

# Package ‘hdpGLM’

May 8, 2026

**Title** Hierarchical Dirichlet Process Generalized Linear Models

**Version** 1.0.5

**Description** Implementation of MCMC algorithms to estimate the Hierarchical Dirichlet Process Generalized Linear Model (hdpGLM) presented in the paper Ferrari (2020) Modeling Context-Dependent Latent Heterogeneity, Political Analysis <DOI:10.1017/pan.2019.13> and <doi:10.18637/jss.v107.i10>.

**Depends** R (>= 3.3.3)

**License** MIT + file LICENSE

**URL** <https://github.com/DiogoFerrari/hdpGLM>

**BugReports** <https://github.com/DiogoFerrari/hdpGLM/issues>

**Encoding** UTF-8

**LazyData** true

**LinkingTo** Rcpp, RcppArmadillo

**Imports** coda, data.table, dplyr, formula.tools, ggplot2, stringr, ggridges, ggpubr, Hmisc, LaplacesDemon, magrittr, methods, MASS, mvtnorm, Rcpp, purrr, tibble, tidy

**RoxygenNote** 7.3.2

**Suggests** knitr, rmarkdown

**VignetteBuilder** knitr

**NeedsCompilation** yes

**Author** Diogo Ferrari [aut, cre]

**Maintainer** Diogo Ferrari <diogoferrari@gmail.com>

**Repository** CRAN

**Date/Publication** 2025-10-17 02:50:02 UTC

## Contents

classify . . . . .	2
coef.dpGLM . . . . .	3

coef.hdpGLM . . . . .	3
hdpGLM . . . . .	4
hdpGLM_classify . . . . .	7
hdpGLM_simParameters . . . . .	8
hdpGLM_simulateData . . . . .	9
mcmc_info.dpGLM . . . . .	11
mcmc_info.hdpGLM . . . . .	11
nclusters . . . . .	11
plot.dpGLM . . . . .	12
plot.hdpGLM . . . . .	14
plot_beta . . . . .	16
plot_beta_sim . . . . .	17
plot_hdpglm . . . . .	18
plot_pexp_beta . . . . .	20
plot_tau . . . . .	22
predict.dpGLM . . . . .	25
predict.hdpGLM . . . . .	25
print.dpGLM . . . . .	26
print.dpGLM_data . . . . .	27
print.hdpGLM . . . . .	27
print.hdpGLM_data . . . . .	28
summary.dpGLM . . . . .	28
summary.dpGLM_data . . . . .	29
summary.hdpGLM . . . . .	29
summary.hdpGLM_data . . . . .	30
summary_tidy . . . . .	31
welfare . . . . .	31
welfare2 . . . . .	32

## Index 33

---

classify	<i>Classify data points</i>
----------	-----------------------------

---

### Description

This function returns a data frame with the data points classified according to the estimation of cluster probabilities generated by the output of the function [hdpGLM](#)

### Usage

```
classify(data, samples)
```

### Arguments

data	a data frame with the data set used to estimate the <a href="#">hdpGLM</a> model
samples	the output of <a href="#">hdpGLM</a>

---

coef.dpGLM	<i>Extract dpGLM fitted coefficients</i>
------------	--

---

**Description**

This function gives the posterior mean of the coefficients

**Usage**

```
## S3 method for class 'dpGLM'  
coef(object, ...)
```

**Arguments**

object            a dpGLM object returned by the function hdpGLM  
...                The additional parameters accepted are:

---

coef.hdpGLM	<i>Extract hdpGLM fitted coefficients</i>
-------------	---

---

**Description**

This function gives the posterior mean of the coefficients

**Usage**

```
## S3 method for class 'hdpGLM'  
coef(object, ...)
```

**Arguments**

object            a dpGLM object returned by the function hdpGLM  
...                The additional parameters accepted are:

hdpGLM

*hdpGLM: A package for computing Hierarchical Dirichlet Process Generalized Linear Models*

## Description

Further information is available at: <http://www.diogoferrari.com/hdpGLM/index.html>

References:

- Ferrari, D. (2020). Modeling Context-Dependent Latent Effect Heterogeneity. *Political Analysis*, 28(1), 20–46.
- Mukhopadhyay, S., & Gelfand, A. E. (1997). Dirichlet Process Mixed Generalized Linear Models. *Journal of the American Statistical Association*, 92(438), 633–639.
- Hannah, L. A., Blei, D. M., & Powell, W. B. (2011). Dirichlet Process Mixtures of Generalized Linear Models. *Journal of Machine Learning Research*, 12(Jun), 1923–1953.
- Heckman, J. J., & Vytlacil, E. J. (2007). Econometric Evaluation of Social Programs, Part I: Causal Models, Structural Models and Econometric Policy Evaluation. *Handbook of Econometrics*, 6(), 4779–4874.

The function estimates a semi-parametric mixture of Generalized Linear Models. It uses a (hierarchical) Dependent Dirichlet Process Prior for the mixture probabilities.

## Usage

```
hdpGLM(
  formula1,
  formula2 = NULL,
  data,
  mcmc,
  family = "gaussian",
  K = 100,
  context.id = NULL,
  constants = NULL,
  weights = NULL,
  n.display = 1000,
  na.action = "exclude",
  imp.bin = "R"
)
```

## Arguments

- |          |  |
|----------|--|
| formula1 | a single symbolic description of the linear model of the mixture GLM components to be fitted. The syntax is the same as used in the <code>lm</code> function.  |
| formula2 | either NULL (default) or a single symbolic description of the linear model of the hierarchical component of the model. It specifies how the average parameter of the base measure of the Dirichlet Process Prior varies linearly as a function of group level covariates. If NULL, it will use a single base measure to the DPP mixture model. |

