Package ‘bpnreg’

October 12, 2022

Type Package

Title Bayesian Projected Normal Regression Models for Circular Data

Version 2.0.2

Description Fitting Bayesian multiple and mixed-effect regression models for circular data based on the projected normal distribution. Both continuous and categorical predictors can be included. Sampling from the posterior is performed via an MCMC algorithm. Posterior descriptives of all parameters, model fit statistics and Bayes factors for hypothesis tests for inequality constrained hypotheses are provided. See Cremers, Mulder & Klugkist (2018) <doi:10.1111/bmsp.12108> and Nuñez-Antonio & Guttiérez-Peña (2014) <doi:10.1016/j.csda.2012.07.025>.

Depends R (>= 3.6.0)

License GPL-3

URL https://github.com/joliencremers/bpnreg

BugReports https://github.com/joliencremers/bpnreg/issues

Encoding UTF-8

LazyData true

LinkingTo Rcpp (>= 1.0.2), RcppArmadillo (>= 0.10.1.2.0), BH (>= 1.69.0.1)

Imports Rcpp (>= 1.0.2), haven (>= 2.1.1), methods (>= 3.6.0)

RoxygenNote 7.1.1

Suggests qpdf, knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation yes

Author Jolien Cremers [aut, cre]

Maintainer Jolien Cremers <joliencremers@gmail.com>

Repository CRAN

Date/Publication 2021-08-06 08:20:06 UTC

R topics documented:

`BFc` *Bayes Factors*

Description

BF gives Bayes factors for inequality constrained hypotheses on circular mean differences.

Usage

```
BFc(object, hypothesis, type = "anchor")
```

Arguments

`object`	an object used to select a method.
`hypothesis`	the inequality constrained hypothesis to test.
`type`	type of hypothesis to test `c("anchor", "isotropic")`. As of yet only anchor hypotheses can be tested.

Details

the methods BFc.bpnr and BFc.bpnme have their own help page.

Examples

```
library(bpnreg)
fit.Motor <- bpnr(pred.I = Phaserad ~ 1 + Cond, data = Motor,
its = 100, burn = 10, n.lag = 3)
BFc(fit.Motor, hypothesis = "Condsemi.imp < Condimp")
```

`BFc.bpnme` *Bayes Factors for a Bayesian circular mixed-effects model*

Description

Outputs Bayes Factors for inequality constrained hypotheses on the circular differences between several levels of a categorical variable and the baseline.

Usage

```
## S3 method for class 'bpnme'
BFc(object, hypothesis, type = "anchor")
```

Arguments

object	a bpnme object obtained from the function bpnme.
hypothesis	the inequality constrained hypothesis to test.
type	type of hypothesis to test c("anchor", "isotropic"). As of yet only anchor hypotheses can be tested.

Value

Bayes Factors for inequality constrained hypotheses on mean differences.

Examples

```
library(bpnreg)
fit.Maps <- bpnme(pred.I = Error.rad ~ Maze + Trial.type + L.c + (1|Subject),
data = Maps, its = 100, burn = 1, n.lag = 1)
BFc(fit.Maps, hypothesis = "Maze1 < Trial.type1")
```

BFc.bpnr — *Bayes Factors for a Bayesian circular regression model*

Description

Outputs Bayes Factors for the circular differences between several levels of a categorical variable and the baseline.

Usage

```
## S3 method for class 'bpnr'
BFc(object, hypothesis, type = "anchor")
```

Arguments

object	a bpnr object obtained from the function bpnr().
hypothesis	the inequality constrained hypothesis to test.
type	type of hypothesis to test c("anchor", "isotropic"). As of yet only anchor hypotheses can be tested.

Value

Bayes Factors for inequality constrained hypotheses on mean differences.

Examples

```
library(bpnreg)
fit.Motor <- bpnr(pred.I = Phaserad ~ 1 + Cond, data = Motor,
its = 100, burn = 10, n.lag = 3)
BFc(fit.Motor, hypothesis = "Condsemi.imp < Condimp")
```

bpnme *Fit a Bayesian circular mixed-effects model*

Description

This function fits a Bayesian circular mixed-effects model based on the projected normal distribution.

Usage

```
bpnme(
  pred.I,
  data,
  pred.II = pred.I,
  its = 1000,
  burn = 1,
  n.lag = 1,
  seed = NULL
)
```

Arguments

`pred.I`	model equation for effects of component 1.
`data`	the dataframe used for analysis.
`pred.II`	model equation for effects of component 2.
`its`	output iterations of the MCMC sampler.
`burn`	number of burn-in iterations.
`n.lag`	amount of lag for the iterations and burn-in.
`seed`	user-specified random seed.

Details

Because the model is based on the projected normal distribution, a model equation has to be given for the fixed and random effects of the two components. By default the model equation of the second component `pred.II` is set to be equal to that of the first component. Note that the circular outcome needs to be measured in radians on a scale from 0 to 2π. For more information about the projected normal distribution see Presnell, Morrisson & Littell (1998). The model can handle at most one grouping factor.

A tutorial on how to use this function can be found in Cremers & Klugkist (2018). More details on the sampling algorithm and interpretation of the coefficients from the model can be found in Nuñez-Antonio & Guttiérrez-Peña (2014) and Cremers, Pennings, Mainhard & Klugkist (2019). The uninformative priors for the fixed effect regression coefficients of the two components are set to N(0, 10000). Note that the model is only developed for models with a single nesting variable.

