Package 'bqror'

December 14, 2024

Type Package

Title Bayesian Quantile Regression for Ordinal Models

Version 1.7.1

URL https://github.com/prajual/bqror

Imports MASS, pracma, GIGrvg, truncnorm, NPflow, invgamma, stats, progress

Maintainer Prajual Maheshwari <prajual1391@gmail.com>

Description

Package provides functions for estimation and inference in Bayesian quantile regression with ordinal outcomes. An ordinal model with 3 or more outcomes (labeled OR1 model) is estimated by a combination of Gibbs sampling and Metropolis-Hastings (MH) algorithm. Whereas an ordinal model with exactly 3 outcomes (labeled OR2 model) is estimated using a Gibbs sampling algorithm. The summary output presents the posterior mean, posterior standard deviation, 95% credible intervals, and the inefficiency factors along with the two model comparison measures – logarithm of marginal likelihood and the deviance information criterion (DIC). The package also provides functions for computing the covariate effects and other functions that aids either the estimation or inference in quantile ordinal models.
Rahman, M. A. (2016).“Bayesian Quantile Regression for Ordinal Models.” Bayesian Analysis, 11(1): 1-24 <doi:10.1214/15-BA939>.
Yu, K., and Moyeed, R. A. (2001). “Bayesian Quantile Regression.” Statistics and Probability Letters, 54(4): 437–447 <doi:10.1016/S0167-7152(01)00124-9>.
Koenker, R., and Bassett, G. (1978).“Regression Quantiles.” Econometrica, 46(1): 33-50 <doi:10.2307/1913643>.
Chib, S. (1995). “Marginal likelihood from the Gibbs output.” Journal of the American Statistical Association, 90(432):1313–1321, 1995. <doi:10.1080/01621459.1995.10476635>.
Chib, S., and Jeliazkov, I. (2001). “Marginal likelihood from the Metropolis-Hastings output.” Journal of the American Statistical Association, 96(453):270–281, 2001. <doi:10.1198/016214501750332848>.

License GPL (>= 2)

Encoding UTF-8

LazyData true

Repository CRAN

RoxygenNote 7.3.1

Depends R (>= 3.5.0)

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

NeedsCompilation no

Author Mohammad Arshad Rahman Developer [aut],
Prajual Maheshwari [cre]

Date/Publication 2024-12-14 07:50:06 UTC

Contents

alcdf *cdf of an asymmetric Laplace distribution*

Description

This function computes the cumulative distribution function (cdf) of an asymmetric Laplace (AL) distribution.

Usage

```
alcdf(x, mu, sigma, p)
```

Arguments

`x`	scalar value.
`mu`	location parameter of an AL distribution.
`sigma`	scale parameter of an AL distribution.
`p`	quantile or skewness parameter, p in (0,1).

Details

Computes the cdf of an AL distribution.

$$CDF(x) = F(x) = P(X \leq x)$$

where X is a random variable that follows AL(μ, σ, p)

Value

Returns the cumulative probability value at point ‘"x"‘.

References

Rahman, M. A. (2016). ‘"Bayesian Quantile Regression for Ordinal Models."‘ Bayesian Analysis, 11(1): 1-24. DOI: 10.1214/15-BA939

Yu, K., and Zhang, J. (2005). ‘"A Three-Parameter Asymmetric Laplace Distribution."‘ Communications in Statistics - Theory and Methods, 34(9-10), 1867-1879. DOI: 10.1080/03610920500199018

See Also

cumulative distribution function, asymmetric Laplace distribution

Examples

```
set.seed(101)
x <- -0.5428573
mu <- 0.5
sigma <- 1
p <- 0.25
output <- alcdf(x, mu, sigma, p)

# output
#   0.1143562
```

`alcdfstd` *cdf of a standard asymmetric Laplace distribution*

Description

This function computes the cdf of a standard AL distribution i.e. AL$(0, 1, p)$.

Usage

```
alcdfstd(x, p)
```

Arguments

`x`	scalar value.
`p`	quantile level or skewness parameter, p in (0,1).

Details

Computes the cdf of a standard AL distribution.

$$cdf(x) = F(x) = P(X \leq x)$$

where X is a random variable that follows AL$(0, 1, p)$.

Value

Returns the cumulative probability value at point x for a standard AL distribution.