<code>data</code>	a data.frame with all the variables specified in <code>formula1</code> and <code>formula2</code> . Note: it is advisable to scale the variables before the estimation
<code>mcmc</code>	a named list with the following elements <ul style="list-style-type: none"> <li>- <code>burn.in</code> (required): an integer greater or equal to 0 indicating the number iterations used in the burn-in period of the MCMC.</li> <li>- <code>n.iter</code> (required): an integer greater or equal to 1 indicating the number of iterations to record after the burn-in period for the MCMC.</li> <li>- <code>epsilon</code> (optional): a positive number. Default is 0.01. Used when <code>family='binomial'</code> or <code>family='multinomial'</code>. It is used in the Stormer-Verlet Integrator (a.k.a leapfrog integrator) to solve the Hamiltonian Monte Carlo in the estimation of the model.</li> <li>- <code>leapFrog</code> (optional) an integer. Default is 40. Used when <code>family='binomial'</code> or <code>family='multinomial'</code>. It indicates the number of steps taken at each iteration of the Hamiltonian Monte Carlo for the Stormer-Verlet Integrator.</li> <li>- <code>hmc_iter</code> (optional) an integer. Default is 1. Used when <code>family='binomial'</code> or <code>family='multinomial'</code>. It indicates the number of HMC iteration(s) for each Gibbs iteration.</li> </ul>
<code>family</code>	a character with either <code>'gaussian'</code> , <code>'binomial'</code> , or <code>'multinomial'</code> . It indicates the family of the GLM components of the mixture model.
<code>K</code>	an integer indicating the maximum number of clusters to truncate the Dirichlet Process Prior in order to use the blocked Gibbs sampler.
<code>context.id</code>	string with the name of the column in the data that uniquely identifies the contexts. If NULL (default) contexts will be identified by numerical indexes and unique context-level variables. The user is advised to pre-process the data to provide meaningful labels for the contexts to facilitate later visualization and analysis of the results.
<code>constants</code>	either NULL or a list with the constants of the model. If not NULL, it must contain a vector named <code>mu_beta</code> , whose size must be equal to the number of covariates specified in <code>formula1</code> plus one for the constant term; <code>Sigma_beta</code> , which must be a squared matrix, and each dimension must be equal to the size of the vector <code>mu_beta</code> ; and <code>alpha</code> , which must be a single number. If <code>@param family</code> is <code>'gaussian'</code> , then it must also contain <code>s2_sigma</code> and <code>df_sigma</code> , both single numbers. If NULL, the defaults are <code>mu_beta=0</code> , <code>Sigma_beta=diag(10)</code> , <code>alpha=1</code> , <code>df_sigma=10</code> , <code>s2_sigma=10</code> (all with the dimension automatically set to the correct values).
<code>weights</code>	numeric vector with the same size as the number of rows of the data. It must contain the weights of the observations in the data set. NOTE: FEATURE NOT IMPLEMENTED YET
<code>n.display</code>	an integer indicating the number of iterations to wait before printing information about the estimation process. If zero, it does not display any information. Note: displaying information at every iteration ( <code>n.display=1</code> ) may increase the time to estimate the model slightly.
<code>na.action</code>	string with action to be taken for the NA values. (currently, only <code>exclude</code> is available)
<code>imp.bin</code>	string, either <code>"R"</code> or <code>"C++"</code> indicating the language of the implementation of the binomial model.

## Details

The package implements a hierarchical Dirichlet process Generalized Linear Model as proposed in Ferrari (2020) Modeling Context-Dependent Latent Effect Heterogeneity, which expands the non-parametric Bayesian models proposed in Mukhopadhyay and Gelfand (1997), Hannah (2011), and Heckman and Vytlačil (2007) to deal with context-dependent cases. The package can be used to estimate latent heterogeneity in the marginal effect of GLM linear coefficients, to cluster data points based on that latent heterogeneity, and to investigate the occurrence of Simpson's Paradox due to latent or omitted features.

This function estimates a Hierarchical Dirichlet Process generalized linear model, which is a semi-parametric Bayesian approach to regression estimation with clustering. The estimation is conducted using Blocked Gibbs Sampler if the output variable is gaussian distributed. It uses Metropolis-Hastings inside Gibbs if the output variable is binomial or multinomial distributed. This is specified using the parameter family. See:

Ferrari, D. (2020). Modeling Context-Dependent Latent Effect Heterogeneity, *Political Analysis*, 28(1), 20–46. doi:10.1017/pan.2019.13.

Ferrari, D. (2023). "hdpGLM: An R Package to Estimate Heterogeneous Effects in Generalized Linear Models Using Hierarchical Dirichlet Process." *Journal of Statistical Software*, 107(10), 1-37. doi:10.18637/jss.v107.i10.

Ishwaran, H., & James, L. F., Gibbs sampling methods for stick-breaking priors, *Journal of the American Statistical Association*, 96(453), 161–173 (2001).

Neal, R. M., Markov chain sampling methods for dirichlet process mixture models, *Journal of computational and graphical statistics*, 9(2), 249–265 (2000).

## Value

The function returns a list with elements `samples`, `pik`, `max_active`, `n.iter`, `burn.in`, and `time.elapsed`. The `samples` element contains a MCMC object (from `coda` package) with the samples from the posterior distribution. The `pik` is a  $n \times K$  matrix with the estimated probabilities that the observation  $i$  belongs to the cluster  $k$ .

## Author(s)

**Maintainer:** Diogo Ferrari <diogoferrari@gmail.com>

## See Also

Useful links:

- <https://github.com/DiogoFerrari/hdpGLM>
- Report bugs at <https://github.com/DiogoFerrari/hdpGLM/issues>

## Examples

```
## Note: this example is for illustration. You can run the example
## manually with increased number of iterations to see the actual
## results, as well as the data size (n)
```

```
set.seed(10)
n = 300
data = tibble::tibble(x1 = rnorm(n, -3),
                      x2 = rnorm(n, 3),
                      z = sample(1:3, n, replace=TRUE),
                      y = I(z==1) * (3 + 4*x1 - x2 + rnorm(n)) +
                        I(z==2) * (3 + 2*x1 + x2 + rnorm(n)) +
                        I(z==3) * (3 - 4*x1 - x2 + rnorm(n))
                      )

mcmc = list(burn.in = 0, n.iter = 20)
samples = hdpGLM(y~ x1 + x2, data=data, mcmc=mcmc, family='gaussian',
                n.display=30, K=50)

summary(samples)
plot(samples)
plot(samples, separate=TRUE)

## compare with GLM
## lm(y~ x1 + x2, data=data, family='gaussian')
```

---

hdpGLM_classify	<i>Deprecated</i>
-----------------	-------------------

---

## Description

Deprecated

## Usage

```
hdpGLM_classify(data, samples)
```

## Arguments

data	a data frame with the data set used to estimate the <a href="#">hdpGLM</a> model
samples	the output of <a href="#">hdpGLM</a>

---

hdpGLM\_simParameters *Simulate the parameters of the model*

---

### Description

This function generates parameters that can be used to simulate data sets from the Hierarchical Dirichlet Process of Generalized Linear Model (hdpGLM) or dpGLM