Value

A bpnme object, which can be further analyzed using the associated functions traceplot.bpnme, BFc.bpnme, coef_lin.bpnme, coef_circ.bpnme, coef_ran.bpnme, fit.bpnme and print.bpnme.

A bpnr object contains the following elements (some elements are not returned if not applicable)

beta1 A matrix of posterior samples for the fixed effects coefficients for the first component.

beta2 A matrix of posterior samples for the fixed effects coefficients for the second component.

b1 An array of posterior samples for the random effects coefficients for the first component.

b2 An array of posterior samples for the random effects coefficients for the second component.

omega1 An array of posterior samples for the random effect variances of the first component.

omega2 An array of posterior samples for the random effect variances of the second component.

predictiva A list containing the posterior density values for all timepoints of individuals in the dataset for all iterations. The rowsums of this matrix are the likelihood values for all iterations

circular.ri A vector of posterior samples for the circular random intercepts.

N Number of observed cases.

its Number of output iterations.

n.lag One in n.lag iterations will be saved as output iteration. Set lag to 1 to save all iterations (default).

burn Burn-in time for the MCMC sampler.

p1 Number of fixed effect parameters predicting the first component.

p2 Number of fixed effect parameters predicting the second component.

q1 Number of random effect parameters predicting the first component.

q2 Number of random effect parameters predicting the second component.

a.x A matrix of posterior samples for a.x which describes the location of the inflection point of the regression curve on the axis of the predictor.

a.c A matrix of posterior samples for a.c which describes the location of the inflection point of the regression curve on the axis of the circular outcome.

b.c A matrix of posterior samples for b.c which describes the slope of the tangent line at the inflection point.

SAM A matrix of posterior samples for the circular regression slopes at the mean.

AS A matrix of posterior samples for the average slopes of the circular regression.

SSDO A matrix of posterior samples for the signed shortest distance to the origin.

circ.diff A matrix of posterior samples for the circular difference found between levels of categorical variables and the intercept.

cRSnum A vector of posterior samples of the circular random slope estimates for the continuous variables

cRScat A vector of posterior samples of the circular random slope estimates for the categorical variables

cRS A vector of posterior samples of the circular random slope estimates `cRI` A vector of posterior samples of the mean resultant length of the circular random intercept, a measure of concentration.

`Call` The matched call.

`lin.coef.I` The mean, mode, standard deviation and 95 posterior density of the linear fixed effect coefficients for `beta1`.

`lin.coef.II` The mean, mode, standard deviation and 95 confidence interval of the highest posterior density of the linear fixed effect coefficients for `beta2`.

`circ.coef` The mean, mode, standard deviation and 95 density for `a.x`, `a.c`, `SSDO`, and the circular fixed effect coefficients `b.c`, `AS`, and `SAM`

`circ.coef.cat` The mean, mode, standard deviation and 95 confidence interval of the highest posterior density the circular difference between levels of categorical variables and the intercept.

`circ.coef.means` The mean, mode, standard deviation and 95 confidence interval of the highest posterior density of circular means of the categorical variables.

`model.fit` A list of information criteria for assessment of model fit.

`lin.res.varrand.I` The mean, mode, standard deviation and 95 variances of the random intercepts and slopes of component I.

`lin.res.varrand.II` The mean, mode, standard deviation and 95
of component II.

`circ.res.varrand` The mean, mode, standard deviation and 95 random intercepts and slopes.

`mm` A list of information, model matrices, sample size, etc. on the specified model.

Source

Cremers, J., Mainhard, M.T. & Klugkist, I. (2018). Assessing a Bayesian Embedding Approach to Circular Regression Models. Methodology, 14, 69-81.

Cremers, J. & Klugkist, I. (2018). One direction? A tutorial for circular data with examples in cognitive psychology. Frontiers in Psychology: Cognitive Science.

Cremers, J., Pennings, H.J.M., Mainhard, M.T. & Klugkist, I. (2019). Circular Modelling of Circumplex Measurements for Interpersonal Behavior. Assessment, Online First.

Nuñez-Antonio, G. & Gutiérrez-Peña, E. (2014). A Bayesian model for longitudinal circular data based on the projected normal distribution. Computational Statistics and Data Analysis, 71, 506-519.

Presnell, B., Morrison, S.P. & Littell, R.C. (1998). Projected multivariate linear models for directional data. Journal of the American Statistical Association, 93 (443), 1068 - 1077.

Examples

```
library(bpnreg)
bpnme(Error.rad ~ Maze + Trial.type + (1|Subject), Maps, its = 100)
```

bpnr *Fit a Bayesian circular regression model*

Description

This function fits a Bayesian circular regression model based on the projected normal distribution.

Usage

```
bpnr(
  pred.I,
  data,
  pred.II = pred.I,
  its = 1000,
  burn = 1,
  n.lag = 1,
  seed = NULL
)
```

Arguments

pred.I	model equation for effects of component 1.
data	the dataframe used for analysis.
pred.II	model equation for effects of component 2.
its	output iterations of the MCMC sampler.
burn	number of burn-in iterations.
n.lag	amount of lag for the iterations and burn-in.
seed	user-specified random seed.

Details

Because the model is based on the projected normal distribution, a model equation has to be given for two components. By default the equation of the second component `pred.II` is set to be equal to that of the first component. Note that the circular outcome needs to be measured in radians on a scale from 0 to 2π. For more information about the projected normal distribution see Presnell, Morrisson & Littell (1998).