References

Rahman, M. A. (2016). ‘"Bayesian Quantile Regression for Ordinal Models."‘ Bayesian Analysis, 11(1): 1-24. DOI: 10.1214/15-BA939

Yu, K., and Zhang, J. (2005). ‘"A Three-Parameter Asymmetric Laplace Distribution."‘ Communications in Statistics - Theory and Methods, 34(9-10), 1867-1879. DOI: 10.1080/03610920500199018

See Also

asymmetric Laplace distribution

Examples

```
set.seed(101)
x <-  -0.5428573
p <- 0.25
output <- alcdfstd(x, p)

# output
#   0.1663873
```

bqror *Bayesian quantile regression for ordinal models*

Description

Package provides functions for estimation and inference in Bayesian quantile regression with ordinal outcomes. An ordinal model with 3 or more outcomes (labeled OR1 model) is estimated by a combination of Gibbs sampling and Metropolis-Hastings (MH) algorithm. Whereas an ordinal model with exactly 3 outcomes (labeled OR2 model) is estimated using a Gibbs sampling algorithm. The summary output presents the posterior mean, posterior standard deviation, 95% credible intervals, and the inefficiency factors along with the two model comparison measures logarithm of marginal likelihood and the deviance information criterion (DIC). The package also provides functions for computing the covariate effects and other functions that aids either the estimation or inference in quantile ordinal models.

Details

$$Package : bqror$$

$$Type : Package$$

$$Version : 1.7.0$$

$$License : GPL(>= 2)$$

Package **bqror** provides the following functions:

- For an ordinal model with three or more outcomes:

quantregOR1, covEffectOR1, logMargLikeOR1, dicOR1, qrnegLogLikensumOR1, ineffactorOR1, qrminfundtheorem, drawbetaOR1, drawwOR1, drawlatentOR1, drawdeltaOR1, alcdfstd, alcdf

- For an ordinal model with three outcomes:

quantregOR2, covEffectOR2, logMargLikeOR2, dicOR2, qrnegLogLikeOR2, ineffactorOR2, drawlatentOR2, drawbetaOR2, drawsigmaOR2, drawnuOR2, rndald

- Extractor Functions:

summary.bqrorOR1, summary.bqrorOR2

Author(s)

Mohammad Arshad Rahman

Prajual Maheshwari <prajual1391@gmail.com>

References

Rahman, M. A. (2016). ‘"Bayesian Quantile Regression for Ordinal Models."‘ Bayesian Analysis, 11(1): 1-24. DOI: 10.1214/15-BA939

Yu, K., and Moyeed, R. A. (2001). ‘"Bayesian Quantile Regression."‘ Statistics and Probability Letters, 54(4): 437 - 447. DOI:10.1016/S0167-7152(01)00124-9

Koenker, R., and Bassett, G. (1978).‘"Regression Quantiles."‘ Econometrica, 46(1): 33-50. DOI: 10.2307/1913643

Greenberg, E. (2012). ‘"Introduction to Bayesian Econometrics."‘ Cambridge University Press. Cambridge, DOI: 10.1017/CBO9781139058414

See Also

rgig, mvrnorm, ginv, rtruncnorm, mvnpdf, rinvgamma, mldivide, rand, qnorm, rexp, rnorm, std, sd, acf, Reshape, progress_bar, dinvgamma, logLik

covEffectOR1 *Covariate effect in the OR1 model*

Description

This function computes the average covariate effect for different outcomes of the OR1 model at a specified quantile. The covariate effects are calculated marginally of the parameters and the remaining covariates.

Usage

```
covEffectOR1(modelOR1, y, xMat1, xMat2, p, verbose)
```

Arguments

`modelOR1` output from the quantregOR1 function.

`y` observed ordinal outcomes, column vector of size $(nx1)$.

`xMat1` covariate matrix of size (nxk) including a column of ones with or without column names. If the covariate of interest is continuous, then the column for the covariate of interest remains unchanged (xMat1 = x). If it is an indicator variable then replace the column for the covariate of interest with a column of zeros.

`xMat2` covariate matrix x with suitable modification to an independent variable including a column of ones with or without column names. If the covariate of interest is continuous, then add the incremental change to each observation in the column for the covariate of interest. If the covariate is an indicator variable, then replace the column for the covariate of interest with a column of ones.

`p` quantile level or skewness parameter, p in (0,1).

`verbose` whether to print the final output and provide additional information or not, default is TRUE.