### Usage

```
hdpGLM_simParameters(
  K,
  nCov = 2,
  nCovj = 0,
  J = 1,
  pi = NULL,
  same.K = FALSE,
  seed = NULL,
  context.effect = NULL,
  same.clusters.across.contexts = NULL,
  context.dependent.cluster = NULL
)
```

### Arguments

K	integer, the number of clusters. If there are multiple contexts, K is the average number of clusters across contexts, and each context gets a number of clusters sampled from a Poisson distribution, except if same.K is TRUE.
nCov	integer, the number of covariates of the GLM components
nCovj	an integer indicating the number of covariates determining the average parameter of the base measure of the Dirichlet process prior
J	an integer representing the number of contexts @param parameters either NULL or a list with the parameters to generate the model. If not NULL, it must contain a sublist name beta, a vector named tau, and a vector named pi. The sublist beta must be a list of vectors, each one with size nCov+1 to be the coefficients of the GLM mixtures components that will generate the data. For the vector tau, if nCovj=0 (single-context case) then it must be a 1x1 matrix containing 1. If nCovj>0, it must be a (nCov+1)x(nCovj+1) matrix. The vector pi must add up to 1 and have length K.
pi	either NULL or a vector with length K that add up to 1. If not NULL, it determines the mixture probabilities
same.K	boolean, used when data is sampled from more than one context. If TRUE all contexts get the same number of clusters. If FALSE, each context gets a number of clusters sampled from a Poisson distribution with expectation equals to K (current not implemented)

`seed` a seed for `set.seed`  
`context.effect` either NULL or a two dimensional integer vector. If it is NULL, all the coefficients (beta) of the individual level covariates are functions of context-level features (tau). If it is not NULL, the first component of the vector indicates the index of the lower level covariate (X) whose linear effect beta depends on context (tau) (0 is the intercept). The second component indicates the index context-level covariate (W) whose linear coefficient (tau) is non-zero.  
`same.clusters.across.contexts` boolean, if TRUE all the contexts will have the same number of clusters AND each cluster will have the same coefficient beta.  
`context.dependent.cluster` integer, indicates which cluster will be context-dependent. If zero, all clusters will be context-dependent

### Value

The function returns a list with the parameters used to generate data sets from the hdpGLM model. This list can be used in the function `hdpGLM_simulateData`

### Examples

```
pars = hdpGLM_simParameters(nCov=2, K=2, nCovj=3, J=20,
  same.clusters.across.contexts=FALSE, context.dependent.cluster=0)
```

---

`hdpGLM_simulateData` *Simulate a Data Set from hdpGLM*

---

### Description

Simulate a Data Set from hdpGLM

### Usage

```
hdpGLM_simulateData(
  n,
  K,
  nCov = 2,
  nCovj = 0,
  J = 1,
  family = "gaussian",
  parameters = NULL,
  pi = NULL,
  same.K = FALSE,
  seed = NULL,
  context.effect = NULL,
  same.clusters.across.contexts = NULL,
  context.dependent.cluster = NULL
)
```

**Arguments**

<code>n</code>	integer, the sample size of the data. If there are multiple contexts, each context will have <code>n</code> cases.
<code>K</code>	integer, the number of clusters. If there are multiple contexts, <code>K</code> is the average number of clusters across contexts, and each context gets a number of clusters sampled from a Poisson distribution, except if <code>same.K</code> is TRUE.
<code>nCov</code>	integer, the number of covariates of the GLM components.
<code>nCovj</code>	an integer indicating the number of covariates determining the average parameter of the base measure of the Dirichlet process prior
<code>J</code>	an integer representing the number of contexts @param parameters either NULL or a list with the parameters to generate the model. If not NULL, it must contain a sublist name <code>beta</code> , a vector named <code>tau</code> , and a vector named <code>pi</code> . The sublist <code>beta</code> must be a list of vectors, each one with size <code>nCov+1</code> to be the coefficients of the GLM mixtures components that will generate the data. For the vector <code>tau</code> , if <code>nCovj=0</code> (single-context case) then it must be a 1x1 matrix containing 1. If <code>nCovj&gt;0</code> , it must be a $(nCov+1) \times (nCovj+1)$ matrix. The vector <code>pi</code> must add up to 1 and have length <code>K</code> .
<code>family</code>	a character with either 'gaussian', 'binomial', or 'multinomial'. It indicates the family of the GLM components of the mixture model.
<code>parameters</code>	a list with the parameter values of the model. Format should be the same of the output of the function <code>hdpGLM_simulateParameters()</code>
<code>pi</code>	either NULL or a vector with length <code>K</code> that add up to 1. If not NULL, it determines the mixture probabilities
<code>same.K</code>	boolean, used when data is sampled from more than one context. If TRUE all contexts get the same number of clusters. If FALSE, each context gets a number of clusters sampled from a Poisson distribution with expectation equals to <code>K</code> (current not implemented)
<code>seed</code>	a seed for <code>set.seed</code>
<code>context.effect</code>	either NULL or a two dimensional integer vector. If it is NULL, all the coefficients ( <code>beta</code> ) of the individual level covariates are functions of context-level features ( <code>tau</code> ). If it is not NULL, the first component of the vector indicates the index of the lower level covariate ( <code>X</code> ) whose linear effect <code>beta</code> depends on context ( <code>tau</code> ) (0 is the intercept). The second component indicates the index context-level covariate ( <code>W</code> ) whose linear coefficient ( <code>tau</code> ) is non-zero.
<code>same.clusters.across.contexts</code>	boolean, if TRUE all the contexts will have the same number of clusters AND each cluster will have the same coefficient <code>beta</code> .
<code>context.dependent.cluster</code>	integer, indicates which cluster will be context-dependent. If zero, all clusters will be context-dependent

---

mcmc_info.dpGLM	<i>mcmc</i>
-----------------	-------------

---

**Description**

Generic method to return the MCMC information

**Usage**

```
mcmc_info.dpGLM(x, ...)
```

**Arguments**

x	a dpGLM object returned by the function hdpGLM
...	ignore

---

mcmc_info.hdpGLM	<i>mcmc</i>
------------------	-------------

---

**Description**

Generic method to return the MCMC information

**Usage**

```
mcmc_info.hdpGLM(x, ...)
```

**Arguments**

x	a hdpGLM object returned by the function hdpGLM
...	ignore

---

nclusters	<i>nclusters</i>
-----------	------------------

---

**Description**

This function returns the number of clusters found in the estimation

**Usage**

```
nclusters(object)
```

**Arguments**

object	a dpGLM object returned by the function hdpGLM
--------	--

plot.dpGLM

*Default plot for class dpGLM***Description**

This function generates density plots with the posterior distribution generated by the function [hdpGLM](#)