A tutorial on how to use this function can be found in Cremers & Klugkist (2018). More details on the sampling algorithm and interpretation of the coefficients from the model can be found in Cremers, Mulder & Klugkist (2018) and Cremers, Mainhard & Klugkist (2018). The uninformative priors for the regression coefficients of the two components are set to N(0, 10000).

Value

A bpnr object, which can be further analyzed using the associated functions traceplot.bpnr, BFc.bpnr, coef_lin.bpnr, coef_circ.bpnr, fit.bpnr and print.bpnr.

A bpnr object contains the following elements (some elements are not returned if not applicable)

- `beta1` A matrix of posterior samples for the coefficients `beta1` of the first component.
- `beta2` A matrix of posterior samples for the coefficients `beta2` for the second component.
- `Likelihood` A matrix containing the posterior density values for all individuals in the dataset for all iterations. The rowsums of this matrix are the likelihood values for all iterations
- `its` Number of output iterations.
- `n.lag` One in `n.lag` iterations will be saved as output iteration. Set lag to 1 to save all iterations (default).
- `burn-in` Burn-in time for the MCMC sampler.
- `p1` Number of parameters predicting the first component.
- `p2` Number of parameters predicting the second component.
- `theta` The circular outcome vector measured in radians.
- `a.x` A matrix of posterior samples for `a.x` which describes the location of the inflection point of the regression curve on the axis of the predictor.
- `a.c` A matrix of posterior samples for `a.c` which describes the location of the inflection point of the regression curve on the axis of the circular outcome.
- `b.c` A matrix of posterior samples for `b.c` which describes the slope of the tangent line at the inflection point.
- `SAM` A matrix of posterior samples for the circular regression slopes at the mean.
- `AS` A matrix of posterior samples for the average slopes of the circular regression.
- `SSDO` A matrix of posterior samples for the signed shortest distance to the origin.
- `circ.diff` A matrix of posterior samples for the circular difference between levels of categorical variables and the intercept.
- `Call` The matched call.
- `lin.coef.I` The mean, mode, standard deviation and 95 confidence interval of the highest posterior density of the linear coefficients for `beta1`.
- `lin.coef.II` The mean, mode, standard deviation and 95 density of the linear coefficients for `beta2`.
- `circ.coef` The mean, mode, standard deviation and 95 confidence interval of the highest posterior density for the `a.x`, `a.c`, `b.c`, `AS`, `SAM` and `SSDO` of the circular coefficients.
- `circ.coef.cat` The mean, mode, standard deviation and 95 density the circular difference between levels of categorical variables and the intercept.
- `circ.coef.means` The mean, mode, standard deviation and 95 circular means of the categorical variables.
- `model.fit` A list of information criteria for assessment of model fit.
- `mm` A list of information, model matrices, sample size, etc. on the specified model.

Source

Cremers, J., Mulder, K.T. & Klugkist, I. (2018). Circular interpretation of regression coefficients. British Journal of Mathematical and Statistical Psychology, 71(1), 75-95.

Cremers, J., Mainhard, M.T. & Klugkist, I. (2018). Assessing a Bayesian Embedding Approach to Circular Regression Models. Methodology, 14, 69-81.

Cremers, J. & Klugkist, I. (2018). One direction? A tutorial for circular data with examples in cognitive psychology. Frontiers in Psychology: Cognitive Science.

Presnell, B., Morrison, S.P. & Littell, R.C. (1998). Projected multivariate linear models for directional data. Journal of the American Statistical Association, 93 (443), 1068 - 1077.

Examples

```
library(bpnreg)
bpnr(Phaserad ~ Cond + AvAmp, Motor)
```

`bpnreg` *bpnreg: A package to analyze Bayesian projected normal circular regression models*

Description

This package contains functions to analyze circular regression type models (multivariate and mixed-effects). It is based on the 'embedding' approach to circular modelling and makes use of the projected normal distribution. Its estimation method is a Bayesian MCMC sampler. Further technical details can be found in Cremers, Mulder & Klugkist (2018) and Cremers & Klugkist (2018).

Details

A tutorial on how to use this package can be found in Cremers & Klugkist (2018). More details on the sampling algorithm and interpretation of the coefficients from the model can be found in Cremers, Mulder & Klugkist (2018), Nuñez-Antonio & Gutiérrez-Peña, Cremers, Mainhard & Klugkist (2018) and Cremers, Pennings, Mainhard & Klugkist (2019).

Functions

The main functions of the package are:

`bpnr`, which runs an MCMC sampler in `C++` and returns an S3 object of type `bpnr`, which can be further analyzed through associated functions.

`bpnme`, which runs an MCMC sampler in `R` and returns an S3 object of type `bpnme`, which can be further analyzed through associated functions.

Datasets

Datasets included in this package are:

Motor, A dataset from a study by Puglisi et.al. (2017) on the role of attention in human motor resonance.

Maps, A dataset from a study by Warren et.al. (2017) on the geometry of human knowledge of navigation space.

Source

Cremers, J., Pennings, H.J.M., Mainhard, M.T. & Klugkist, I. (2019). Circular Modelling of Circumplex Measurements for Interpersonal Behavior. Assessment.

Cremers, J., Mulder, K.T. & Klugkist, I. (2018). Circular interpretation of regression coefficients. British Journal of Mathematical and Statistical Psychology, 71(1), 75-95.

Cremers, J., Mainhard, M.T. & Klugkist, I. (2018). Assessing a Bayesian Embedding Approach to Circular Regression Models. Methodology, 14, 69-81.

