Package 'bayesbr'

July 16, 2021

Title Beta Regression on a Bayesian Model

Version 0.0.1.0

Description Applies the Beta regression model in the Bayesian statistical view with the possibility of adding a spatial effect in the parameters, the Beta regression is used when the response variable is a proportion variable, that is, it only accepts values between 0 and 1. The package 'bayesbr' uses 'rstan' package to build the Bayesian statistical models. The main function of the package receives as a parameter a form informing the independent variable and the co-variables of the model to be made, as output it returns a list with the results of the model. For more details see Ferrari and Cribari-Neto (2004) <doi:10.1080/0266476042000214501> and Hoffman and Gelman (2014) <arXiv:1111.4246>.

Encoding UTF-8

LazyData true

Biarch true

Depends R (>= 3.5.0)

Imports methods, Rcpp (>= 0.12.0), rstan (>= 2.18.1), rstantools (>= 2.0.0), coda, tidyr, dplyr, stringr, ggplot2, magrittr, fdrtool (>= 1.2.15), Formula(>= 1.2-3), loo(>= 2.2.0), RcppParallel

LinkingTo BH (>= 1.66.0), Rcpp (>= 0.12.0), RcppEigen (>= 0.3.3.3.0), rstan (>= 2.18.1), StanHeaders (>= 2.18.0), RcppParallel

SystemRequirements GNU make

License GPL-2

RoxygenNote 7.1.1

URL `https://github.com/pjoao266/bayesbr`

BugReports `https://github.com/pjoao266/bayesbr/issues`

Suggests knitr, rmarkdown, shiny, highcharter, DT, shinydashboard, dashboardthemes, shinyalert, shinyjs, openxlsx, tidyverse

VignetteBuilder knitr

NeedsCompilation yes

Author Joao Melo [aut, cre],
Vinicius Mayrink [aut]

Maintainer Joao Melo <pjoao266@gmail.com>

Repository CRAN

Date/Publication 2021-07-16 19:00:02 UTC

R topics documented:

bayesbr-package *bayesbr: A package for Bayesian Beta Regression*

Description

The package fits or the beta regression model under the view of Bayesian statistics using the No-U-Turn-Sampler (NUTS) method for computational calculations.The model can be adjusted considering or not the spatial effect on the parameters. In addition to showing the coefficients, the package also has functions for displaying residuals, checking the model's convergence, checking the quality of the model and other utilities that may be useful.

References

arXiv:1111.4246 Hoffman, M. D., and Gelman, A. (2014). The No-U-Turn sampler: adaptively setting path lengths in Hamiltonian Monte Carlo. *Journal of Machine Learning Research*, **15**(1), 1593-1623.

doi: 10.1080/0266476042000214501 Ferrari, S.L.P., and Cribari-Neto, F. (2004). Beta Regression for Modeling Rates and Proportions. *Journal of Applied Statistics*, **31**(7), 799–815.

AIC_bayesbr *Akaike Information Criterion*

Description

A function that receives the estimated model data, uses the information from the loglik and the number of estimated parameters and returns the AIC, an estimator for the quality of the estimation of a model.

Usage

```
AIC_bayesbr(x)
```

Arguments

x	an object of the class *bayesbr*, containing the list returned from the `bayesbr` function.

Details

Proposed by Akaike (1974) the AIC (Akaike Information Criterion) measures the quality of the adjustment made by the model, when comparing adjusted models with the same data, the smaller the AIC the better the adjustment.

The AIC theory requires that the log-likelihood has been maximized, but as we are in the context of Bayesian statistics, the log-likelihood as explained in the `logLik.bayesbr` is made with the average of the a priori distribution for each theta and applying this value in the formula to calculate the loglik. The AIC is calculated by

$$AIC = 2 * k - 2 * L,$$

where k is the number of covariates used in the model, and L is the average of the loglik chain returned by the function logLik.bayesbr.

Value

A number corresponding to the AIC (Akaike Information Criterion) of the estimated model.

References

doi: 10.1109/TAC.1974.1100705 Akaike, H. (1974). A new look at the statistical model identification. *IEEE transactions on automatic control*, **19**(6), 716-723.

See Also

logLik.bayesbr,BIC_bayesbr,DIC_bayesbr

Examples

```
data("CarTask",package = "bayesbr")

car_bayesbr <- bayesbr(probability ~ NFCCscale + task,
                       data = CarTask,iter =100)
aic = AIC_bayesbr(car_bayesbr)
```

---

bayesbr *Bayesian Beta Regression with RStan*

Description

Fit of beta regression model under the view of Bayesian statistics, using the mean of the posterior distribution as estimates for the mean (theta) and the precision parameter (zeta).

Usage

```
bayesbr(formula=NULL,data=NULL,m_neighborhood = NULL,
na.action=c("exclude","replace"),mean_betas = NULL,
variance_betas = NULL,mean_gammas = NULL,
variance_gammas = NULL ,iter = 10000,warmup = iter/2,
chains = 1,pars=NULL,a = NULL,b = NULL,
atau_delta = NULL, btau_delta = NULL,atau_xi = NULL,
btau_xi = NULL,rho = NULL,spatial_theta = NULL,spatial_zeta=NULL,
resid.type = c("quantile","sweighted",
            "pearson","ordinary"),...)
```

Arguments

- `formula` — symbolic description of the model (of type `y ~ x` or `y ~ x | z`;). See more at `formula`
- `data` — data frame or list with the variables passed in the formula parameter, if `data = NULL` the function will use the existing variables in the global environment.
- `m_neighborhood` — A neighborhood matrix with n rows and n columns, with n the number of observations of the model to be adjusted. This matrix should only contain a value of 0 on the main diagonal, and a value of 0 or 1 at position i j, to inform whether observation i is next to observation j. It must be symmetric, because if i is a neighbor of j, j is also a neighbor of i. This matrix will be used to calculate the model's covariance matrix, if one of the conditions is not accepted or the neighborhood matrix is not informed, the model will be adjusted without the spatial effect.
- `na.action` — Characters provided or treatment used in NA values. If `na.action` is equal to exclude (default value), the row containing the NA will be excluded in all variables of the model. If `na.action` is equal to replace, the row containing the NA will be replaced by the average of the variable in all variables of the model.
- `mean_betas, variance_betas` — vectors including a priori information of mean and variance for the estimated beta respectively, beta is the name given to the coefficient of each covariate that influences theta. PS: the size of the vectors must equal p + 1, p being the number of covariates for theta.
- `mean_gammas, variance_gammas` — vectors including a priori information of mean and variance for the estimated ranges respectively, gamma is the name given to the coefficient of each covariate that influences zeta. PS: the size of the vectors must be equal to q + 1, q being the number of covariates for zeta.
- `iter` — A positive integer specifying the number of iterations for each chain (including warmup). The default is 10000.
- `warmup` — A positive integer specifying the number of iterations that will be in the warm-up period, will soon be discarded when making the estimates and inferences. Warmup must be less than `iter` and its default value is `iter/2`.
- `chains` — A positive integer specifying the number of Markov chains. The default is 1.
- `pars` — A vector of character strings specifying parameters of interest. The default is NULL indicating all parameters in the model.
- `a, b` — Positive integer specifying the a priori information of the parameters of the gamma distribution for the zeta, if there are covariables explaining zeta `a` and `b` they will not be used. The default value for `a` is 1 and default value for `b` is 0.01 .
- `atau_delta, btau_delta, atau_xi, btau_xi` — Positive integer specifying the a priori information of tau parameter of the gamma distribution. The default value for `atau_delta` and `atau_xi` is 0.1 and default value for `btau_delta` and `btau_xi` is 0.1.
- `rho` — value of the time scaling parameter for calculate the covariance matrix.