**Usage**

```
## S3 method for class 'dpGLM'
plot(
  x,
  terms = NULL,
  separate = FALSE,
  hpd = TRUE,
  true.beta = NULL,
  title = NULL,
  subtitle = NULL,
  adjust = 1,
  ncols = NULL,
  only.occupied.clusters = TRUE,
  focus.hpd = FALSE,
  legend.position = "top",
  colour = "grey",
  alpha = 0.4,
  display.terms = TRUE,
  plot.mean = TRUE,
  legend.label.true.value = "True",
  ...
)
```

**Arguments**

x	a dpGLM object with the samples from generated by <a href="#">hdpGLM</a>
terms	string vector with the name of covariates to plot. If NULL (default), all covariates are plotted.
separate	boolean, if TRUE the linear coefficients beta will be displayed in their separate clusters.
hpd	boolean, if TRUE and separate=T, the 95% HPDI lines will be displayed.
true.beta	either NULL (default) or a data.frame with the true values of the linear coefficients beta if they are known. The data.frame must contain a column named k indicating the cluster of beta, and a column named Parameter with the name of the linear coefficients (beta1, beta2, ..., beta_dx, where dx is the number of covariates at the individual level, and beta1 is the coefficient of the intercept term). It must contain a column named True with the true value of the betas.

title	string, the title of the plot
subtitle	string, the subtitle of the plot
adjust	the bandwidth used is actually $\text{adjust} \times \text{bw}$ . This makes it easy to specify values like 'half the default' bandwidth.
ncols	integer, the number of columns in the plot
only.occupied.clusters	boolean, if TRUE it shows only the densities of the clusters that actually have data points assigned to it with high probability
focus.hpd	boolean, if TRUE and separate is also TRUE it will display only the 95% HPDI of the posterior density of the linear coefficients beta
legend.position	one of four options: "bottom" (default), "top", "left", or "right". It indicates the position of the legend
colour	= string with color to fill the density plot
alpha	number between 0 and 1 indicating the degree of transparency of the density
display.terms	boolean, if TRUE (default), the covariate name is displayed in the plot
plot.mean	boolean, if TRUE the posterior mean of every cluster is displayed
legend.label.true.value	a string with the value to display in the legend when the true.beta is used
...	ignored

## Examples

```
# Note: this example is just for illustration. MCMC iterations are very reduced
set.seed(10)
n = 20
data = tibble::tibble(x1 = rnorm(n, -3),
                      x2 = rnorm(n, 3),
                      z = sample(1:3, n, replace=TRUE),
                      y = I(z==1) * (3 + 4*x1 - x2 + rnorm(n)) +
                        I(z==2) * (3 + 2*x1 + x2 + rnorm(n)) +
                        I(z==3) * (3 - 4*x1 - x2 + rnorm(n)) ,
                      )

## estimation
mcmc = list(burn.in=1, n.iter=50)
samples = hdpGLM(y ~ x1 + x2, data=data, mcmc=mcmc, n.display=1)

plot(samples)
```

plot.hdpGLM

*Plot***Description**

Generic function to plot the posterior density estimation produced by the function `hdpGLM`

**Usage**

```
## S3 method for class 'hdpGLM'
plot(
  x,
  terms = NULL,
  j.label = NULL,
  j.idx = NULL,
  title = NULL,
  subtitle = NULL,
  true.beta = NULL,
  ncol = NULL,
  legend.position = "bottom",
  display.terms = TRUE,
  context.id = NULL,
  ylab = NULL,
  xlab = NULL,
  x.axis.size = 1.1,
  y.axis.size = 1.1,
  title.size = 1.2,
  panel.title.size = 1.5,
  legend.size = 1.1,
  rel.height = 0.01,
  fill.col = "#00000044",
  border.col = "white",
  ...
)
```

**Arguments**

<code>x</code>	an object of the class <code>hdpGLM</code> generated by the function <a href="#">hdpGLM</a>
<code>terms</code>	string vector with the name of the individual-level covariates to plot. If <code>NULL</code> (default), all covariates are plotted.
<code>j.label</code>	string vector with the names of the contexts to plot. An alternative is to use the context indexes with the parameter <code>j.idx</code> instead of the context labels. If <code>NULL</code> (default) and <code>j.idx</code> is also <code>NULL</code> , the posterior distribution of all contexts are plotted. Note: if contexts to plot are selected using <code>j.label</code> , the parameter <code>context.id</code> must also be provided.

j.idx	integer vector with the index of the contexts to plot. An alternative is to use the context labels with the parameter j.label instead of the indexes. If NULL (default) and j.label is also NULL, the posterior distribution of all contexts are plotted
title	string, the title of the plot
subtitle	string, the subtitle of the plot
true.beta	a data.frame with the true values of the linear coefficients beta if they are known. The data.frame must contain a column named j with the index of the context associated with that particular linear coefficient beta. It must match the indexes used in the data set for each context. Another column named k must be provided, indicating the cluster of beta, and a column named Parameter with the name of the linear coefficients (beta1, beta2, ..., beta_dx, where dx is the number of covariates at the individual level, and beta1 is the coefficient of the intercept term). It must contain a column named True with the true value of the betas. Finally, the data.frame must contain columns with the context-level covariates as used in the estimation of the <a href="#">hdpGLM</a> function (see Details below).
ncol	integer, the number of columns in the plot
legend.position	one of four options: "bottom" (default), "top", "left", or "right". It indicates the position of the legend
display.terms	boolean, if TRUE (default), the covariate name is displayed in the plot
context.id	string with the name of the column containing the labels identifying the contexts. This variable should have been specified when the estimation was conducted using the function <a href="#">hdpGLM</a> .
ylab	string, the label of the y-axis
xlab	string, the label of the x-axis
x.axis.size	numeric, the relative size of the label in the x-axis
y.axis.size	numeric, the relative size of the label in the y-axis
title.size	numeric, the relative size of the title of the plot
panel.title.size	numeric, the relative size of the titles in the panel of the plot
legend.size	numeric, the relative size of the legend
rel.height	see <code>ggridges::geom_density_ridges</code>
fill.col	string with the color of the densities
border.col	string with the color of the border of the densities
...	Additional arguments accepted are: true.beta: a data.frame with the true values of the linear coefficients beta if they are known. The data.frame must contain a column named j with the index of the context associated with that particular linear coefficient beta. It must match the indexes used in the data set for each context. Another column named k must be provided, indicating the cluster of beta, and a column named Parameter with the name of the linear coefficients (beta1, beta2, ..., beta_dx,

where `dx` is the number of covariates at the individual level, and `beta1` is the coefficient of the intercept term). It must contain a column named `True` with the true value of the betas. Finally, the `data.frame` must contain columns with the context-level covariates as used in the estimation of the [hdpGLM](#) function (see Details below).