Cremers, J. & Klugkist, I. (2018). One direction? A tutorial for circular data with examples in cognitive psychology. Frontiers in Psychology: Cognitive Science.

Nuñez-Antonio, G. & Gutiérrez-Peña, E. (2014). A Bayesian model for longitudinal circular data based on the projected normal distribution. Computational Statistics and Data Analysis, 71, 506-519.

---

circ_coef — *Compute circular coefficients from linear coefficients*

Description

circ_coef computes the coordinates of the inflection point of a circular effect, the slope at the inflection point and the unsigned and signed shortest distance to the origin.

Usage

```
circ_coef(a1, a2, b1, b2)
```

Arguments

a1	intercept of the first linear component.
a2	intercept of the second linear component.
b1	slope of the first linear component.
b2	slope of the second linear component.

Value

A dataframe containing the coordinates of the inflection point of a circular effect, the slope at the inflection point and the unsigned and signed shortest distance to the origin.

circ_coef_rcpp *Compute circular coefficients*

Description

Compute circular coefficients

Usage

```
circ_coef_rcpp(a1, a2, b1, b2)
```

Arguments

a1	intercept estimate of component I.
a2	intercept estimate of component I.
b1	slope estimate of component I.
b2	slope estimate of component I.

coef_circ *Circular coefficients*

Description

`coef_circ` gives posterior summaries of the circular coefficients.

Usage

```
coef_circ(object, type = "continuous", units = "radians")
```

Arguments

object	an object used to select a method.
type	one of `c("continuous", "categorical")` to get either the coefficients for the continuous or categorical predictor variables.
units	one of `c("degrees", "radians")` to get categorical coefficients estimates and estimates for `$ac$`, `$bc$`, `AS` and `SAM` in degrees or radians.

Details

the methods coef_circ.bpnr and coef_circ.bpnme have their own help page.

Examples

```
library(bpnreg)
fit.Motor <- bpnr(pred.I = Phaserad ~ 1 + Cond, data = Motor,
its = 100, burn = 10, n.lag = 3)
coef_circ(fit.Motor)
coef_circ(fit.Motor, type = "categorical")
```

---

coef_circ.bpnme — *Obtain the circular coefficients of a Bayesian circular mixed-effects model*

Description

Gives the coefficients tables of the circular coefficients for a Bayesian circular mixed-effects model.

Usage

```
## S3 method for class 'bpnme'
coef_circ(object, type = "continuous", units = "radians")
```

Arguments

`object`	a `bpnme` `object` obtained from the function `bpnme`
`type`	one of `c("continuous", "categorical")` to get either the coefficients for the continuous or categorical predictor variables
`units`	one of `c("degrees", "radians")` to get categorical coefficients estimates and estimates for `$ac$`, `$bc$`, `AS` and `SAM` in degrees or radians.

Value

A matrix or list with posterior summaries of the circular coefficients in a Bayesian circular mixed-effects model.

Examples

```
library(bpnreg)
fit.Maps <- bpnme(pred.I = Error.rad ~ Maze + Trial.type + L.c + (1|Subject),
data = Maps,
its = 100, burn = 1, n.lag = 1)
coef_circ(fit.Maps)
```

coef_circ.bpnr — *Obtain the circular coefficients of a Bayesian circular regression model*

Description

Gives the coefficients tables of the circular coefficients for a Bayesian circular regression model.

Usage

```
## S3 method for class 'bpnr'
coef_circ(object, type = "continuous", units = "radians")
```

Arguments

`object`	a `bpnr object` obtained from the function `bpnr`
`type`	one of `c("continuous", "categorical")` to get either the coefficients for the continuous or categorical predictor variables
`units`	one of `c("degrees", "radians")` to get categorical coefficients estimates and estimates for `$ac$`, `$bc$`, `AS` and `SAM` in degrees or radians.

Value

A matrix or list with posterior summaries of the circular coefficients in a Bayesian circular regression model.

Examples

```
library(bpnreg)
fit.Motor <- bpnr(pred.I = Phaserad ~ 1 + Cond, data = Motor,
its = 100, burn = 10, n.lag = 3)
coef_circ(fit.Motor)
coef_circ(fit.Motor, type = "categorical")
```

coef_lin — *Linear coefficients*

Description

`coef_lin` gives posterior summaries of the linear coefficients.

Usage

```
coef_lin(object)
```

Arguments

`object`	an object used to select a method.

Details

the methods coef_lin.bpnr and coef_lin.bpnme have their own help page.

Examples

```
library(bpnreg)
fit.Motor <- bpnr(pred.I = Phaserad ~ 1 + Cond, data = Motor,
its = 100, burn = 10, n.lag = 3)
coef_lin(fit.Motor)
```

`coef_lin.bpnme` *Obtain the linear coefficients of a Bayesian circular mixed-effects model*

Description

Gives the coefficients tables of the linear coefficients for a Bayesian circular mixed-effects model.

Usage

```
## S3 method for class 'bpnme'
coef_lin(object)
```

Arguments

`object`	a `bpnme` `object` obtained from the function `bpnme`.

Value

A matrix with posterior summaries of the linear coefficients in a Bayesian circular mixed-effects model.

Examples

```
library(bpnreg)
fit.Maps <- bpnme(pred.I = Error.rad ~ Maze + Trial.type + L.c + (1|Subject),
data = Maps,
its = 100, burn = 1, n.lag = 1)
coef_lin(fit.Maps)
```

coef_lin.bpnr *Obtain the linear coefficients of a Bayesian circular regression model*

Description

Gives the coefficients tables of the linear coefficients for a circular regression model.