Details

This function computes the average covariate effect for different outcomes of the OR1 model at a specified quantile. The covariate effects are computed, using the MCMC draws, marginally of the parameters and the remaining covariates.

Value

Returns a list with components:

`avgDiffProb:` vector with change in predicted probability for each outcome category.

References

Rahman, M. A. (2016). ‘"Bayesian Quantile Regression for Ordinal Models."‘ Bayesian Analysis, 11(1): 1-24. DOI: 10.1214/15-BA939

Jeliazkov, I., Graves, J., and Kutzbach, M. (2008). ‘"Fitting and Comparison of Models for Multivariate Ordinal Outcomes."‘ Advances in Econometrics: Bayesian Econometrics, 23: 115‘-‘156. DOI: 10.1016/S0731-9053(08)23004-5

Jeliazkov, I. and Rahman, M. A. (2012). ‘"Binary and Ordinal Data Analysis in Economics: Modeling and Estimation"‘ in Mathematical Modeling with Multidisciplinary Applications, edited by X.S. Yang, 123-150. John Wiley ‘&‘ Sons Inc, Hoboken, New Jersey. DOI: 10.1002/9781118462706.ch6

Examples

```
set.seed(101)
data("data25j4")
y <- data25j4$y
xMat1 <- data25j4$x
k <- dim(xMat1)[2]
J <- dim(as.array(unique(y)))[1]
```

```
b0 <- array(rep(0, k), dim = c(k, 1))
B0 <- 10*diag(k)
d0 <- array(0, dim = c(J-2, 1))
D0 <- 0.25*diag(J - 2)
modelOR1 <- quantregOR1(y = y, x = xMat1, b0, B0, d0, D0,
burn = 10, mcmc = 40, p = 0.25, tune = 1, accutoff = 0.5, maxlags = 400, verbose = FALSE)
xMat2 <- xMat1
xMat2[,3] <- xMat2[,3] + 0.02
res <- covEffectOR1(modelOR1, y, xMat1, xMat2, p = 0.25, verbose = TRUE)

# Summary of Covariate Effect:

#               Covariate Effect
# Category_1         -0.0072
# Category_2         -0.0012
# Category_3         -0.0009
# Category_4          0.0093
```

covEffectOR2 *Covariate effect in the OR2 model*

Description

This function computes the average covariate effect for different outcomes of the OR2 model at a specified quantile. The covariate effects are calculated marginally of the parameters and the remaining covariates.

Usage

```
covEffectOR2(modelOR2, y, xMat1, xMat2, gammacp2, p, verbose)
```

Arguments

modelOR2	output from the quantregOR2 function.
y	observed ordinal outcomes, column vector of size $(nx1)$.
xMat1	covariate matrix of size (nxk) including a column of ones with or without column names. If the covariate of interest is continuous, then the column for the covariate of interest remains unchanged. If it is an indicator variable then replace the column for the covariate of interest with a column of zeros.
xMat2	covariate matrix x with suitable modification to an independent variable including a column of ones with or without column names. If the covariate of interest is continuous, then add the incremental change to each observation in the column for the covariate of interest. If the covariate is an indicator variable, then replace the column for the covariate of interest with a column of ones.
gammacp2	one and only cut-point other than 0.
p	quantile level or skewness parameter, p in (0,1).
verbose	whether to print the final output and provide additional information or not, default is TRUE.

Details

This function computes the average covariate effect for different outcomes of the OR2 model at a specified quantile. The covariate effects are computed, using the Gibbs draws, marginally of the parameters and the remaining covariates.

Value

Returns a list with components:

`avgDiffProb:` vector with change in predicted probability for each outcome category.

References

Rahman, M. A. (2016). ‘"Bayesian Quantile Regression for Ordinal Models."‘ Bayesian Analysis, 11(1): 1-24. DOI: 10.1214/15-BA939

Jeliazkov, I., Graves, J., and Kutzbach, M. (2008). ‘"Fitting and Comparison of Models for Multivariate Ordinal Outcomes."‘ Advances in Econometrics: Bayesian Econometrics, 23: 115‘-‘156. DOI: 10.1016/S0731-9053(08)23004-5

Jeliazkov, I., and Rahman, M. A. (2012). ‘"Binary and Ordinal Data Analysis in Economics: Modeling and Estimation"‘ in Mathematical Modeling with Multidisciplinary Applications, edited by X.S. Yang, 123-150. John Wiley ‘&‘ Sons Inc, Hoboken, New Jersey. DOI: 10.1002/9781118462706.ch6