`spatial_theta, spatial_zeta`
: A Boolean variable to inform whether the adjusted model will have an effect on the theta parameter, or on the zeta parameter or both parameters.

`resid.type`
: A character containing the residual type returned by the model among the possibilities. The type of residue can be *quantile*, *sweighted*, *pearson* or *ordinary*. The default is *quantile*.

`...`
: Other optional parameters from RStan

Details

Beta Regression was suggested by Ferrari and Cribari-Neto (2004), but with the look of classical statistics, this package makes use of the `Rstan` to, from the prior distribution of the data, obtain the posterior distribution and the estimates from a Bayesian perspective. Beta regression is useful when the response variable is in the range between 0 and 1, being used for adjusting probabilities and proportions.

It is possible to estimate coefficients for the explanatory covariates for the theta and zeta parameters of the Beta distribution. Linear predictors are passed as parameters for both zeta and zeta, from these linear predictors a transformation of scales is made.

Hamiltonian Monte Carlo (HMC) is a Markov chain Monte Carlo (MCMC) algorithm, from the HMC there is an extension known as the No-U-Turn Sampler (NUTS) that makes use of recursion to obtain its calculations and is used by `RStan`. In the context of the `bayesbr` package, NUTS was used to obtain a posteriori distribution from model data and a priori distribution.

See `predict.bayesbr`, `residuals.bayesbr`,`summary.bayesbr`,`logLik.bayesbr` and `pseudo.r.squared` for more details on all methods. Because it is in the context of Bayesian statistics, in all calculations that were defined using maximum verisimilitude, this was sub-replaced by the mean of the posterior distribution of the parameters of interest of the formula.

Value

`bayesbr` return an object of class *bayesbr*, a list of the following items.

coefficients a list with the mean and precision elements containing the estimated coefficients of model and table with the means, medians, standard deviations and the Highest Posterior Density (HPD) Interval,

call the original function call,

formula the original formula,

y the response proportion vector,

stancode lines of code containing the .STAN file used to estimate the model,

info a list containing model information such as the argument pars passed as argument, name of variables, indicator for effect spatial in model, number of: iterations, warmups, chains, covariables for theta, covariables for zeta and observations of the sample. In addition there is an element called samples, with the posterior distribution of the parameters of interest,

fitted.values a vector containing the estimates for the values corresponding to the theta of each observation of the variable response, the estimate is made using the mean of the a prior theta distribution,

model the full model frame,

residuals a vector of residuals,

residuals.type the type of returned residual,

delta a matrix with the means, medians, standard deviations and the Highest Posterior Density (HPD) Interval of the delta parameter (spatial effect in theta parameter). The estimation for the delta parameter, informs the influence that a given region has on the response variable, neighboring observations are expected to have close estimates for delta.

xi a matrix with the means, medians, standard deviations and the Highest Posterior Density (HPD) Interval of the xi parameter (spatial effect in zeta parameter). The estimation for the xi parameter, informs the influence that a given region has on the response variable, neighboring observations are expected to have close estimates for xi.

loglik log-likelihood of the fitted model(using the mean of the parameters in the posterior distribution),

AIC a value containing the Akaike's Information Criterion (AIC) of the fitted model,

BIC a value containing the Bayesian Information Criterion (BIC) of the fitted model,

DIC a value containing the Deviance Information Criterion (DIC) of the fitted model,

WAIC a vector containing the Widely Applicable Information Criterion (WAIC) of the fitted model and their standard error, see more in `waic`

LOOIC a vector containing the LOO (Efficient approximate leave-one-out cross-validation) Information Criterion of the fitted model and their standard error, see more in `loo`

pseudo.r.squared pseudo-value of the square R (correlation to the square of the linear predictor and the a posteriori means of theta).

References

doi: 10.1080/0266476042000214501 Ferrari, S.L.P., and Cribari-Neto, F. (2004). Beta Regression for Modeling Rates and Proportions. *Journal of Applied Statistics*, **31**(7), 799–815.

arXiv:1111.4246 Hoffman, M. D., and Gelman, A. (2014). The No-U-Turn sampler: adaptively setting path lengths in Hamiltonian Monte Carlo. *Journal of Machine Learning Research*, **15**(1), 1593-1623.

doi: 10.18637/jss.v076.i01 Carpenter, B., Gelman, A., Hoffman, M. D., Lee, D., Goodrich, B., Betancourt, M., ... & Riddell, A. (2017). Stan: A probabilistic programming language. *Journal of statistical software*, **76**(1).

See Also

`summary.bayesbr`, `residuals.bayesbr`, `formula`

Examples

```
data("StressAnxiety",package="bayesbr")

bbr = bayesbr(anxiety ~ stress | stress, data = StressAnxiety,
              iter = 100)
summary(bbr)
residuals(bbr, type="ordinary")
print(bbr)
```

```
data("StressAnxiety", package = "bayesbr")
bbr2 <- bayesbr(anxiety ~ stress | stress,
               data = StressAnxiety, iter = 1000,
               warmup= 450, mean_betas = c(0,1),
               variance_betas = 15)

envelope(bbr2,sim=100,conf=0.95)
loglikPlot(bbr2$loglik)
```

---

bayesbr_app *The bayesbr Application*

Description

A function that runs the shiny application designed for using bayesbr package functions through a visual interface.

Usage

```
bayesbr_app()
```

Details

See the application manual: How to use bayesbr shiny app in vignnetes of bayesbr package.

See Also

bayesbr

BIC_bayesbr *Bayesian Information Criterion*

Description

A function that receives data from the estimated model, uses the information from the loglik, the number of observations of the model and the number of estimated parameters and returns the BIC, an estimator for the quality of the estimation of a model.

Usage

```
BIC_bayesbr(x)
```

Arguments

`x`	an object of the class *bayesbr*, containing the list returned from the `bayesbr` function.

Details

Proposed by Stone (1979) the BIC (Bayesian Information Criterion) measures the quality of the adjustment made by the model, when comparing adjusted models with the same data, the smaller the BIC the better the adjustment.

The BIC theory requires that the log-likelihood has been maximized, but as we are in the context of Bayesian statistics, the log-likelihood as explained in the `logLik.bayesbr` is made with the average of the a priori distribution for each theta and applying this value in the formula to calculate the loglik.

The BIC is calculated by

$$BIC = log(n) * k - 2 * L,$$

where `n` is the number of observations of the model variables, `k` is the number of covariates used in the model, and L is the average of the loglik chain returned by the function `logLik.bayesbr`.

Value

A number corresponding to the BIC (Bayesian Information Criterion) of the estimated model.

References

Schwarz, G. (1978). Estimating the dimension of a model. *The annals of statistics*, **6**(2), 461-464.

See Also

`bayesbr`, `AIC_bayesbr`, `DIC_bayesbr`

Examples

```
data("CarTask",package = "bayesbr")

car_bayesbr <- bayesbr(probability ~ NFCCscale + task, data = CarTask,
                       iter =100)
bic = BIC_bayesbr(car_bayesbr)
```

`bodyfat` *Percentage of Body Fat*

Description

A data frame that contains the proportion of cord fat for individuals calculated through various body measurements of weight, height and circumferences of 252 men who participated in the study by Dr. A. Garth Fisher, Human Performance Research Center, Brigham Young University.