`true.tau`: a `data.frame` with four columns. The first must be named `w` and it indicates the index of each context-level covariate, starting with 0 for the intercept term. The second column named `beta` must contain the indexes of the betas of individual-level covariates, starting with 0 for the intercept term. The third column named `Parameter` must be named `tau<w><beta>`, where `w` and `beta` must be the actual values displayed in the columns `w` and `beta`. Finally, it must have a column named `True` with the true value of the parameter.

---

plot\_beta

*Plot beta posterior distribution*

---

### Description

Plot the posterior distribution of the linear parameters `beta` for each context

### Usage

```
plot_beta(
  samples,
  X = NULL,
  context.id = NULL,
  true.beta = NULL,
  title = NULL,
  subtitle = NULL,
  plot.mean = FALSE,
  plot.grid = FALSE,
  showKhat = FALSE,
  col = NULL,
  xlab.size = NULL,
  ylab.size = NULL,
  title.size = NULL,
  legend.size = NULL,
  xtick.distance = NULL,
  left.margin = 0,
  ytick.distance = NULL,
  col.border = "white"
)
```

### Arguments

`samples` an output of the function [hdpGLM](#)

X	a string vector with the name of the first-level covariates whose associated tau should be displayed
context.id	string with the name of the column containing the labels identifying the contexts. This variable should have been specified when the estimation was conducted using the function <code>hdpGLM</code> .
true.beta	a <code>data.frame</code> with the true values of the linear coefficients beta if they are known. The <code>data.frame</code> must contain a column named <code>j</code> with the index of the context associated with that particular linear coefficient beta. It must match the indexes used in the data set for each context. Another column named <code>k</code> must be provided, indicating the cluster of beta, and a column named <code>Parameter</code> with the name of the linear coefficients ( <code>beta1</code> , <code>beta2</code> , ..., <code>beta_dx</code> , where <code>dx</code> is the number of covariates at the individual level, and <code>beta1</code> is the coefficient of the intercept term). It must contain a column named <code>True</code> with the true value of the betas. Finally, the <code>data.frame</code> must contain columns with the context-level covariates as used in the estimation of the <code>hdpGLM</code> function (see Details below).
title	string, title of the plot
subtitle	string, the subtitle of the plot
plot.mean	boolean, if TRUE the posterior mean of every cluster is displayed
plot.grid	boolean, if TRUE a grid is displayed in the background
showKhat	boolean, if TRUE a message with the number of estimated clusters by context is displayed
col	string, color of the densities
xlab.size	numeric, size of the breaks in the x-axis
ylab.size	numeric, size of the breaks in the y-axis
title.size	numeric, size of the title
legend.size	numeric, size of the legend
xtick.distance	numeric, distance between x-axis marks and bottom of the figure
left.margin	numeric, distance between left margin and left side of the figure
ytick.distance	numeric, distance between y-axis marks and bottom of the figure
col.border	string, color of the border of the densities

---

plot\_beta\_sim

*Plot simulated data*


---

### Description

Create a plot with the beta sampled from its distribution, as a function of context-level feature  $W$ . Only works for the hierarchical model (`hdpGLM`), not the `dpGLM`

### Usage

```
plot_beta_sim(data, w.idx, ncol = NULL)
```

**Arguments**

data	the output of the function <code>hdpGLM_simulateData</code>
w.idx	integer, the index of the context level covariate the plot
ncol	integer, the number of columns in the grid of the plot

---

plot_hdpglm	<i>Plot posterior distributions</i>
-------------	-------------------------------------

---

**Description**

this function creates a plot with two grids. One is the grid with posterior expectation of betas as function of context-level covariates. The other is the posterior distribution of tau

**Usage**

```
plot_hdpglm(
  samples,
  X = NULL,
  W = NULL,
  ncol.taus = 1,
  ncol.betas = NULL,
  ncol.w = NULL,
  nrow.w = NULL,
  smooth.line = FALSE,
  pred.pexp.beta = FALSE,
  title.tau = NULL,
  true.tau = NULL,
  title.beta = NULL,
  tau.x.axis.size = 1.1,
  tau.y.axis.size = 1.1,
  tau.title.size = 1.2,
  tau.panel.title.size = 1.4,
  tau.legend.size = 1,
  beta.x.axis.size = 1.1,
  beta.y.axis.size = 1.1,
  beta.title.size = 1.2,
  beta.panel.title.size = 1.4,
  beta.legend.size = 1,
  tau.xlab = NULL
)
```

**Arguments**

samples	an output of the function <code>hdpGLM</code>
X	a string vector with the name of the first-level covariates whose associated tau should be displayed

W	a string vector with the name of the context-level covariate(s) whose linear effect will be displayed. If NULL, the linear effect tau of all context-level covariates are displayed. Note: the context-level covariate must have been included in the estimation of the model.
ncol.taus	integer with the number of columns of the grid containing the posterior distribution of tau
ncol.betas	integer with the number of columns of the posterior expectation of betas as function of context-level features
ncol.w	integer with the number of columns to use to display the different context-level covariates
nrow.w	integer with the number of rows to use to display the different context-level covariates
smooth.line	boolean, if TRUE the plot will display a regression line representing the regression of the posterior expectation of the linear coefficients betas on the context-level covariates. Default FALSE
pred.pexp.beta	boolean, if TRUE the plots will display a line with the predicted posterior expectation of betas obtained using the posterior expectation of taus, the linear coefficients of the expectation of beta
title.tau	string, the title for the posterior distribution of the context effects
true.tau	a data.frame with four columns. The first must be named w and it indicates the index of each context-level covariate, starting with 0 for the intercept term. The second column named beta must contain the indexes of the betas of individual-level covariates, starting with 0 for the intercept term. The third column named Parameter must be named tau<w><beta>, where w and beta must be the actual values displayed in the columns w and beta. Finally, it must have a column named True with the true value of the parameter.
title.beta	string, the title for the posterior expectation of beta as function of context-level covariate
tau.x.axis.size	numeric, relative size of the x-axis of the plot with tau
tau.y.axis.size	numeric, relative size of the y-axis of the plot with tau
tau.title.size	numeric, relative size of the title of the plot with tau
tau.panel.title.size	numeric, relative size of the title of the panels of the plot with tau
tau.legend.size	numeric, relative size of the legend of the plot with tau
beta.x.axis.size	numeric, relative size of the x-axis of the plot with beta
beta.y.axis.size	numeric, relative size of the y-axis of the plot with beta
beta.title.size	numeric, relative size of the title of the plot with beta
beta.panel.title.size	numeric, relative size of the title of the panels of the plot with beta

beta.legend.size            numeric, relative size of the legend of the plot with beta

tau.xlab                    string, the label of the x-axis for the plot with tau