Usage

```
## S3 method for class 'bpnr'
coef_lin(object)
```

Arguments

`object`	a `bpnr object` obtained from the function `bpnr`.

Value

A matrix with posterior summaries of the linear coefficients in a Bayesian circular regression model.

Examples

```
library(bpnreg)
fit.Motor <- bpnr(pred.I = Phaserad ~ 1 + Cond, data = Motor,
its = 100, burn = 10, n.lag = 3)
coef_lin(fit.Motor)
```

coef_ran *Random effect variances*

Description

`coef_ran` gives posterior summaries of the circular or linear random effect variances.

Usage

```
coef_ran(object, type = "linear")
```

Arguments

`object`	an object used to select a method.
`type`	one of `c("linear", "circular")` to get either the linear or circular random effect variances.

Details

the method coef_ran.bpnme has its own help page.

Examples

```
library(bpnreg)
fit.Maps <- bpnme(pred.I = Error.rad ~ Maze + Trial.type + L.c + (1|Subject),
data = Maps,
its = 100, burn = 1, n.lag = 1)
coef_ran(fit.Maps)
coef_ran(fit.Maps, type = "circular")
```

`coef_ran.bpnme` *Obtain random effect variances of a Bayesian circular mixed-effects model*

Description

Gives posterior summaries of the circular or linear random effect variances. The circular random intercept variance and circular random slope variance of categorical predictors is computed as 1 - mean resultant length.

Usage

```
## S3 method for class 'bpnme'
coef_ran(object, type = "linear")
```

Arguments

object	a `bpnme` `object` obtained from the function `bpnme`.
type	one of `c("linear", "circular")` to get either the linear or circular random effect variances.

Value

A matrix with posterior summaries of the random effect variances.

Examples

```
library(bpnreg)
fit.Maps <- bpnme(pred.I = Error.rad ~ Maze + Trial.type + L.c + (1|Subject),
data = Maps,
its = 100, burn = 1, n.lag = 1)
coef_ran(fit.Maps)
coef_ran(fit.Maps, type = "circular")
```

DIC_reg *Compute Model Fit Measures Regression Model*

Description

Compute Model Fit Measures Regression Model

Usage

```
DIC_reg(theta, beta1, beta2, Likelihood, X1, X2)
```

Arguments

theta	circular outcome values
beta1	regression coefficients for the second component for each mcmc iteration from pnr function
beta2	regression coefficients for the second component for each mcmc iteration from pnr function
Likelihood	likelihood values for each individual and mcmc itertion from pnr function
X1	model matrix for the first component
X2	model matrix for the second component

eigen_val *Compute Eigenvalues*

Description

Compute Eigenvalues

Usage

```
eigen_val(X)
```

Arguments

X	A matrix.

eigen_vec — *Compute Eigenvectors*

Description

Compute Eigenvectors

Usage

```
eigen_vec(X)
```

Arguments

X	A matrix.

fit — *Model fit*

Description

`fit` gives several model fit statistics.

Usage

```
fit(object)
```

Arguments

object	an object used to select a method.

Details

the methods fit.bpnr and fit.bpnme have their own help page.

Examples

```
library(bpnreg)
fit.Motor <- bpnr(pred.I = Phaserad ~ 1 + Cond, data = Motor,
its = 100, burn = 10, n.lag = 3)
fit(fit.Motor)
```

fit.bpnme *Model fit for a Bayesian circular mixed-effects model*

Description

Outputs several model fit statistics for the Bayesian circular mixed-effects model

Usage

```
## S3 method for class 'bpnme'
fit(object)
```

Arguments

`object`	a `bpnme` `object` obtained from the function `bpnme`.

Value

a matrix containing the computed log pointwise predictive density (lppd), Deviance Information Criterion (DIC), an alternative version of the DIC (DIC_alt), and the Watanabe-Akaike Information Criterion computed in two different ways (WAIC1, WAIC2). The matrix also contains the number of parameters or 'effective number' of parameters that the several statistics are based on. Computation of the criteria is done according to Gelman et.al (2014) in *Bayesian Data Analysis*.

Examples

```
library(bpnreg)
fit.Maps <- bpnme(pred.I = Error.rad ~ Maze + Trial.type + L.c + (1|Subject),
data = Maps,
its = 100, burn = 1, n.lag = 1)
fit(fit.Maps)
```

fit.bpnr *Model fit for a Bayesian circular regression model*

Description

Outputs several model fit statistics for the Bayesian circular regression model

Usage

```
## S3 method for class 'bpnr'
fit(object)
```

Arguments

`object`	a `bpnr object` obtained from the function `bpnr()`.

Value

a matrix containing the computed log pointwise predictive density (lppd), Deviance Information Criterion (DIC), an alternative version of the DIC (DIC_alt), and the Watanabe-Akaike Information Criterion computed in two different ways (WAIC1, WAIC2). The matrix also contains the number of parameters or 'effective number' of parameters that the several statistics are based on. Computation of the criteria is done according to Gelman et.al (2014) in *Bayesian Data Analysis*.