Usage

```
data(bodyfat)
```

Format

This data frame contains the observations of 252 men:

case Case number.

brozek Percent body fat using Brozek's equation: 457/Density - 414.2

siri Percent body fat using Siri's equation: 495/Density - 450

density Density determined from underwater weighing (gm/cm**3).

age Age (years).

weight Weight (kg/100).

height Height (m).

neck Neck circumference (m).

chest Chest circumference (m).

abdomen Abdomen circumference (m) "at the umbilicus and level with the iliac crest".

forearm Forearm circumference (m).

hip Hip circumference (m).

thigh Thigh circumference (m).

knee Knee circumference (m).

ankle Ankle circumference (m).

biceps Biceps (extended) circumference (m).

wrist Wrist circumference (m) "distal to the styloid processes".

Details

It is possible to find some errors in the table or strange data:

One man (case 42) was measured with over 200 pounds in weight who is less than 3 feet tall, considered that he had a typo when typing 29.5 inches and transformed the data into 69.5 inches;

There was a man with a negative percentage of body fat, it was decided to exclude this data from the table.

Changes to units of measure:

Weight was transformed from lbs to kg / 100 (value 1 corresponds to 100kg);

Height has been transformed from inches to meters;

All columns that were represented in centimeters were transformed into meters.

References

doi: 10.1080/10691898.1996.11910505 Johnson, R. W. (1996). Fitting percentage of body fat to simple body measurements. *Journal of Statistics Education*, **4**(1).

doi: 10.1249/00005768198504000000037 Penrose, K. W., Nelson, A. G., & Fisher, A. G. (1985). Generalized body composition prediction equation for men using simple measurement techniques. *Medicine & Science in Sports & Exercise*, **17**(2), 189.

doi: 10.1016/j.csda.2006.05.006 Royston, P., & Sauerbrei, W. (2007). Improving the robustness of fractional polynomial models by preliminary covariate transformation: A pragmatic approach. *Computational statistics & data analysis*, **51**(9), 4240-4253.

Examples

```
data(bodyfat,package="bayesbr")

bbr = bayesbr(siri ~ age+wrist*neck+chest+
            thigh+wrist| wrist, data = bodyfat,
            iter = 100)

summary(bbr)


bbr = bayesbr(siri ~ I(age/100)+height+chest+
            thigh+wrist| wrist,
            data = bodyfat,iter = 1000)
```

CarTask *Probability Judgment for Car Dealership with Partition*

Description

Participants who responded to the study were expected to judge the likelihood of a customer trades in a coupe or that a customer buys a car from a specific seller among four possible sellers.

Usage

```
data(CarTask)
```

Format

A data frame with 155 observations on the following 3 variables.

`task` A variable specified as conditions. When 0 the set value is `Car`, when 1 the set value is `Salesperson`.

`probability` a numeric vector of the estimated probability.

`NFCCscale` a numeric vector of the NFCC scale.

Details

Study participants were graduate students from The Australian National University, some students received credits in Psychology for participating in the study.

With the Needs for Closing and Needs for Certainty scales strongly correlated, the NFCCscale is a combined scale between the previous two.

For `task` the questions were:

Car What is the probability that a customer trades in a coupe?

Salesperson What is the probability that a customer buys a car from Carlos?

The `task` variable that was a qualitative variable was transformed into a quantitative variable to be used by the package functions.

References

doi: 10.3102/1076998610396893 Smithson, M., Merkle, E.C., and Verkuilen, J. (2011). Beta Regression Finite Mixture Models of Polarization and Priming. *Journal of Educational and Behavioral Statistics*, **36**(6), 804–831.

doi: 10.1080/15598608.2009.10411918 Smithson, M., and Segale, C. (2009). Partition Priming in Judgments of Imprecise Probabilities. *Journal of Statistical Theory and Practice*, **3**(1), 169–181.

Examples

```
data("CarTask", package = "bayesbr")

car_bayesbr <- bayesbr(probability ~ NFCCscale + task, data = CarTask,
                       iter =100)
```

`DIC_bayesbr` *Deviance Information Criterion*

Description

A function that receives data from the estimated model, uses the information from the loglik and returns the DIC, an estimator for the quality of the estimation of a model.

Usage

```
DIC_bayesbr(x)
```

Arguments

x	an object of the class *bayesbr*, containing the list returned from the bayesbr function.

Details

Proposed by Spiegelhalter (2002) the DIC (Deviance Information Criterion) measures the quality of the adjustment made by the model, when comparing adjusted models with the same data, the smaller the BIC the better the adjustment.

It is particularly useful in Bayesian model selection problems where the posterior distributions of the models have been obtained by Markov chain Monte Carlo (MCMC) simulation. DIC is an asymptotic approximation as the sample size becomes large, like AIC. It is only valid when the posterior distribution is approximately multivariate normal.

DIC is calculate using the loglik calculated from the posterior distribution of the parameters and a calculation from the average of the posterior distribution of the parameters. To see the formula visit Spiegelhalter (2002).

Value

A number corresponding to the DIC (Deviance Information Criterion) of the estimated model.

References

doi: 10.1111/14679868.00353 Spiegelhalter, D. J., Best, N. G., Carlin, B. P., & Van Der Linde, A. (2002). Bayesian measures of model complexity and fit. *Journal of the royal statistical society: Series b (statistical methodology)*, **64**(4), 583-639.

doi: 10.1111/j.14679574.2005.00278.x Van Der Linde, A. (2005). DIC in variable selection. *Statistica Neerlandica*, **59**(1), 45-56.

See Also

bayesbr, AIC_bayesbr, BIC_bayesbr

Examples

```
data("CarTask",package="bayesbr")

car_bayesbr <- bayesbr(probability ~ NFCCscale + task, data = CarTask,
                       iter =100)
dic = DIC_bayesbr(car_bayesbr)
```

envelope — *Envelope Graph: Residuals vs Half-Normal Values*

Description

A graph showing the absolute values of the residuals ordered against the quantiles of simulations of the half-normal distribution.

Usage

```
envelope(x, sim = 1000, conf = 0.95, resid.type = c("",
"quantile", "sweighted","pearson","ordinary"))
```

Arguments

`x`	an object of the class *bayesbr*, containing the list returned from the `bayesbr` function.
`sim`	a positive integer containing the number of simulations of the half-normal distribution.
`conf`	a probability containing the confidence level for the quantiles made under the half-normal samples.
`resid.type`	the residual type that will be used in the graph

Details

Atkinson (1985) proposed to use quantiles from a simulated population of the halfnormal distribution, this is used because (blablabla read the book, right). From the distribution of the absolute values of the residual in the graph, it is possible to measure the quality of the model estimation.

Value

A graph showing the absolute values of the residuals ordered against the quantiles of simulations of the half-normal distribution.

References

Atkinson, A. C. (1985). Plots, transformations, and regression: An introduction to graphical methods of diagnostic regression analysis. *Oxford: Clarendon Press*.

See Also

`residuals.bayesbr`, `loglikPlot`, `bayesbr`

Examples

```
data("CarTask", package = "bayesbr")

bbr = bayesbr(probability~task + NFCCscale, iter = 100,
            data=CarTask, mean_betas = c(1, 0.5,1.2),variance_betas=10)

envelope(bbr,sim = 100, conf=0.9, resid.type="quantile")

envelope(bbr,sim = 1000, conf=0.99, resid.type="ordinary")
```

`fitted.values` *Fitted Values for Theta on Beta Regression*

Description

A function that receives information from an estimated model uses data from the estimated theta for each iteration and returns the average of each theta in the sample.