### Examples

```
library(magrittr)
# Note: this example is just for illustration. MCMC iterations are very reduced
set.seed(10)
n = 20
data.context1 = tibble::tibble(x1 = rnorm(n, -3),
                               x2 = rnorm(n, 3),
                               z = sample(1:3, n, replace=TRUE),
                               y = I(z==1) * (3 + 4*x1 - x2 + rnorm(n)) +
                                   I(z==2) * (3 + 2*x1 + x2 + rnorm(n)) +
                                   I(z==3) * (3 - 4*x1 - x2 + rnorm(n)) ,
                               w = 20
                               )
data.context2 = tibble::tibble(x1 = rnorm(n, -3),
                               x2 = rnorm(n, 3),
                               z = sample(1:2, n, replace=TRUE),
                               y = I(z==1) * (1 + 3*x1 - 2*x2 + rnorm(n)) +
                                   I(z==2) * (1 - 2*x1 + x2 + rnorm(n)),
                               w = 10
                               )
data = data.context1 %>%
  dplyr::bind_rows(data.context2)

## estimation
mcmc = list(burn.in=1, n.iter=50)
samples = hdpGLM(y ~ x1 + x2, y ~ w, data=data, mcmc=mcmc, n.display=1)

plot_hdpglm(samples)
plot_hdpglm(samples, ncol.taus=2, ncol.betas=2, X='x1')
plot_hdpglm(samples, ncol.taus=2, ncol.betas=2, X='x1', ncol.w=2, nrow.w=1,
             pred.pexp.beta=TRUE, smooth.line=TRUE )
```

---

plot\_pexp\_beta

*Plot beta posterior expectation*

---

### Description

This function plots the posterior expectation of beta, the linear effect of the individual level covariates, as function of the context-level covariates

**Usage**

```
plot_pexp_beta(
  samples,
  X = NULL,
  W = NULL,
  pred.pexp.beta = FALSE,
  ncol.beta = NULL,
  ylab = NULL,
  nrow.w = NULL,
  ncol.w = NULL,
  smooth.line = FALSE,
  title = NULL,
  legend.position = "top",
  col.pred.line = "red",
  x.axis.size = 1.1,
  y.axis.size = 1.1,
  title.size = 12,
  panel.title.size = 1.4,
  legend.size = 1
)
```

**Arguments**

<code>samples</code>	an output of the function <a href="#">hdpGLM</a>
<code>X</code>	a string vector with the name of the first-level covariates whose associated tau should be displayed
<code>W</code>	a string vector with the name of the context-level covariate(s) whose linear effect will be displayed. If NULL, the linear effect tau of all context-level covariates are displayed. Note: the context-level covariate must have been included in the estimation of the model.
<code>pred.pexp.beta</code>	boolean, if TRUE the plots will display a line with the predicted posterior expectation of betas obtained using the posterior expectation of taus, the linear coefficients of the expectation of beta
<code>ncol.beta</code>	integer with number of columns of the grid used for each group of context-level covariates
<code>ylab</code>	string, the label of the y-axis
<code>nrow.w</code>	integer with the number of rows of the grid
<code>ncol.w</code>	integer with the number of columns of the grid
<code>smooth.line</code>	boolean, if TRUE the plot will display a regression line representing the regression of the posterior expectation of the linear coefficients betas on the context-level covariates. Default FALSE
<code>title</code>	string, title of the plot
<code>legend.position</code>	one of four options: "bottom" (default), "top", "left", or "right". It indicates the position of the legend

col.pred.line string with color of fitted line. Only works if pred.pexp.beta=TRUE  
 x.axis.size numeric, the relative size of the label in the x-axis  
 y.axis.size numeric, the relative size of the label in the y-axis  
 title.size numeric, absolute size of the title  
 panel.title.size numeric, the relative size of the titles in the panel of the plot  
 legend.size numeric, the relative size of the legend

### Examples

```
library(magrittr)
set.seed(66)

# Note: this example is just for illustration. MCMC iterations are very reduced
set.seed(10)
n = 20
data.context1 = tibble::tibble(x1 = rnorm(n, -3),
                              x2 = rnorm(n, 3),
                              z = sample(1:3, n, replace=TRUE),
                              y =I(z==1) * (3 + 4*x1 - x2 + rnorm(n)) +
                                I(z==2) * (3 + 2*x1 + x2 + rnorm(n)) +
                                I(z==3) * (3 - 4*x1 - x2 + rnorm(n)) ,
                              w = 20
                              )
data.context2 = tibble::tibble(x1 = rnorm(n, -3),
                              x2 = rnorm(n, 3),
                              z = sample(1:2, n, replace=TRUE),
                              y =I(z==1) * (1 + 3*x1 - 2*x2 + rnorm(n)) +
                                I(z==2) * (1 - 2*x1 + x2 + rnorm(n)),
                              w = 10
                              )
data = data.context1 %>%
  dplyr::bind_rows(data.context2)

## estimation
mcmc = list(burn.in=1, n.iter=50)
samples = hdpGLM(y ~ x1 + x2, y ~ w, data=data, mcmc=mcmc, n.display=1)

plot_pexp_beta(samples)
plot_pexp_beta(samples, X='x1', ncol.w=2, nrow.w=1)
plot_pexp_beta(samples, X='x1', ncol.beta=2)
plot_pexp_beta(samples, pred.pexp.beta=TRUE, W="w", X=c("x1", "x2"))
plot_pexp_beta(samples, W='w', smooth.line=TRUE, pred.pexp.beta=TRUE, ncol.beta=2)
```

**Description**

Function to plot posterior distribution of tau

**Usage**

```
plot_tau(
  samples,
  X = NULL,
  W = NULL,
  title = NULL,
  true.tau = NULL,
  show.all.taus = FALSE,
  show.all.betas = FALSE,
  ncol = NULL,
  legend.position = "top",
  x.axis.size = 1.1,
  y.axis.size = 1.1,
  title.size = 1.2,
  panel.title.size = 1.4,
  legend.size = 1,
  xlab = NULL
)
```