Examples

```
library(bpnreg)
fit.Motor <- bpnr(pred.I = Phaserad ~ 1 + Cond, data = Motor,
its = 100, burn = 10, n.lag = 3)
fit(fit.Motor)
```

---

hmode *Estimate the mode by finding the highest posterior density interval*

Description

Estimate the mode by finding the highest posterior density interval

Usage

```
hmode(x, cip)
```

Arguments

`x`	a sample from which to estimate the interval
`cip`	bandwidth for the algorithm, ranging from 0 to 1

Value

a scalar containing the estimate of the mode

hmodeC *Estimate the mode by finding the highest posterior density interval*

Description

Estimate the mode by finding the highest posterior density interval

Usage

```
hmodeC(x, cip)
```

Arguments

x	a sample from which to estimate the interval
cip	bandwidth for the algorithm, ranging from 0 to 1

Value

a scalar containing the estimate of the mode

hmodeci *Find the highest density interval.*

Description

Find the highest density interval.

Usage

```
hmodeci(x, cip)
```

Arguments

x	a sample from which to estimate the interval
cip	bandwidth for the algorithm, ranging from 0 to 1

Value

a vector of length 2 containing the lower and upper bound of the interval.

hmodeciC — *Find the highest density interval of a circular variable*

Description

Find the highest density interval of a circular variable

Usage

```
hmodeciC(x, cip)
```

Arguments

x	a sample from which to estimate the interval
cip	bandwidth for the algorithm, ranging from 0 to 1

Value

a vector of length 2 containing the lower and upper bound of the interval

hpd_est — *Compute the 95 percent HPD of a vector of linear data*

Description

Compute the 95 percent HPD of a vector of linear data

Usage

```
hpd_est(x)
```

Arguments

x	a vector of linear data

Examples

```
library(bpnreg)
hpd_est(Motor$AvAmp)
```

hpd_est_circ *Compute the 95 percent HPD of a vector of circular data*

Description

Compute the 95 percent HPD of a vector of circular data

Usage

```
hpd_est_circ(x)
```

Arguments

x	a vector of circular data

Examples

```
library(bpnreg)
hpd_est_circ(subset(Motor, Cond = "exp")$Phaserad)
```

lik_reg *Compute the Likelihood of the PN distribution (regression)*

Description

Compute the Likelihood of the PN distribution (regression)

Usage

```
lik_reg(X1, X2, theta, beta1, beta2, n)
```

Arguments

X1	the model matrix of the first component
X2	the model matrix of the second component
theta	a circular outcome value
beta1	estimated linear coefficients of the first component
beta2	estimated linear coefficients of the second component
n	sample size

Maps *The geometry of human knowledge of navigation space.*

Description

A dataset from a study by Warren et.al. (2017) on the geometry of human knowledge of navigation space.

Usage

```
Maps
```

Format

A data frame with 160 rows and 8 variables:

Subject a numeric variable indicating the participant number

Trial.no a numeric variable indicating the trial number of the target a participant had to locate (1-8)

Maze a between subjects factor variable indicating the type of maze a participant was in (0 = Euclidean, 1 = non-Euclidean)

Trial.type a within subjects factor indicating the type of target (0 = standard, 1 = probe)

Error a numeric variable containing the angular error in degrees

Learn a numeric variable indicating the number of trials a participant completed in the training phase

L.c mean centered Learning

Error.rad a numeric variable containing the angular error in radians

Details

In their study Warren et.al. (2017) conduct an experiment in which a total of 20 participants used virtual reality headsets to navigate through one of two versions of a virtual maze. One version of the maze is the standard or Euclidean maze. The other version of the maze, the non-Euclidean maze, has exactly the same layout as the standard maze apart from that it contains wormholes by which participants can be 'teleported' from one place in the maze to another.

In a training phase participants had learned to navigate between different pairs of start and target objects in one of two versions of the maze. The number of trials each participants completed in this training phase was recorded.

In the test phase of the experiment participants first walked to a start object. When they had reached this object the maze disappeared and only a "textured groundplane" of the maze remained visible. The participants then turned toward the location of the target object that they had remembered during the training phase and started to walk toward the target. The angular difference, angular error, between the initial walking direction of a participant from the start object and the location of the target object was recorded as an outcome variable in the experiment.The angular error is a circular variable and can be described and analyzed using circular statistics.

Apart from the between-subjects factor, the type of maze, the experiment also included a within-subjects factor, trial number. All participants had to complete 8 trials in the test phase of the experiment. In each of these trials they had to walk to a specific target object. An additional within-subjects factor is the type of target object. Pairs of start and target objects were of two types: probe and standard. The probe objects were located near the entrance and exit of a wormhole whereas the standard objects were located at some distance from the wormholes. For each of these two types of objects participants had to find 4 different targets resulting in a total of 8 trials per participant.

Source

doi: 10.1016/j.cognition.2017.05.020

`mean_circ` *Compute the mean of a vector of circular data*

Description

Compute the mean of a vector of circular data

Usage

```
mean_circ(theta, units = "radians")
```

Arguments

`theta`	a circular variable in radians or degrees.
`units`	measurement units of the circular variable c("radians", "degrees").