Usage

```
fitted.values(x)
```

Arguments

`x`	an object of the class *bayesbr*, containing the list returned from the `bayesbr` function.

Value

A vector with the average of theta estimates in the iterations (excluding warmup). The vector size is equal to the number of model observations.

See Also

`bayesbr`,`predict.bayesbr`

`FoodExpenditure` *Spending on Food by Household Income*

Description

Data frame on the proportion of food expenses per household income. 38 house rents were evaluated in a random sample from a large city in the United States.

Usage

```
data("FoodExpenditure")
```

Format

A data frame containing 38 observations on 3 variables.

food household expenditures for food.

income household income.

proportion proportion of household income spent on food.

persons number of persons living in household.

Details

Originally, the `proportion` column did not exist, it was created by the bayesbr package.

Source

Taken from Griffiths et al. (1993, Table 15.4).

References

doi: 10.18637/jss.v034.i02 Cribari-Neto, F., and Zeileis, A. (2010). Beta Regression in R. *Journal of Statistical Software*, **34**(2), 1–24.

doi: 10.1080/0266476042000214501 Ferrari, S.L.P., and Cribari-Neto, F. (2004). Beta Regression for Modeling Rates and Proportions. *Journal of Applied Statistics*, **31**(7), 799–815.

doi10.1002/jae.3950090208 Griffiths, W.E., Hill, R.C., and Judge, G.G. (1993). *Learning and Practicing Econometrics* New York: John Wiley and Sons.

Examples

```
data("FoodExpenditure", package = "bayesbr")

bbr <- bayesbr(proportion ~ income + persons, data = FoodExpenditure,
               iter=100)
residuals(bbr, type="quantile")
```

```
pmse <-pmse(proportion ~ income + persons, test.set=0.4,
          data = FoodExpenditure, iter=100)$PMSE
```

`formula` *Formula Variables*

Description

Transforming a formula object into a list with the variables and their names for the beta regression model of the bayesbr package.

Usage

```
formula(formula, data = NULL)
```

Arguments

`formula`	symbolic description of the model (of type `y ~ x` or `y ~ x | z;`).
`data`	Data frame with regression observations

Details

The form of the formula used for the `Bayesbr` package follows the pattern proposed in `Formula`. The expression y ~ represents that y is the response variable of the beta regression, everything to the right of the ~ operator represents covariates or intercepts for the parameter θ or ζ of the variable response .

The + operator adds one more explanatory covariate for the parameter,the operator : indicates interaction between variables adjacent to the operator, operator * adds the variables adjacent to the operator as covariable and the interaction between them the operator | represents that the next covariates are explanatory for ζ and those that were before the operator are explanatory for θ. So, in the formula `y ~ x1 + x2 | x3 + x4` x1 and x2 are the covariates for the parameter θ and x3 and x4 are the covariates of ζ. θ and ζ are parameters of the variable y answer. The numbers 1 and 0 represent, respectively, the presence or not of the intercept in the construction of the model. By default, the intercept is included, so the number 1 is necessary only when the user wants to include only the intercept for the estimation of the parameter in question. Here are some examples:

`y ~ 0 | x1`: No estimate for θ

`y ~ 1 | 0 + x2`: The estimation for θ will be made only with the intercept, and the estimation for ζ will not use the intercept only the covariable x2

`y~ x3*x4 | x5:x6`: The estimation for θ will be with the covariables `x3` and `x4` and the interaction between them, and the estimation for ζ will be the interaction between variables `x5` and `x6`.

The variables passed to the formula can be environment variables or columns of a dataframe, in which case the dataframe must be informed.

Value

A list containing the following items:

Y A vector containing the model response variable,

X A matrix containing the covariates for theta of the model,

W A matrix containing the covariates of the model for zeta

name_y The name passed in the call to the `bayesbr` function for the variable response,

name_x The name passed in the call to the `bayesbr` function for the covariates for theta,

name_w The name passed in the call to the `bayesbr` function for covariates for zeta.

See Also

`bayesbr`

`GasolineYield` *Estimation of Gasoline Yields from Crude Oil*

Description

Proportion of crude oil converted to gasoline after the transformation processes.

Usage

```
data("GasolineYield")
```

Format

A data frame containing 32 observations on 6 variables.

yield proportion of crude oil converted to gasoline after distillation and fractionation.

gravity crude oil gravity (degrees API).

pressure vapor pressure of crude oil (lbf/in2).

temp10 temperature (degrees F) at which 10 percent of crude oil has vaporized.

temp temperature (degrees F) at which all gasoline has vaporized.

batch factor indicating unique batch of conditions `gravity`, `pressure`, and `temp10`.

Details

This dataset were analyzed by Atkinson (1985) when he used a linear regression model and observed that the linear regression model failed to describe the data well, generating large residues.

The dataset contains 32 observations on the response and on the independent variables. It was observed that there are only ten sets of values for the first three explanatory variables, so these sets served as conditions for controlled distillation. These conditions are listed in the variable \ code batch.

With the Needs for Closing and Needs for Certainty scales strongly correlated, the NFCCscale is a combined scale between the previous two.

References

Atkinson, A.C. (1985). *Plots, Transformations and Regression: An Introduction to Graphical Methods of Diagnostic Regression Analysis*. New York: Oxford University Press.

doi: 10.18637/jss.v034.i02 Cribari-Neto, F., and Zeileis, A. (2010). Beta Regression in R. *Journal of Statistical Software*, **34**(2), 1–24.

Daniel, C., and Wood, F.S. (1971). *Fitting Equations to Data*. New York: John Wiley and Sons.

doi: 10.1080/0266476042000214501 Ferrari, S.L.P., and Cribari-Neto, F. (2004). Beta Regression for Modeling Rates and Proportions. *Journal of Applied Statistics*, **31**(7), 799–815.

Examples

```
data("GasolineYield", package = "bayesbr")


bbr = bayesbr(yield ~ temp + batch, iter = 100,
             data = GasolineYield)

envelope(bbr, conf=0.95, sim = 100, resid.type="quantile")
```

`ImpreciseTask` *Imprecise Probabilities for Sunday Weather and Boeing Stock Task*

Description

In this study, participants had to respond to the greater and lesser probability of the event happening.

Usage

```
data(ImpreciseTask)
```

Format

A data frame with 242 observations on the following 3 variables.

`task` a variable with responses 0 and 1. If 0 `task` is `Boeing stock`, if 1 `task` is `Sunday weather`.

`location` a numeric vector of the average of the lower estimate for the event not to occur and the upper estimate for the event to occur.

`difference` a numeric vector of the differences of the lower and upper estimate for the event to occur.

Details

All study participants were from the first or second year, none of the participants had an in-depth knowledge of probability.

For the sunday weather task see WeatherTask. For the Boeing stock task participants were asked to estimate the probability that Boeing's stock would rise more than those in a list of 30 companies. For each task participants were asked to provide lower and upper estimates for the event to occur and not to occur.

The task variable that was a qualitative variable was transformed into a quantitative variable to be used by the package functions.#'

References

doi: 10.3102/1076998610396893 Smithson, M., Merkle, E.C., and Verkuilen, J. (2011). Beta Regression Finite Mixture Models of Polarization and Priming. *Journal of Educational and Behavioral Statistics*, **36**(6), 804–831.

doi: 10.3102/1076998610396893 Smithson, M., and Segale, C. (2009). Partition Priming in Judgments of Imprecise Probabilities. *Journal of Statistical Theory and Practice*, **3**(1), 169–181. *Journal of Educational and Behavioral Statistics*, **36**(6), 804–831.