**Arguments**

samples	an output of the function <a href="#">hdpgLM</a>
X	a string vector with the name of the first-level covariates whose associated tau should be displayed
W	a string vector with the name of the context-level covariate(s) whose linear effect will be displayed. If NULL, the linear effect tau of all context-level covariates are displayed. Note: the context-level covariate must have been included in the estimation of the model.
title	string, title of the plot
true.tau	a data.frame with four columns. The first must be named w and it indicates the index of each context-level covariate, starting with 0 for the intercept term. The second column named beta must contain the indexes of the betas of individual-level covariates, starting with 0 for the intercept term. The third column named Parameter must be named tau<w><beta>, where w and beta must be the actual values displayed in the columns w and beta. Finally, it must have a column named True with the true value of the parameter.
show.all.taus	boolean, if FALSE (default) the posterior distribution of taus representing the intercept of the expectation of beta are omitted
show.all.betas	boolean, if FALSE (default) the taus affecting only the intercept terms of the outcome variable are omitted
ncol	number of columns of the grid. If NULL, one column is used

**legend.position** one of four options: "bottom" (default), "top", "left", or "right". It indicates the position of the legend  
**x.axis.size** numeric, the relative size of the label in the x-axis  
**y.axis.size** numeric, the relative size of the label in the y-axis  
**title.size** numeric, the relative size of the title of the plot  
**panel.title.size** numeric, the relative size of the titles in the panel of the plot  
**legend.size** numeric, the relative size of the legend  
**xlab** string, the label of the x-axis

### Examples

```

library(magrittr)
set.seed(66)

# Note: this example is just for illustration. MCMC iterations are very reduced
set.seed(10)
n = 20
data.context1 = tibble::tibble(x1 = rnorm(n, -3),
                               x2 = rnorm(n, 3),
                               z = sample(1:3, n, replace=TRUE),
                               y = I(z==1) * (3 + 4*x1 - x2 + rnorm(n)) +
                                   I(z==2) * (3 + 2*x1 + x2 + rnorm(n)) +
                                   I(z==3) * (3 - 4*x1 - x2 + rnorm(n)) ,
                               w = 20
                              )
data.context2 = tibble::tibble(x1 = rnorm(n, -3),
                               x2 = rnorm(n, 3),
                               z = sample(1:2, n, replace=TRUE),
                               y = I(z==1) * (1 + 3*x1 - 2*x2 + rnorm(n)) +
                                   I(z==2) * (1 - 2*x1 + x2 + rnorm(n)),
                               w = 10
                              )

data = data.context1 %>%
  dplyr::bind_rows(data.context2)

## estimation
mcmc = list(burn.in=1, n.iter=50)
samples = hdpGLM(y ~ x1 + x2, y ~ w, data=data, mcmc=mcmc, n.display=1)

plot_tau(samples)
plot_tau(samples, ncol=2)
plot_tau(samples, X='x1', W='w')
plot_tau(samples, show.all.taus=TRUE, show.all.betas=TRUE, ncol=2)

```

---

predict.dpGLM	<i>dpGLM Predicted values</i>
---------------	-------------------------------

---

**Description**

Function returns the predicted (fitted) values of the outcome variable using the estimated posterior expectation of the linear covariate betas produced by the hdpGLM function

**Usage**

```
## S3 method for class 'dpGLM'
predict(object, new_data = NULL, ...)
```

**Arguments**

object	outcome of the function hdpLGM
new_data	data frame with the values of the covariates that are going to be used to generate the predicted/fitted values. The posterior mean is used to create the predicted values
...	family : a string with the family of the output variable: gaussian (default), binomial, etc...

**Value**

It returns a data.frame with the fitted values for the outcome variable, which are produced using the estimated posterior expectation of the linear coefficients beta.

---

predict.hdpGLM	<i>hdpGLM Predicted values</i>
----------------	--------------------------------

---

**Description**

Function returns the predicted (fitted) values of the outcome variable using the estimated posterior expectation of the linear covariate betas produced by the hdpGLM function

**Usage**

```
## S3 method for class 'hdpGLM'
predict(object, new_data = NULL, ...)
```

**Arguments**

object	outcome of the function hdpLGM
new_data	data frame with the values of the covariates that are going to be used to generate the predicted/fitted values. The posterior mean is used to create the predicted values
...	family : a string with the family of the output variable: gaussian (default), binomial, etc...

**Value**

It returns a data.frame with the fitted values for the outcome variable, which are produced using the estimated posterior expectation of the linear coefficients beta.

---

print.dpGLM	<i>Print</i>
-------------	--------------

---

**Description**

Generic method to print the output of the dpGLM function

**Usage**

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

**Arguments**

x	a dpGLM object returned by the function hdpGLM
...	ignore

**Value**

returns a summary of the posterior distribution of the parameters

---

print.dpGLM\_data      *Print*

---

**Description**

Generic method to print the output of the hdpGLM\_simulateData function

**Usage**

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

**Arguments**

x                    a dpGLM\_data object returned by the function hdpGLM\_simulateData  
...                  ignore

**Value**

returns a summary of the simulated data

---

print.hdpGLM          *Print*

---

**Description**

Generic method to print the output of the hdpGLM function

**Usage**

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

**Arguments**

x                    a hdpGLM object returned by the function hdpGLM  
...                  ignore

**Value**

returns a summary of the posterior distribution of the parameters

---

```
print.hdpGLM_data      Print
```

---

**Description**

Generic method to print the output of the hdpGLM\_simulateData function

**Usage**

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

**Arguments**

```
x          a hdpGLM_data object returned by the function hdpGLM_simulateData
...        ignore
```

**Value**

returns a summary of the simulated data

---

```
summary.dpGLM          Summary for dpGLM class
```

---

**Description**

This function provides a summary of the MCMC samples from the dpGLM model

**Usage**

```
## S3 method for class 'dpGLM'
summary(object, ...)
```

**Arguments**

```
object      a dpGLM object returned by the function hdpGLM
...         The additional parameters accepted are:
           true.beta: (see plot.dpGLM)
```

**Details**

Data points are assigned to clusters according to the highest estimated probability of belonging to that cluster

---

summary.dpGLM_data	<i>Summary dpGLM data</i>
--------------------	---------------------------

---

**Description**

This function summarizes the data and parameters used to generate the data using the function `hdpLGM`.