Examples

```
library(bpnreg)
mean_circ(subset(Motor, Cond == "exp")$PhaseDiff, units = "degrees")
```

`mmme` *Create model matrices for a circular mixed-effects regression model*

Description

Create model matrices for a circular mixed-effects regression model

Usage

```
mmme(pred.I, data, pred.II)
```

Arguments

`pred.I`	model equation for effects of component 1
`data`	the dataframe used for analysis
`pred.II`	model equation for effects of component 2

`mmr` *Create model matrices circular regression*

Description

Create model matrices circular regression

Usage

```
mmr(pred.I, data, pred.II)
```

Arguments

`pred.I`	model equation for effects of component 1
`data`	the dataframe used for analysis
`pred.II`	model equation for effects of component 2

`mode_est` *Compute the mode of a vector of linear data*

Description

Compute the mode of a vector of linear data

Usage

```
mode_est(x)
```

Arguments

`x`	a vector of linear data

Examples

```
library(bpnreg)
mode_est(Motor$AvAmp)
```

mode_est_circc — *Compute the mode of a vector of circular data*

Description

Compute the mode of a vector of circular data

Usage

```
mode_est_circ(x)
```

Arguments

x	a vector of circular data

Examples

```
library(bpnreg)
mode_est_circ(subset(Motor, Cond = "exp")$Phaserad)
```

Motor — *Phase differences in hand flexion-extension movements.*

Description

A dataset from a study by Puglisi et.al. (2017) on the role of attention in human motor resonance.

Usage

```
Motor
```

Format

A data frame with 42 rows and 4 variables:

Cond a factor variable indicating the condition a participant was placed in; 1 = 'explicit', 2 = 'semi.implicit' or 3 = 'implicit'

PhaseDiff a numeric variable the phase difference between 'observer' and 'mover' in degrees

AvAmp a numeric variable indicating the average amplitude of the hand movement of the 'observer'

Phaserad a numeric variable the phase difference between 'observer' and 'mover' in radians

Details

In their research Puglisi et.al. (2017) conduct a between subjects experiment in which 'observers' in multiple degrees of explicitness are asked to look at the movement of a hand of the 'mover' or other object in order to evaluate the role of attention in motor resonant response.

The experiment has four conditions:

1. The 'explicit observation' condition (n = 14), where observers are explicitly instructed to observe the hand.

2. The 'semi-implicit observation' condition (n = 14) where the observers have to do a task that requires implicit observation of the hand.

3. The 'implicit observation' condition (n = 14), where observers have to do a task that is independent of the observation of the hand that is moving in front of them.

4. A baseline condition (n=14) where there is no moving hand but observers have to look at an inanimate object that moves in an identical manner to the hand.

The idea of motor resonance is then that the 'observer', because they are looking explicitly or implicitly at the hand of the 'mover', starts moving his or her hand in the same manner. This is the resonant response. In each condition the hand movements of the observers were measured and the phase difference between the observers' hand and the hand they observed (the movers' hand) was calculated. This was not done for the baseline condition because in this condition there was no periodic pattern of movement in the 'observers' hand.

The phase difference is a circular variable and can be described and analyzed using circular statistics.

Source

doi: 10.1371/journal.pone.0177457

`mvrnorm_arma_eigen` *Sample from a multivariate normal distribution*

Description

Sample from a multivariate normal distribution

Usage

```
mvrnorm_arma_eigen(n, mu, sigma)
```

Arguments

`n`	An integer indicating the number of samples to take.
`mu`	A mean vector.
`sigma`	A variance-covariance matrix.

pnme *A Gibbs sampler for a projected normal mixed-effects model*

Description

A Gibbs sampler for a projected normal mixed-effects model

Usage

```
pnme(
  theta_cos,
  theta_sin,
  X1,
  X2,
  Z1,
  Z2,
  ZtZ1,
  ZtZ2,
  R,
  pred,
  its,
  lag,
  burn,
  N
)
```

Arguments

theta_cos	A List with the cosine of the circular dependent variable.
theta_sin	A List with the sine of the circular dependent variable.
X1	A list of fixed effect model matrices for component I.
X2	A list of fixed effect model matrices for component II.
Z1	A list of random effect model matrices for component I.
Z2	A list of random effect model matrices for component II.
ZtZ1	A list of transformed random effect model matrices for component I.
ZtZ2	A list of transformed random effect model matrices for component II.
R	A list of starting values for R.
pred	An empty list for likelihood computation.
its	An integer specifying the number of iterations
lag	An integer specifying the amount of lag.
burn	An integer specifying the number of burn-in iterations.
N	An integer specifying the number of burn-in iterations.

pnr — *A Gibbs sampler for a projected normal regression model*

Description

A Gibbs sampler for a projected normal regression model

Usage

```
pnr(theta, X1, X2, its, lag, burn)
```

Arguments

theta	A vector with the circular dependent variable.
X1	A model matrix for component I.
X2	A model matrix for component II.
its	An integer specifying the number of iterations
lag	An integer specifying the amount of lag.
burn	An integer specifying the number of burn-in iterations.

print.bpnme — *Print output from a Bayesian circular mixed-effects model*

Description

Prints selected output from a Bayesian circular mixed-effects model.

Usage

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

Arguments

x	a `bpnme` `object` obtained from the function `bpnme()`.
...	further arguments passed to or from other methods.

Value

A print of selected output from a Bayesian circular mixed-effects model.

Examples

```
library(bpnreg)
fit.Maps <- bpnme(pred.I = Error.rad ~ Maze + Trial.type + L.c + (1|Subject),
data = Maps,
its = 100, burn = 1, n.lag = 1)
print(fit.Maps)
```

`print.bpnr` *Print output from a Bayesian circular regression model*

Description

Prints selected output from a Bayesian circular regression model.

Usage

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

Arguments

x	a bpnr object obtained from the function bpnr.
...	further arguments passed to or from other methods.

Value

A print of selected output from a Bayesian circular regression model.