Examples

```
data("ImpreciseTask", package = "bayesbr")

bbr = bayesbr(location~difference,iter=100,
            data = ImpreciseTask)
```

---

logLik.bayesbr *Model Log Likelihood for* bayesbr *Objects*

Description

A function that receives the information from the estimated model, the response variable and the theta and zeta chains and returns a vector containing loglik values for each iteration excluding warmups.

Usage

```
## S3 method for class 'bayesbr'
logLik(object,...)
```

Arguments

object	an object of the class *bayesbr*, containing the list returned from the bayesbr function.
...	further arguments passed to or from other methods.

Details

Loglik is commonly used to measure fit quality, or to assess whether an fit has converged. The loglik is calculated using maximum likelihood, but as we are in the Bayesian context we will use the mean of the posterior distribution of the parameters, so the calculation occurs from an adaptation of the original form to the loglik.

Value

The function returns a list with

loglik A vector with the estimated model loglik chain,

matrix_loglik A matrix with all loglik's chain.

References

doi: 10.1080/0266476042000214501 Ferrari, S.L.P., and Cribari-Neto, F. (2004). Beta Regression for Modeling Rates and Proportions. *Journal of Applied Statistics*, **31**(7), 799–815.

See Also

bayesbr,residuals.bayesbr,loglikPlot

Examples

```
data("CarTask", package = "bayesbr")
bbr = bayesbr(probability~task + NFCCscale, iter = 100,
              data=CarTask, mean_betas = c(1, 0.5,1.2))
loglik = bbr$loglik

loglikPlot(loglik)
```

---

loglikPlot *Plot Chain of Loglik Using GGplot*

Description

The function receives a vector containing the model's loglik chain and displays it through a graph using the GGplot package, through this graph it is possible to see if the model has converged.

Usage

```
loglikPlot(loglik)
```

Arguments

`loglik`	A vector with the estimated model loglik chain

See Also

logLik.bayesbr,envelope

Examples

```
data("CarTask", package = "bayesbr")

bbr = bayesbr(probability~task + NFCCscale,data = CarTask,
              iter = 100, mean_betas = c(1, 0.5,1.2))
loglik = bbr$loglik

loglikPlot(loglik)
```

MockJurors *Mock Jurors' Confidence in Their Verdicts*

Description

Answers from mock jurors. It presents the difference in the juror's confidence in a conventional two-option verdict (guilt x absolution) versus a three-option verdict (the new option is "unproven"), in the presence or absence of conflicting testimonial evidence.

Usage

```
data("MockJurors")
```

Format

A data frame containing 104 observations on 3 variables.

verdict a variable indicating whether a two-option or three-option verdict is requested. If `verdict` is 0 is interpreted as two-option, if `verdict` is 1 is interpreted as three-option.

conflict Is there conflicting testimonial evidence? If 0, yes. If 1, no.

confidence jurors degree of confidence in his/her verdict, scaled to the open unit interval.

Details

The data were collected by Professor Daily at the Australian National University among first-year psychology students. Smithson and Verkuilen (2006) used the original confidence data and transformed it to a scale of 0 to 1, using the following calculation: `((original_confidence/100) * 103 -0.5) / 104`.

The `verdict` and `conflict` variables that was a qualitative variable was transformed into a quantitative variable to be used by the package functions.

Source

Example 1 from Smithson and Verkuilen (2006) supplements.

References

doi: 10.1037/1082989X.11.1.54 Smithson, M., and Verkuilen, J. (2006). A Better Lemon Squeezer? Maximum-Likelihood Regression with Beta-Distributed Dependent Variables. *Psychological Methods*, **11**(7), 54–71.

doi: 10.1080/10888430709336633 Pammer, K., and Kevan, A. (2004). The Contribution of Visual Sensitivity, Phonological Processing and Non-Verbal IQ to Children's Reading. *Unpublished manuscript*, The Australian National University, Canberra.

Examples

```
data("MockJurors", package = "bayesbr")


bbr = bayesbr(confidence~verdict+conflict, iter=1000,
              data = MockJurors)
```

---

model.bayesbr — *Matrix with All Variables for* `bayesbr` *Objects*

Description

The function receives all variables and their respective names, and concatenates them in a matrix.

Usage

```
## S3 method for class 'bayesbr'
model(Y,X = NULL,W = NULL,name_y,names_x = NULL,
               names_w = NULL)
```

Arguments

`Y`	A vector containing the model response variable,
`X`	A matrix containing the covariates for theta of the model,
`W`	A matrix containing the covariates of the model for zeta,
`name_y`	The name passed in the call to the `bayesbr` function for the variable response,
`names_x`	The name passed in the call to the `bayesbr` function for the covariates for theta,
`names_w`	The name passed in the call to the `bayesbr` function for covariates for zeta.

Value

A matrix containing all variables in the model and their names used as column names.

See Also

`values`,`bayesbr`

model_frame *Model Matrix/Frame with All Variables for* bayesbr *Objects*

Description

The function receives all variables and their respective names, and concatenates them in a matrix.

Usage

```
model_frame(object,...)
```

Arguments

object	an object of the class *bayesbr*, containing the list returned from the bayesbr function.
...	further arguments passed to or from other methods.

Value

A matrix or Frame containing all variables in the model and their names used as column names.

See Also

values,bayesbr

Examples

```
data("bodyfat",package="bayesbr")

bbr = bayesbr(siri ~ wrist +I(age/100)|chest, data = bodyfat,
              iter = 100)
model_matrix(bbr)

model_frame(bbr)
```

model_matrix *Model Matrix/Frame with All Variables for* bayesbr *Objects*

Description

The function receives all variables and their respective names, and concatenates them in a matrix.

Usage

```
model_matrix(object,...)
```

Arguments

`object`	an object of the class *bayesbr*, containing the list returned from the `bayesbr` function.
`...`	further arguments passed to or from other methods.

Value

A matrix or Frame containing all variables in the model and their names used as column names.

See Also

`values`,`bayesbr`

Examples

```
data("bodyfat",package="bayesbr")

bbr = bayesbr(siri ~ wrist +I(age/100)|chest, data = bodyfat,
              iter = 100)
model_matrix(bbr)

model_frame(bbr)
```

---

pmse — *Prediction Mean Squared Error in a Beta Regression on a Bayesian Model*

Description

A function that selects a part of the database to fit a beta regression model and another part of this database to test the built model, returning the PMSE (prediction mean squared error) that reports the quality of the estimation for that database. In addition, the function also contains all the information that the bayesbr function returns, making it possible to do all analyzes on the fitted model.