**Usage**

```
## S3 method for class 'dpGLM_data'
summary(object, ...)
```

**Arguments**

<code>object</code>	an object of the class <code>dpGLM_data</code>
<code>...</code>	ignored

**Value**

The function returns a list with the summary of the data produced by the standard summary function and a `data.frame` with the true values of beta for each cluster.

---

summary.hdpGLM	<i>Summary for hdpGLM class</i>
----------------	---------------------------------

---

**Description**

This is a generic summary function that describes the output of the function [hdpGLM](#)

**Usage**

```
## S3 method for class 'hdpGLM'
summary(object, ...)
```

**Arguments**

<code>object</code>	an object of the class <code>hdpGLM</code> generated by the function <a href="#">hdpGLM</a>
<code>...</code>	Additional arguments accepted are: <code>true.beta</code> : a <code>data.frame</code> with the true values of the linear coefficients beta if they are known. The <code>data.frame</code> must contain a column named <code>j</code> with the index of the context associated with that particular linear coefficient beta. It must match the indexes used in the data set for each context. Another column named <code>k</code> must be provided, indicating the cluster of beta, and a column named <code>Parameter</code> with the name of the linear coefficients ( <code>beta1</code> , <code>beta2</code> , ..., <code>beta_dx</code> ,

where `dx` is the number of covariates at the individual level, and `beta1` is the coefficient of the intercept term). It must contain a column named `True` with the true value of the betas. Finally, the `data.frame` must contain columns with the context-level covariates as used in the estimation of the `hdpGLM` function (see Details below).

`true.tau`: a `data.frame` with four columns. The first must be named `w` and it indicates the index of each context-level covariate, starting with 0 for the intercept term. The second column named `beta` must contain the indexes of the betas of individual-level covariates, starting with 0 for the intercept term. The third column named `Parameter` must be named `tau<w><beta>`, where `w` and `beta` must be the actual values displayed in the columns `w` and `beta`. Finally, it must have a column named `True` with the true value of the parameter.

### Details

The function `hdpGLM` returns a list with the samples from the posterior distribution along with other elements. That list contains an element named `context.cov` that connects the indexed "C" created during the estimation and the context-level covariates. So each unique context-level covariate gets an index during the estimation. The algorithm only requires the context-level covariates, but it creates such index `C` to help the estimation. If `true.beta` is provided, it must contain indexes for the context as well, which indicates the context of each specific linear coefficient `beta`. Such index will probably be different from the one created by the algorithm. Therefore, when the `true.beta` is provided, we need to connect the context index `C` generated by the algorithm and the column `j` in the `true.beta` `data.frame` in order to compare the true values and the estimated value for each context. That is why we need the values of the context-level covariates as well. The summary uses them as key to merge the true and the estimated values for each context. The true and estimated clusters are matched based on the shortest distance between the estimated posterior average and the true value in each context because the labels of the clusters in the estimation can vary, even though the same data points are classified in the same clusters.

### Value

The function returns a list with two `data.frames`. The first summarizes the posterior distribution of the linear coefficients `beta`. The mean, median, and the 95% HPD interval are provided. The second `data.frame` contains the summary of the posterior distribution of the parameter `tau`.

---

summary.hdpGLM\_data    *Summary*

---

### Description

This function summarizes the data simulated by the function `hdpGLM_simulateData`

### Usage

```
## S3 method for class 'hdpGLM_data'
summary(object, ...)
```

**Arguments**

object            an object of the class hdpGLM\_data, which is produced by the function hdpGLM\_simulateData  
 ...                ignored

**Value**

It returns a list with three elements. The first is a summary of the data, the second a tibble with the linear coefficients beta and their values used to generate the data, and the third element is also a tibble with the true values of tau used to generate the betas.

---

summary_tidy	<i>Tidy summary</i>
--------------	---------------------

---

**Description**

This function provides a summary of the MCMC samples from the dpGLM model

**Usage**

```
summary_tidy(object, ...)
```

**Arguments**

object            a dpGLM object returned by the function hdpGLM  
 ...                The additional parameters accepted are:  
                   true.beta: (see [plot.dpGLM](#))

**Details**

Data points are assigned to clusters according to the highest estimated probability of belonging to that cluster

---

welfare	<i>Fake data set with 2000 observations</i>
---------	---

---

**Description**

A dataset containing simulated data about public opinion

**Usage**

```
welfare
```

**Format**

A data frame with 2000 rows and 4 variables:

**support** support for welfare policies

**inequality** levels of inequality in the neighborhood

**income** individual-level income

**ideology** individual-level ideology

**Source**

Simulated data

---

welfare2

*Fake data set with 2000 observations*

---

**Description**

A dataset containing simulated data about public opinion in different countries

**Usage**

welfare2

**Format**

A data frame with 2000 rows and 6 variables:

**support** support for welfare policies

**inequality** levels of inequality in the neighborhood

**income** individual-level income

**ideology** individual-level ideology

**country** country label or index

**gap** country-level gender gap in country's provision of public good

**Source**

Simulated data

# Index

## \* datasets

welfare, [31](#)  
welfare2, [32](#)

classify, [2](#)

coef.dpGLM, [3](#)

coef.hdpGLM, [3](#)

hdpGLM, [2](#), [4](#), [7](#), [12](#), [14–18](#), [21](#), [23](#), [29](#), [30](#)

hdpGLM-package (hdpGLM), [4](#)

hdpGLM\_classify, [7](#)

hdpGLM\_simParameters, [8](#)

hdpGLM\_simulateData, [9](#)

lm, [4](#)

mcmc\_info.dpGLM, [11](#)

mcmc\_info.hdpGLM, [11](#)

nclusters, [11](#)

plot.dpGLM, [12](#), [28](#), [31](#)

plot.hdpGLM, [14](#)

plot\_beta, [16](#)

plot\_beta\_sim, [17](#)

plot\_hdpglm, [18](#)

plot\_pexp\_beta, [20](#)

plot\_tau, [22](#)

predict.dpGLM, [25](#)

predict.hdpGLM, [25](#)

print.dpGLM, [26](#)

print.dpGLM\_data, [27](#)

print.hdpGLM, [27](#)

print.hdpGLM\_data, [28](#)

summary.dpGLM, [28](#)

summary.dpGLM\_data, [29](#)

summary.hdpGLM, [29](#)

summary.hdpGLM\_data, [30](#)

summary\_tidy, [31](#)

welfare, [31](#)

welfare2, [32](#)