

This data set consider physical information on velocities (km/second) for 82 galaxies reported by Roeder (1990). These are drawn from six well-separated conic sections of the Corona Borealis region.

### Usage

```
data(galaxy)
```

### Format

A data frame with 82 observations on the following variable.

**speed** a numeric vector giving the speed of galaxies ((km/second))

### Source

Roeder, K. (1990) Density estimation with confidence sets exemplified by superclusters and voids in the galaxies, *Journal of the American Statistical Association*, 85: 617-624.

### References

Escobar, M.D. and West, M. (1995) Bayesian Density Estimation and Inference Using Mixtures. *Journal of the American Statistical Association*, 90: 577-588.

### Examples

```
data(galaxy)
## maybe str(galaxy) ; plot(galaxy) ...
```

---

indon

*Indonesian Children's Health Study*

---

### Description

This data set consider respiratory infection information of 250 indonesian children reported by Sommer, Katz, and Tarwotjo (1984). The children, all preschoolers, were seen quarterly for up to six quarters. At each examination, the presence or absence of respiratory infection was noted.

### Usage

```
data(indon)
```

**Format**

A data frame with 1200 observations on the following 9 variables.

**id** an ordered factor giving a unique identifier for the subject in the study.

**gender** a numeric vector giving the gender.

**height** a numeric vector giving the height for age as a percentage of the National Center for Health Statistics standard centered at 90.

**cosv** a numeric vector giving the seasonal cosine for the annual cycle.

**sinv** a numeric vector giving the seasonal sine for the annual cycle.

**xero** a numeric vector giving the presence (1) or absence (0) of xerophthalmia.

**baseage** a numeric vector giving the age at the entry.

**age** a numeric vector giving the age of the child in months centered at 36

**infect** a numeric vector giving the presence (1) or absence (0) of respiratory infection.

**Source**

Sommer, A., Katz, J., and Tarwotjo, I. (1984) Increased risk of respiratory infection and diarrhea in children with pre-existing mild vitamin A deficiency, *American Journal of Clinical Nutrition*, 40: 1090-1095.

**References**

Zeger, S.L., and Karim, M.R. (1991) Generalized linear models with random effects: A Gibbs sampling approach. *Journal of the American Statistical Association*, 86: 79-86.

**Examples**

```
data(indon)
## maybe str(indon) ; plot(indon) ...
```

---

rats

*Rats*

---

**Description**

This example is taken from section 6 of Gelfand and Smith (1990), and concerns 30 young rats whose weights were measured weekly for five weeks.

**Usage**

```
data(rats)
```

**Format**

A data frame with 150 observations on the following 3 variables.

**weight** a numeric vector giving the weight of the rat

**day** a numeric vector giving the day of the weight evaluation

**rat** an ordered factor giving a unique identifier for the subject in the study

**Source**

Gelfand, A.E. (with S. Hills, A. Racine-Poon and A.F.M. Smith) 1990. Illustration of Bayesian Inference in Normal Data Models Using Gibbs Sampling. *Journal Amer. Stat. Assoc.*, 85, 972-985.

**References**

Gelfand, A.E. (with S. Hills, A. Racine-Poon and A.F.M. Smith) 1990. Illustration of Bayesian Inference in Normal Data Models Using Gibbs Sampling. *Journal Amer. Stat. Assoc.*, 85, 972-985.

**Examples**

```
data(rats)
## maybe str(rats) ; plot(rats) ...
```

---

schoolgirls                      *The Heights of Schoolgirls*

---

**Description**

This data set consider growth information of 20 preadolescent schoolgirls reported by Goldstein (1979, Table 4.3, p. 101). The height of girls was measured on a yearly basis from age 6 to 10. The measurements are given at exact years of age, some having been previously adjusted to these. Further, the girls were classified according to the height of their mother into three categories: short mothers, medium mothers and tall mothers.

**Usage**

```
data(schoolgirls)
```

**Format**

A data frame with 100 observations on the following 4 variables.

**height** a numeric vector giving the height in cm

**child** an ordered factor giving a unique identifier for the subject in the study

**age** a numeric vector giving the age of the child in years

**group** a factor with levels 1 (short), 2 (medium), and 3 (tall) giving the mother category

**Details**

Measurements reported by Goldstein(1979) for one of the girls (child 5) were 114.5, 112.0, 126.4, 131.2, and 135.0. In this data set, the second measurement was replaced by 122.0.

**Source**

Goldstein, H. (1979) *The Design and Analysis of Longitudinal Studies*. London: Academic Press.

**References**

Verbeke, G., and Molenberghs, G. (2000) *Linear Mixed Models for Longitudinal Data*. New York: Springer-Verlag.(Section 12.7)

## Examples

```
data(schoolgirls)
## maybe str(schoolgirls) ; plot(schoolgirls) ...
```

---

seizures

*Epileptic seizures*

---

## Description

This data set consider information from a clinical trial of 59 epileptics, reported by Thall and Vail (1990). For each patient, the number of epileptic seizures was recorded during a baseline period of eight weeks. Patients were then randomized to treatment with the anti-epileptic drug progabide, or to a placebo in addition to standard chemotherapy. The number of seizures was then recorded in four consecutive two-weeks intervals.

## Usage

```
data(seizures)
```

## Format

A data frame with 295 observations on the following 5 variables.

**id** an ordered factor giving a unique identifier for the subject in the study.

**seize** a numeric vector giving the number of epileptic seizures.

**visit** a numeric vector giving the number of the visit, 0=baseline, and 1,2,3, and 4 for the four consecutive two-weeks intervals.

**trt** a numeric vector giving the treatment group.

**age** a numeric vector giving the age at the entry.

## Source

Thall, P.F., and Vail, S.C. (1990) Some covariance models for longitudinal count data with overdispersion, *Biometrics*, 46: 657-671.

## References

Diggle, P.J., Liang, K-Y., and Zeger, S.L. (1994) *Analysis of longitudinal data*. Oxford: Clarendon Press.

## Examples

```
data(seizures)
## maybe str(seizures) ; plot(seizures) ...
```

# Index

## \*Topic **datasets**

fleabeetles, 25  
galaxy, 26  
indon, 27  
rats, 28  
schoolgirls, 28  
seizures, 29

## \*Topic **models**

DPbinary, 1  
DPdensity, 4  
DPelicit, 8  
DPglmm, 10  
DPlmm, 14  
DPpsBF, 17  
DPrandom, 18  
DPsurvint, 19  
DPsurvpred, 23

DPbinary, 1  
DPdensity, 4  
DPelicit, 8  
DPglmm, 10, 16  
DPlmm, 13, 14  
DPpsBF, 17  
DPrandom, 7, 13, 16, 18  
DPsurvint, 19, 24  
DPsurvpred, 21, 23

fleabeetles, 25

galaxy, 26

indon, 27

rats, 28

schoolgirls, 28

seizures, 29