Usage

```
pmse(formula = NULL, data = NULL, test.set = 0.3,
na.action = c("exclude", "replace"),mean_betas = NULL,
   variance_betas = NULL,mean_gammas = NULL, variance_gammas = NULL,
    iter = 10000, warmup = iter/2,chains = 1, pars = NULL,
     a = NULL, b = NULL, resid.type = c("quantile",
     "sweighted", "pearson", "ordinary"), ...)
```

Arguments

`formula`	symbolic description of the model (of type `y ~ x` or `y ~ x | z`;). See more at `formula`
`data`	data frame or list with the variables passed in the formula parameter, if `data = NULL` the function will use the existing variables in the global environment.
`test.set`	Defines the proportion of the database that will be used for testing the adjusted model and calculating the PMSE. The rest of the database will be used for modeling. Test.set must be less than "0.5", so that more than 50% of the database is used to adjust the model.
`na.action`	Characters provided or treatment used in NA values. If `na.action` is equal to exclude (default value), the row containing the NA will be excluded in all variables of the model. If `na.action` is equal to replace, the row containing the NA will be replaced by the average of the variable in all variables of the model.
`mean_betas, variance_betas`	vectors including a priori information of mean and variance for the estimated beta respectively, beta is the name given to the coefficient of each covariate that influences theta. PS: the size of the vectors must equal p + 1, p being the number of covariates for theta.
`mean_gammas, variance_gammas`	vectors including a priori information of mean and variance for the estimated ranges respectively, gamma is the name given to the coefficient of each covariate that influences zeta. PS: the size of the vectors must be equal to q + 1, q being the number of covariates for zeta.
`iter`	A positive integer specifying the number of iterations for each chain (including warmup). The default is 10000.
`warmup`	A positive integer specifying the number of iterations that will be in the warm-up period, will soon be discarded when making the estimates and inferences. Warmup must be less than `iter` and its default value is `iter/2`.
`chains`	A positive integer specifying the number of Markov chains. The default is 1.
`pars`	A vector of character strings specifying parameters of interest. The default is NULL indicating all parameters in the model.
`a, b`	Positive integer specifying the a priori information of the parameters of the gamma distribution for the zeta, if there are covariables explaining zeta `a` and `b` they will not be used.
`resid.type`	A character containing the residual type returned by the model among the possibilities. The type of residue can be *quantile*, *sweighted*, *pearson* or *ordinary*. The default is *quantile*.
`...`	Other optional parameters from RStan

Value

`pmse` return an object of class *pmse_bayesbr* containing the value of the prediction mean squared error and an object of the class *bayesbr* with the following items:

coefficients a list with the mean and precision elements containing the estimated coefficients of model and table with the means, medians, standard deviations and the Highest Posterior Density (HPD) Interval,

call the original function call,

formula the original formula,

y the response proportion vector,

stancode lines of code containing the .STAN file used to estimate the model,

info a list containing model information such as the argument pars passed as argument, name of variables, number of: iterations, warmups, chains, covariables for theta, covariables for zeta and observations of the sample. In addition there is an element called samples, with the posterior distribution of the parameters of interest,

fitted.values a vector containing the estimates for the values corresponding to the theta of each observation of the variable response, the estimate is made using the mean of the a prior theta distribution,

model the full model frame,

residuals a vector of residuals

residuals.type the type of returned residual,

loglik log-likelihood of the fitted model(using the mean of the parameters in the posterior distribution),

BIC a value containing the Bayesian Information Criterion (BIC) of the fitted model,

pseudo.r.squared pseudo-value of the square R (correlation to the square of the linear predictor and the a posteriori means of theta).

References

doi: 10.1080/0266476042000214501 Ferrari, S.L.P., and Cribari-Neto, F. (2004). Beta Regression for Modeling Rates and Proportions. *Journal of Applied Statistics*, **31**(7), 799–815.

doi: 10.1016/j.jeconom.2005.07.014 Clark, T. E., & West, K. D. (2006). Using out-of-sample mean squared prediction errors to test the martingale difference hypothesis. *Journal of econometrics*, **135**(1-2), 155-186.

See Also

`bayesbr`,`residuals.bayesbr`,`predict.bayesbr`

Examples

```
data("bodyfat",package="bayesbr")

bbr = pmse(siri ~ age + weight| biceps + forearm, data = bodyfat,
             test.set = 0.25, iter = 100)

pmse = bbr$PMSE
model = bbr$model
summary(model)
residuals(model,type="sweighted")
```

```
bbr2 = pmse(siri ~ age + weight + height +
              wrist | biceps + forearm, data = bodyfat,
              test.set = 0.4, iter = 1000,
              mean_betas = 3,variance_betas = 10)
pmse2 = bbr2$PMSE
model2 = bbr2$model
residuals(model2,type="sweighted")
```

`predict.bayesbr` *Prediction Method for* `bayesbr` *Objects*

Description

A function that informs various types of prediction through a beta regression by the Bayesian view.

Usage

```
## S3 method for class 'bayesbr'
predict(object, newdata = NULL, type = c("response", "link",
"precision", "variance", "quantile"), na.action=c("exclude",
"replace"),at = 0.5,...)
```

Arguments

`object`	an object of the class *bayesbr*, containing the list returned from the `bayesbr` function.
`newdata`	A data frame in which to look for variables with which to predict. If omitted, the original observations are used.
`type`	A character containing the types of predictions: if type is *"response"* the function will calculate fitted values for theta; if type is *"link"* the function will calculate the linear predictor for theta and zeta;if type is *"precision"* the function will calculate fitted values for zeta parameter;if type is *"variance"* the function will calculate fitted variances of response; if type is *"quantile"* the function will calculate fitted quantiles of theta values;
`na.action`	Characters provided or treatment used in NA values. If `na.action` is equal to exclude (default value), the row containing the NA will be excluded in all variables of the model. If `na.action` is equal to replace, the row containing the NA will be replaced by the average of the variable in all variables of the model.
`at`	numeric vector indicating the quantiles that will be informed by the function (only if `type = "quantile"`). Its default is 0.5.
`...`	further arguments passed to or from other methods.

See Also

`bayesbr`,`residuals.bayesbr`,`pmse`

Examples

```
data("CarTask", package = "bayesbr")

bbr = bayesbr(probability~ NFCCscale,data=CarTask,
             iter = 100, mean_betas = c(1,1.2))

predict(bbr, type = "response")
predict(bbr, type = "link")
predict(bbr, type = "precision")
predict(bbr, type = "variance")
predict(bbr, type = "quantile", at = c(0.25, 0.5, 0.75))


df = data.frame(NFCCscale = rnorm(10,4,1.4))

predict(bbr, newdata = df, type = "response")
predict(bbr, newdata = df, type = "link")
predict(bbr, newdata = df, type = "precision")
predict(bbr, newdata = df, type = "variance")
predict(bbr, newdata = df, type = "quantile", at = c(0.25, 0.5, 0.75))
```

---

print.bayesbr *Print for* bayesbr *Objects*

Description

A method that receives a list of the bayesbr type and its items and displays the estimated coefficients.

Usage

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

Arguments

x	an object of the class *bayesbr*, containing the list returned from the bayesbr function.
...	further arguments passed to or from other methods.

See Also

bayesbr, summary.bayesbr, residuals.bayesbr

Examples

```
data("bodyfat",package="bayesbr")

bbr = bayesbr(brozek ~ wrist + density:thigh |chest, data = bodyfat,
             iter = 100)
print(bbr)
```

pseudo.r.squared *Pseudo R Squared Calculate*

Description

The function receives the model information, as well as the variable response and the predicted theta values and calculates the model's pseudo.r.squared, using the formula proposed by Cribarri-Neto and Ferrari.

Usage

```
pseudo.r.squared(x)
```

Arguments

x	an object of the class *bayesbr*, containing the list returned from the bayesbr function.

Details

Ferarri and Cribari-Neto (2004) defined the pseudo.r.squared as the square of the correlation between the theta estimated by the maximum likelihood and the logis of the variable response of the model. But as we are in the context of Bayesian statistics, the estimated theta is given by the mean of the posterior distribution of the parameter. So the informed pseudo.r.squared is a Bayesian adaptation to what was suggested by Ferarri and Cribari-Neto (2004).

Value

A number containing the pseudo r squared of the adjusted model, this value can be used to assess the quality of the model.

References

doi: 10.1080/0266476042000214501 Ferrari, S., & Cribari-Neto, F. (2004). Beta regression for modelling rates and proportions. *Journal of applied statistics*, **31**(7), 799-815.