Examples

```
set.seed(101)
data("data25j3")
y <- data25j3$y
xMat1 <- data25j3$x
k <- dim(xMat1)[2]
b0 <- array(rep(0, k), dim = c(k, 1))
B0 <- 10*diag(k)
n0 <- 5
d0 <- 8
output <- quantregOR2(y, xMat1, b0, B0, n0, d0, gammacp2 = 3,
burn = 10, mcmc = 40, p = 0.25, accutoff = 0.5, maxlags = 400, verbose = FALSE)
xMat2 <- xMat1
xMat2[,3] <- xMat2[,3] + 0.02
res <- covEffectOR2(output, y, xMat1, xMat2, gammacp2 = 3, p = 0.25, verbose = TRUE)

# Summary of Covariate Effect:

#                Covariate Effect
# Category_1          -0.0073
# Category_2          -0.0030
# Category_3           0.0103
```

data25j3 *Simulated data from the OR2 model for $p = 0.25$ (i.e., 25th quantile)*

Description

Simulated data from the OR2 model for $p = 0.25$ (i.e., 25th quantile)

Usage

```
data(data25j3)
```

Details

This data contains 500 observations generated from a quantile ordinal model with 3 outcomes at the 25th quantile (i.e., $p = 0.25$). The model specifics for generating the data are as follows: $\beta = (-4, 6, 5)$, X ~ Unif(0, 1), and ϵ ~ AL$(0, \sigma = 1, p = 0.25)$. The cut-points $(0, 3)$ are used to classify the continuous values of the dependent variable into 3 categories, which form the ordinal outcomes.

Value

Returns a list with components

`x:`	a matrix of covariates, including a column of ones.
`y:`	a column vector of ordinal outcomes.

References

Rahman, M. A. (2016). ‘"Bayesian Quantile Regression for Ordinal Models."‘ Bayesian Analysis, 11(1): 1-24. DOI: 10.1214/15-BA939

Yu, K., and Zhang, J. (2005). ‘"A Three-Parameter Asymmetric Laplace Distribution."‘ Communications in Statistics - Theory and Methods, 34(9-10), 1867-1879. DOI: 10.1080/03610920500199018

See Also

mvrnorm, Asymmetric Laplace Distribution

`data25j4` *Simulated data from the OR1 model for $p = 0.25$ (i.e., 25th quantile)*

Description

Simulated data from the OR1 model for $p = 0.25$ (i.e., 25th quantile)

Usage

```
data(data25j4)
```

Details

This data contains 500 observations generated from a quantile ordinal model with 4 outcomes at the 25th quantile (i.e., $p = 0.25$). The model specifics for generating the data are as follows: $\beta = (-4, 5, 6)$, X ~ Unif(0, 1), and ϵ ~ AL$(0, \sigma = 1, p = 0.25)$. The cut-points $(0, 2, 4)$ are used to classify the continuous values of the dependent variable into 4 categories, which form the ordinal outcomes.

Value

Returns a list with components

`x:`	a matrix of covariates, including a column of ones.
`y:`	a column vector of ordinal outcomes.

References

Rahman, M. A. (2016). ‘"Bayesian Quantile Regression for Ordinal Models."‘ Bayesian Analysis, 11(1): 1-24. DOI: 10.1214/15-BA939

Yu, K., and Zhang, J. (2005). ‘"A Three-Parameter Asymmetric Laplace Distribution."‘ Communications in Statistics - Theory and Methods, 34(9-10), 1867-1879. DOI: 10.1080/03610920500199018

See Also

mvrnorm, Asymmetric Laplace Distribution

data50j3 *Simulated data from the OR2 model for $p = 0.5$ (i.e., 50th quantile)*

Description

Simulated data from the OR2 model for $p = 0.5$ (i.e., 50th quantile)

Usage

```
data(data50j3)
```

Details

This data contains 500 observations generated from a quantile ordinal model with 3 outcomes at the 50th quantile (i.e., $p = 0.5$). The model specifics for generating the data are as follows: $\beta = (-4, 6, 5)$, X ~ Unif(0, 1), and ϵ ~ AL($0, \sigma = 1, p = 0.5$). The cut-points $(0, 3)$ are used to classify the continuous values of the dependent variable into 3 categories, which form the ordinal outcomes.