Examples

```
library(bpnreg)
fit.Motor <- bpnr(pred.I = Phaserad ~ 1 + Cond, data = Motor,
its = 100, burn = 10, n.lag = 3)
print(fit.Motor)
```

rho — *Compute a mean resultant length*

Description

Compute a mean resultant length

Usage

```
rho(theta)
```

Arguments

theta	a circular variable in radians.

rho_circ — *Compute the mean resultant length of a vector of circular data*

Description

Compute the mean resultant length of a vector of circular data

Usage

```
rho_circ(theta, units = "radians")
```

Arguments

theta	a circular variable in radians or degrees.
units	measurement units of the circular variable c("radians", "degrees").

Examples

```
library(bpnreg)
rho_circ(subset(Motor, Cond == "exp")$PhaseDiff, units = "degrees")
```

sd_circ *Compute the standard deviation of a vector of circular data*

Description

Compute the standard deviation of a vector of circular data

Usage

```
sd_circ(theta, units = "radians")
```

Arguments

theta	a circular variable in radians or degrees.
units	measurement units of the circular variable c("radians", "degrees").

Examples

```
library(bpnreg)
sd_circ(subset(Motor, Cond == "exp")$PhaseDiff, units = "degrees")
```

slice_rcpp *A slice sampler for the latent lengths r*

Description

A slice sampler for the latent lengths r

Usage

```
slice_rcpp(X1, X2, theta, beta1, beta2, n, r)
```

Arguments

X1	A model matrix for component I.
X2	A model matrix for component II.
theta	A vector with the circular dependent variable.
beta1	A matrix containing the coefficients of component I for the current iteration.
beta2	A matrix containing the coefficients of component II for the current iteration.
n	An integer indicating the sample size of the data.
r	A matrix with the estimates of r of the previous iteration.

theta_bar	*Compute a mean direction*

Description

Compute a mean direction

Usage

```
theta_bar(theta)
```

Arguments

theta	a circular variable in radians.

traceplot	*Traceplots*

Description

Traceplot function for a bpnr object or bpnme object.

Usage

```
traceplot(object, parameter = "SAM", variable = NULL)
```

Arguments

object	an object used to select a method.
parameter	one of c("b1", "b2", beta1", "beta2", a.x", "a.c","b.c", "SAM", "AS", "SSDO", "circ.diff", "omega1", "omega2", "cRI", "cRS") to indicate for which parameter a traceplot is required. beta1 and beta2 are the linear intercepts and coefficients of the first and second component for a regression model and the fixed effects coefficients of a mixed-effects model. b1 and b2 are the random effects coefficients of a mixed-effects model. circ.diff are the circular differences with the intercept on the outcome variable for the different levels of categorical variables. omega1 and omega2 are the linear random effect variances and cRI and cRS are the variances of the circular random intercept and circular random slope.
variable	a character string with variable name(s) to indicate for which variable(s) a traceplot is required.

Examples

```
library(bpnreg)
fit.Motor <- bpnr(pred.I = Phaserad ~ 1 + Cond, data = Motor,
its = 100, burn = 10, n.lag = 3)
traceplot(fit.Motor, parameter = "beta1")
```

`traceplot.bpnme` *Traceplots for a Bayesian circular mixed-effects model*

Description

General plot function for a `bpnme object`.

Usage

```
## S3 method for class 'bpnme'
traceplot(object, parameter = "SAM", variable = NULL)
```

Arguments

`object` a `bpnme object` obtained from the function `bpnme`

`parameter` one of `c(beta1", "beta2", "b1", "b2", a.x", "a.c","b.c", "SAM", "AS", "SSDO", "circ.diff", "omega1", "omega2", "cRI", "cRS")` to indicate for which parameter a traceplot is required. `beta1` and `beta2` are the fixed effects coefficients of a mixed-effects model. `b1` and `b2` are the random effects coefficients of a mixed-effects model. `circ.diff` are the circular differences with the intercept on the outcome variable for the different levels of categorical variables. `omega1` and `omega2` are the linear random effect variances and `cRI` and `cRS` are the variances of the circular random intercept and circular random slope.

`variable` a character string with variable name(s) to indicate for which variable(s) a traceplot is required. This cannot be used in combination with the parameters `c("omega1", "omega2", "cRI", "cRS")`.

Examples

```
library(bpnreg)
fit.Maps <- bpnme(pred.I = Error.rad ~ Maze + Trial.type + L.c + (1|Subject),
data = Maps,
its = 100, burn = 1, n.lag = 1)
traceplot(fit.Maps)
```

traceplot.bpnr *Traceplots for a Bayesian circular regression model*

Description

General plot function for a bpnr object.

Usage

```
## S3 method for class 'bpnr'
traceplot(object, parameter = "SAM", variable = NULL)
```

Arguments

object	a bpnr object obtained from the function bpnr
parameter	one of c("beta1", "beta2", "a.x", "a.c", "b.c", "SAM", "AS","SSDO", "circ.diff") to indicate for which parameter a traceplot is required. beta1 and beta2 are the linear intercepts and coefficients of the first and second component. circ.diff are the circular differences with the intercept on the outcome variable for the different levels of categorical variables.
variable	a character string with variable name(s) to indicate for which variable(s) a traceplot is required.

Examples

```
library(bpnreg)
fit.Motor <- bpnr(pred.I = Phaserad ~ 1 + Cond, data = Motor,
its = 100, burn = 10, n.lag = 3)
traceplot(fit.Motor, parameter = "beta1")
```

Index