See Also

bayesbr,fitted.values,AIC_bayesbr

`ReadingSkills` *Dyslexia and IQ Predicting Reading Accuracy*

Description

Data to verify the importance of non-verbal IQ in children's reading skills in dyslexic and non-dyslexic children.

Usage

```
data("ReadingSkills")
```

Format

A data frame containing 44 observations on 3 variables.

accuracy reading score scaled to the open unit interval (see below).

dyslexia Is the child dyslexic? If 0, no; If 1, yes.

iq non-verbal intelligence quotient transformed to z-scores.

Details

The data were collected by Pammer and Kevan (2004). The original precision score was transformed by Smithson and Verkuilen (2006) so that the values of `precision` are always between 0 to 1, enabling the use of beta regression.

First, the original accuracy was scaled using the minimal and maximal score (`a` and `b`, respectively) that can be obtained in the test: `(original_accuracy -a) / (b -a)` (`a` and `b` are not provided). Subsequently, the scaled score is transformed to the unit interval using a continuity correction: `(scaled_accuracy * (n-1) -0.5) / n`(either with some rounding or using `n = 50` rather than 44).

The `dyslexia` variable that was a qualitative variable was transformed into a quantitative variable to be used by the package functions.

Source

Example 3 from Smithson and Verkuilen (2006) supplements.

References

doi: 10.18637/jss.v034.i02 Cribari-Neto, F., and Zeileis, A. (2010). Beta Regression in R. *Journal of Statistical Software*, **34**(2), 1–24. `https://www.jstatsoft.org/article/view/v034i02`.

doi: 10.18637/jss.v048.i11 Grün, B., Kosmidis, I., and Zeileis, A. (2012). Extended Beta Regression in R: Shaken, Stirred, Mixed, and Partitioned. *Journal of Statistical Software*, **48**(11), 1–25. `https://www.jstatsoft.org/article/view/v048i11`.

doi: 10.1080/10888430709336633 Pammer, K., and Kevan, A. (2004). The Contribution of Visual Sensitivity, Phonological Processing and Non-Verbal IQ to Children's Reading. *Unpublished manuscript*, The Australian National University, Canberra.

doi: 10.1037/1082989X.11.1.54 Smithson, M., and Verkuilen, J. (2006). A Better Lemon Squeezer? Maximum-Likelihood Regression with Beta-Distributed Dependent Variables. *Psychological Methods*, **11**(7), 54–71.

Examples

```
data("ReadingSkills", package = "bayesbr")

bbr = bayesbr(accuracy~iq+dyslexia, iter=1000,warmup=300,
             data=ReadingSkills)
summary(bbr)
```

`residuals.bayesbr` *Residuals for* `bayesbr` *Objects*

Description

A function that receives model information and calculates the residuals according to the required residual.

Usage

```
## S3 method for class 'bayesbr'
residuals(object, type = c("", "quantile", "sweighted", "pearson","ordinary"),...)
```

Arguments

`object`	an object of the class *bayesbr*, containing the list returned from the `bayesbr` function.
`type`	A character containing the residual type returned by the model among the possibilities. The type of residue can be *quantile*, *sweighted*, *pearson* or *ordinary*. The default is *quantile*.
`...`	further arguments passed to or from other methods.

Details

The definitions of the waste generated by the package are available in Espinheira (2008): "pearson" in Equation 2, "sweighted" in Equation 7; and in Pereira (2019): "quantile" in Equation 5;

The type of residue "response" is calculated from the difference between the estimated theta and the variable response of the model.

Value

A vector containing the model residual according to the type of residual calculated

References

doi: 10.1080/0266476042000214501 Ferrari, S., & Cribari-Neto, F. (2004). Beta regression for modelling rates and proportions. *Journal of applied statistics*, **31**(7), 799-815.

doi: 10.1080/00949650701829380 Simas, A. B., & Cordeiro, G. M. (2009). Adjusted Pearson residuals in exponential family nonlinear models. *Journal of Statistical Computation and Simulation*, **79**(4), 411-425.

doi: 10.1080/02664760701834931 Espinheira, P. L., Ferrari, S. L., & Cribari-Neto, F. (2008). On beta regression residuals. *Journal of Applied Statistics*, **35**(4), 407-419.

doi: 10.1080/00949655.2012.736993 Anholeto, T., Sandoval, M. C., & Botter, D. A. (2014). Adjusted Pearson residuals in beta regression models. *Journal of Statistical Computation and Simulation*, **84**(5), 999-1014.

doi: 10.1080/03610918.2017.1381740 Pereira, G. H. (2019). On quantile residuals in beta regression. *Communications in Statistics-Simulation and Computation*, **48**(1), 302-316.

See Also

`bayesbr`,`summary.bayesbr`,`predict.bayesbr`

Examples

```
data("CarTask", package = "bayesbr")

bbr = bayesbr(probability~task + NFCCscale,data=CarTask,
              iter = 100, mean_betas = c(1, 0.5,1.2))

residuals(bbr, type = "quantile")
residuals(bbr, type = "ordinary")
residuals(bbr, type = "sweighted")
residuals(bbr, type = "pearson")
```

`StressAnxiety` *Dependency of Anxiety on Stress*

Description

For this data, stress and anxiety were measured among nonclinical women in Townsville, Queensland, Australia.

Usage

```
data("StressAnxiety")
```

Format

A data frame containing 166 observations on 2 variables.

stress score, linearly transformed to the open unit interval (see below).

anxiety score, linearly transformed to the open unit interval (see below).

Details

Both variables were evaluated on the scales from 0 to 42, Smithson and Verkuilen (2006) transformed them in a range from 0 to 1.

Source

Example 2 from Smithson and Verkuilen (2006) supplements.

References

doi: 10.1037/1082989X.11.1.54 Smithson, M., and Verkuilen, J. (2006). A Better Lemon Squeezer? Maximum-Likelihood Regression with Beta-Distributed Dependent Variables. *Psychological Methods*, **11**(7), 54–71.

Examples

```
data("StressAnxiety", package = "bayesbr")
bbr <- bayesbr(anxiety ~ stress | stress,
              data = StressAnxiety, iter = 100)

summary(bbr)
```

---

summary.bayesbr — *Summary for* bayesbr *Objects*

Description

A method that receives a list of the bayesbr type and its items and displays the main information of the model, such as the residuals, a table containing statistics on the estimated coefficients and information to evaluate the quality of the model.

Usage

```
## S3 method for class 'bayesbr'
summary(object,type = c("","quantile","sweighted",
"pearson","ordinary"), prob = 0.95,...)
```

Arguments

`object`	an object of the class *bayesbr*, containing the list returned from the `bayesbr` function.
`type`	A character containing the residual type returned by the model among the possibilities. The type of residue can be *quantile*, *sweighted*, *pearson* or *ordinary*. The default is *quantile*.
`prob`	a probability containing the credibility index for the HPD interval for the coefficients of the covariates.
`...`	further arguments passed to or from other methods.

See Also

bayesbr,residuals.bayesbr,print.bayesbr,predict.bayesbr

Examples

```
data("bodyfat",package="bayesbr")

bbr = bayesbr(siri ~ age + weight +
              wrist | biceps + forearm,
              data = bodyfat, iter = 100)

summary(bbr)
summary(bbr, type="pearson")
summary(bbr, prob = 0.9)
summary(bbr, prob = 0.99, resid.type="sweighted")

bbr2 = bayesbr(siri ~ age + weight + height +
           wrist | biceps + forearm, data = bodyfat,
           iter = 100,mean_betas = 3,
           variance_betas = 10)

summary(bbr2)
summary(bbr2, type="sweighted")
summary(bbr2, prob = 0.96)
summary(bbr2, prob = 0.95, resid.type="quantile")
```

---

summary_delta — *Coefficients for deltas*

Description

A function that uses posterior distribution values of the model and calculates the estimates for delta parametrer.