Value

Returns a list with components

`x:`	a matrix of covariates, including a column of ones.
`y:`	a column vector of ordinal outcomes.

References

Rahman, M. A. (2016). ‘"Bayesian Quantile Regression for Ordinal Models."‘ Bayesian Analysis, 11(1): 1-24. DOI: 10.1214/15-BA939

Yu, K., and Zhang, J. (2005). ‘"A Three-Parameter Asymmetric Laplace Distribution."‘ Communications in Statistics - Theory and Methods, 34(9-10), 1867-1879. DOI: 10.1080/03610920500199018

See Also

mvrnorm, Asymmetric Laplace Distribution

data50j4 *Simulated data from the OR1 model for $p = 0.5$ (i.e., 50th quantile)*

Description

Simulated data from the OR1 model for $p = 0.5$ (i.e., 50th quantile)

Usage

```
data(data50j4)
```

Details

This data contains 500 observations generated from a quantile ordinal model with 4 outcomes at the 50th quantile (i.e., $p = 0.5$). The model specifics for generating the data are as follows: $\beta = (-4, 5, 6)$, X ~ Unif(0, 1), and ϵ ~ AL$(0, \sigma = 1, p = 0.5)$. The cut-points $(0, 2, 4)$ are used to classify the continuous values of the dependent variable into 4 categories, which form the ordinal outcomes.

Value

Returns a list with components

`x:`	a matrix of covariates, including a column of ones.
`y:`	a column vector of ordinal outcomes.

References

Rahman, M. A. (2016). ‘"Bayesian Quantile Regression for Ordinal Models."‘ Bayesian Analysis, 11(1): 1-24. DOI: 10.1214/15-BA939

Yu, K., and Zhang, J. (2005). ‘"A Three-Parameter Asymmetric Laplace Distribution."‘ Communications in Statistics - Theory and Methods, 34(9-10), 1867-1879. DOI: 10.1080/03610920500199018

See Also

mvrnorm, Asymmetric Laplace Distribution

data75j3 *Simulated data from the OR2 model for $p = 0.75$ (i.e., 75th quantile)*

Description

Simulated data from the OR2 model for $p = 0.75$ (i.e., 75th quantile)

Usage

```
data(data75j3)
```

Details

This data contains 500 observations generated from a quantile ordinal model with 3 outcomes at the 75th quantile (i.e., $p = 0.75$). The model specifics for generating the data are as follows: $\beta = (-4, 6, 5)$, X ~ Unif(0, 1), and ϵ ~ AL$(0, \sigma = 1, p = 0.75)$. The cut-points $(0, 3)$ are used to classify the continuous values of the dependent variable into 3 categories, which form the ordinal outcomes.

Value

Returns a list with components

`x:`	a matrix of covariates, including a column of ones.
`y:`	a column vector of ordinal outcomes.

References

Rahman, M. A. (2016). ‘"Bayesian Quantile Regression for Ordinal Models."‘ Bayesian Analysis, 11(1): 1-24. DOI: 10.1214/15-BA939

Yu, K., and Zhang, J. (2005). ‘"A Three-Parameter Asymmetric Laplace Distribution."‘ Communications in Statistics - Theory and Methods, 34(9-10), 1867-1879. DOI: 10.1080/03610920500199018

See Also

mvrnorm, Asymmetric Laplace Distribution

`data75j4` *Simulated data from the OR1 model for $p = 0.75$ (i.e., 75th quantile)*

Description

Simulated data from the OR1 model for $p = 0.75$ (i.e., 75th quantile)

Usage

```
data(data75j4)
```

Details

This data contains 500 observations generated from a quantile ordinal model with 4 outcomes at the 75th quantile (i.e., $p = 0.75$). The model specifics for generating the data are as follows: $\beta = (-4, 5, 6)$, X ~ Unif(0, 1), and ϵ ~ AL$(0, \sigma = 1, p = 0.75)$. The cut-points $(0, 2, 4)$ are used to classify the continuous values of the dependent variable into 4 categories, which form the ordinal outcomes.

Value

Returns a list with components

`x:`	a matrix of covariates, including a column of ones.
`y:`	a column vector of ordinal outcomes.

References

Rahman, M. A. (2016). ‘"Bayesian Quantile Regression for Ordinal Models."‘ Bayesian Analysis, 11(1): 1-24. DOI: 10.1214/15-BA939

Yu, K., and Zhang, J. (2005). ‘"A Three-Parameter Asymmetric Laplace Distribution."‘ Communications in Statistics - Theory and Methods, 34(9-10), 1867-1879. DOI: 10.1080/03610920500199018

See Also

mvrnorm, Asymmetric Laplace Distribution

dicOR1 *Deviance Information Criterion in the OR1 model*

Description

Function for computing the Deviance Information Criterion (DIC) in the OR1 model (ordinal quantile model with 3 or more outcomes).

Usage

```
dicOR1(y, x, betadraws, deltadraws, postMeanbeta, postMeandelta, burn, mcmc, p)
```

Arguments

`y` observed ordinal outcomes, column vector of size $(nx1)$.

`x` covariate matrix of size (nxk) including a column of ones with or without column names.

`betadraws` dataframe of the MCMC draws of β, size is $(kxnsim)$.

`deltadraws` dataframe of the MCMC draws of δ, size is $((J-2)xnsim)$.

`postMeanbeta` posterior mean of the MCMC draws of β.

`postMeandelta` posterior mean of the MCMC draws of δ.

`burn` number of burn-in MCMC iterations.

`mcmc` number of MCMC iterations, post burn-in.

`p` quantile level or skewness parameter, p in (0,1).

Details

Deviance is -2*(log likelihood) and has an important role in statistical model comparison because of its relation with Kullback-Leibler information criterion.

This function provides the DIC, which can be used to compare two or more models at the same quantile. The model with a lower DIC provides a better fit.

Value

Returns a list with components

$$DIC = 2 * avgdDeviance - dev$$

$$pd = avgdDeviance - dev$$

$$dev = -2 * (logLikelihood)$$

.

References

Spiegelhalter, D. J., Best, N. G., Carlin, B. P. and Linde, A. (2002). ‘"Bayesian Measures of Model Complexity and Fit."‘ Journal of the Royal Statistical Society B, Part 4: 583-639. DOI: 10.1111/1467-9868.00353

Gelman, A., Carlin, J. B., Stern, H. S., and Rubin, D. B. ‘"Bayesian Data Analysis."‘ 2nd Edition, Chapman and Hall. DOI: 10.1002/sim.1856

See Also

decision criteria

Examples

```
set.seed(101)
data("data25j4")
y <- data25j4$y
xMat <- data25j4$x
k <- dim(xMat)[2]
J <- dim(as.array(unique(y)))[1]
b0 <- array(rep(0, k), dim = c(k, 1))
B0 <- 10*diag(k)
d0 <- array(0, dim = c(J-2, 1))
D0 <- 0.25*diag(J - 2)
output <- quantregOR1(y = y, x = xMat, b0, B0, d0, D0,
burn = 10, mcmc = 40, p = 0.25, tune = 1, accutoff = 0.5, maxlags = 400, verbose = FALSE)
mcmc <- 40
deltadraws <- output$deltadraws
betadraws <- output$betadraws
burn <- 0.25*mcmc
nsim <- burn + mcmc
postMeanbeta <- output$postMeanbeta
postMeandelta <- output$postMeandelta
dic <- dicOR1(y, xMat, betadraws, deltadraws,
postMeanbeta, postMeandelta, burn, mcmc, p = 0.25)

# DIC
#   1375.329
# pd
#   139.1751
# dev
#   1096.979
```

dicOR2 *Deviance Information Criterion in the OR2 model*

Description

Function for computing the DIC in the OR2 model (ordinal quantile model with exactly 3 outcomes).

Usage

```
dicOR2(y, x, betadraws, sigmadraws, gammacp, postMeanbeta,
postMeansigma, burn, mcmc, p)
```

Arguments

y — observed ordinal outcomes, column vector of size $(nx1)$.

x — covariate matrix of size (nxk) including a column of ones with or without column names.

betadraws — dataframe of the MCMC draws of β, size is $(kxnsim)$.

sigmadraws — dataframe of the MCMC draws of σ, size is $(nsimx1)$.

gammacp — row vector of cut-points including -Inf and Inf.

postMeanbeta — posterior mean of the MCMC draws of β.

postMeansigma — posterior mean of the MCMC draws of σ.

burn — number of burn-in MCMC iterations.

mcmc — number of MCMC iterations, post burn-in.

p — quantile level or skewness parameter, p in (0,1).