Usage

```
summary_delta(x,prob=0.95)
```

Arguments

x	an object of the class *bayesbr*, containing the list returned from the bayesbr function.
prob	a probability containing the credibility index for the HPD interval for the coefficients of the covariates.

Value

A list containing the estimates for delta parametrer, this list contains the following items:

table a table with the means, medians, standard deviations and the Highest Posterior Density (HPD) Interval,

coeff a vector containing the estimated coefficients.

See Also

`summary_xi`,`values`,`summary.bayesbr`

---

summary_mean *Variable Coefficients for Theta*

Description

A function that uses the beta values of the posterior distribution of the model and calculates the estimates for each theta covariate.

Usage

```
summary_mean(x,prob=0.95)
```

Arguments

x	an object of the class *bayesbr*, containing the list returned from the `bayesbr` function.
prob	a probability containing the credibility index for the HPD interval for the coefficients of the covariates.

Value

A list containing the estimates for the covariables of theta, this list contains the following items:

table a table with the means, medians, standard deviations and the Highest Posterior Density (HPD) Interval,

coeff a vector containing the estimated coefficients for the variables.

See Also

`summary_precision`,`values`,`summary.bayesbr`

`summary_precision` *Variable Coefficients for Zeta*

Description

A function that uses the gamma values of the posterior distribution of the model and calculates the estimates for each zeta covariate.

Usage

```
summary_precision(x,prob=0.95)
```

Arguments

`x`	an object of the class *bayesbr*, containing the list returned from the `bayesbr` function.
`prob`	a probability containing the credibility index for the HPD interval for the coefficients of the covariates.

Value

A list containing the estimates for the covariables of zeta, this list contains the following items:

table a table with the means, medians, standard deviations and the Highest Posterior Density (HPD) Interval,

coeff a vector containing the estimated coefficients for the variables.

See Also

`summary_mean`,`values`,`summary.bayesbr`

`summary_tau_delta` *Coefficients for tau_delta*

Description

A function that uses values of the posterior distribution of the model and calculates the estimates for tau_delta parametrer.

Usage

```
summary_tau_delta(x,prob=0.95)
```

Arguments

`x`	an object of the class *bayesbr*, containing the list returned from the `bayesbr` function.
`prob`	a probability containing the credibility index for the HPD interval for the coefficients of the covariates.

Value

A list containing the estimates for tau parametrer, this list contains the following items:

table a table with the means, medians, standard deviations and the Highest Posterior Density (HPD) Interval,

coeff a vector containing the estimated coefficients.

See Also

`summary_delta`,`values`,`summary_tau_xi`

`summary_tau_xi` *Coefficients for tau_xi*

Description

A function that uses values of the posterior distribution of the model and calculates the estimates for tau_xi parametrer.

Usage

```
summary_tau_xi(x,prob=0.95)
```

Arguments

`x`	an object of the class *bayesbr*, containing the list returned from the `bayesbr` function.
`prob`	a probability containing the credibility index for the HPD interval for the coefficients of the covariates.

Value

A list containing the estimates for tau parametrer, this list contains the following items:

table a table with the means, medians, standard deviations and the Highest Posterior Density (HPD) Interval,

coeff a vector containing the estimated coefficients.

See Also

`summary_xi`,`values`,`summary_tau_delta`

summary_xi	*Coefficients for xis*

Description

A function that uses posterior distribution values of the model and calculates the estimates for xi parametrer.

Usage

```
summary_xi(x,prob=0.95)
```

Arguments

x	an object of the class *bayesbr*, containing the list returned from the `bayesbr` function.
prob	a probability containing the credibility index for the HPD interval for the coefficients of the covariates.

Value

A list containing the estimates for xi parametrer, this list contains the following items:

table a table with the means, medians, standard deviations and the Highest Posterior Density (HPD) Interval,

coeff a vector containing the estimated coefficients.

See Also

`summary_delta`,`values`,`summary.bayesbr`

values	*Values of a Posteriori Distribution*

Description

A function that uses the values returned from the sampling function of RStan and returns the parameter chain of the posterior distribution, the parameters can be beta, gamma, theta or zeta.

Usage

```
values(
 type = c("beta", "gamma", "theta", "zeta", "tau_delta", "tau_xi", "xi", "delta"),
  obj,
  iter,
  warmup,
  n,
  par
)
```

Arguments

`type`	Characters indicating which values will be returned by the function
`obj`	containing the data returned from the sampling function of the Rstan package. This type of object is used because it returns the values of the posterior distribution of the model.
`iter`	A positive integer specifying the number of iterations for each chain (including warmup).
`warmup`	A positive integer specifying the number of iterations that will be in the warm-up period.
`n`	The number of observations of the model's variable response.
`par`	A number containing the number of parameters for theta or zeta. If type is equal to beta or theta par is similar to p (number of parameters for theta), otherwise even is similar to q (number of parameters for zeta). When type is equal to 'delta', 'xi', 'tau_xi', or 'tau_delta' the par variable verify spatial effect in adjusted model.

Details

The function `values` returns the parameter of interest by taking the data returned by the Stan function excluding the warmup period data. All data returned is in the format of 5 decimal places.

Value

A list containing the values according to the type argument, the values are returned excluding the warmups.

See Also

`summary_mean`,`summary_precision`,`model.bayesbr`

`WeatherTask` *Precise and Imprecise Probabilities and Priming for Weather Task*

Description

In this experiment, participants judged the likelihood of Sunday being the hottest day of week

Usage

```
data(WeatherTask)
```

Format

A data frame with 345 observations on the following 3 variables.

`priming` a variable. If 0, `two-fold` (case prime); If 1, `seven-fold` (class prime).

`eliciting` a variable. If 0, `precise`;If 1, `imprecise` (lower and upper limit).

`agreement` a numeric vector, probability indicated by participants or the average between minimum and maximum probability indicated.

Details

All study participants were from the first or second year, none of the participants had an in-depth knowledge of probability.

For `priming` the questions were:

two-fold [What is the probability that] the temperature at Canberra airport on Sunday will be higher than every other day next week?

seven-fold [What is the probability that] the highest temperature of the week at Canberra airport will occur on Sunday?

For `eliciting` the instructions were if

precise to assign a probability estimate,

imprecise to assign a lower and upper probability estimate.

The `priming` and `eliciting` variables that was a qualitative variable was transformed into a quantitative variable to be used by the package functions.

Source

Taken from Smithson et al. (2011) supplements.

References

doi: 10.3102/1076998610396893 Smithson, M., Merkle, E.C., and Verkuilen, J. (2011). Beta Regression Finite Mixture Models of Polarization and Priming. *Journal of Educational and Behavioral Statistics*, **36**(6), 804–831.

doi: 10.3102/1076998610396893 Smithson, M., and Segale, C. (2009). Partition Priming in Judgments of Imprecise Probabilities. *Journal of Statistical Theory and Practice*, **3**(1), 169–181.

Examples

```
data("WeatherTask", package = "bayesbr")

bbr <- bayesbr(agreement~eliciting+priming, data = WeatherTask,
              iter = 200)
```

Index