Details

Deviance is -2*(log likelihood) and has an important role in statistical model comparison because of its relation with Kullback-Leibler information criterion.

This function provides the DIC, which can be used to compare two or more models at the same quantile. The model with a lower DIC provides a better fit.

Value

Returns a list with components

$$DIC = 2 * avgdeviance - dev$$

$$pd = avgdeviance - dev$$

$$dev = -2 * (logLikelihood)$$

.

References

Spiegelhalter, D. J., Best, N. G., Carlin, B. P. and Linde, A. (2002). ‘"Bayesian Measures of Model Complexity and Fit."‘ Journal of the Royal Statistical Society B, Part 4: 583-639. DOI: 10.1111/1467-9868.00353

Gelman, A., Carlin, J. B., Stern, H. S., and Rubin, D. B. ‘"Bayesian Data Analysis."‘ 2nd Edition, Chapman and Hall. DOI: 10.1002/sim.1856

See Also

decision criteria

Examples

```
set.seed(101)
data("data25j3")
y <- data25j3$y
xMat <- data25j3$x
k <- dim(xMat)[2]
b0 <- array(rep(0, k), dim = c(k, 1))
B0 <- 10*diag(k)
n0 <- 5
d0 <- 8
output <- quantregOR2(y = y, x = xMat, b0, B0, n0, d0, gammacp2 = 3,
burn = 10, mcmc = 40, p = 0.25, accutoff = 0.5, maxlags = 400, verbose = FALSE)
betadraws <- output$betadraws
sigmadraws <- output$sigmadraws
gammacp <- c(-Inf, 0, 3, Inf)
postMeanbeta <- output$postMeanbeta
postMeansigma <- output$postMeansigma
mcmc = 40
burn <- 10
nsim <- burn + mcmc
dic <- dicOR2(y, xMat, betadraws, sigmadraws, gammacp,
postMeanbeta, postMeansigma, burn, mcmc, p = 0.25)

# DIC
#   801.8191
# pd
#   6.608594
# dev
#   788.6019
```

`drawbetaOR1` *Samples β in the OR1 model*

Description

This function samples β from its conditional posterior distribution in the OR1 model (ordinal quantile model with 3 or more outcomes).

Usage

```
drawbetaOR1(z, x, w, tau2, theta, invB0, invB0b0)
```

Arguments

z	continuous latent values, vector of size $(nx1)$.
x	covariate matrix of size (nxk) including a column of ones with or without column names.

`w` latent weights, column vector of size size $(nx1)$.

`tau2` 2/(p(1-p)).

`theta` (1-2p)/(p(1-p)).

`invB0` inverse of prior covariance matrix of normal distribution.

`invB0b0` prior mean pre-multiplied by invB0.

Details

This function samples β, a vector, from its conditional posterior distribution which is an updated multivariate normal distribution.

Value

Returns a list with components

`beta:` β, a column vector of size $(kx1)$, sampled from its condtional posterior distribution.

`Btilde:` variance parameter for the posterior multivariate normal distribution.

`btilde:` mean parameter for the posterior multivariate normal distribution.

References

Rahman, M. A. (2016). ‘"Bayesian Quantile Regression for Ordinal Models."‘ Bayesian Analysis, 11(1): 1-24. DOI: 10.1214/15-BA939

See Also

Gibbs sampling, normal distribution, mvrnorm, inv

Examples

```
set.seed(101)
data("data25j4")
xMat <- data25j4$x
p <- 0.25
n <- dim(xMat)[1]
k <- dim(xMat)[2]
w <- array( (abs(rnorm(n, mean = 2, sd = 1))), dim = c (n, 1))
theta <- 2.666667
tau2 <- 10.66667
z <- array( (rnorm(n, mean = 0, sd = 1)), dim = c(n, 1))
b0 <- array(0, dim = c(k, 1))
B0 <- diag(k)
invB0 <- matrix(c(
     1, 0, 0,
     0, 1, 0,
     0, 0, 1),
     nrow = 3, ncol = 3, byrow = TRUE)
invB0b0 <- invB0 %*% b0
output <- drawbetaOR1(z, xMat, w, tau2, theta, invB0, invB0b0)
```

```
# output$beta
#   -0.2481837 0.7837995 -3.4680418
```

drawbetaOR2 *Samples β in the OR2 model*

Description

This function samples β from its conditional posterior distribution in the OR2 model (ordinal quantile model with exactly 3 outcomes).

Usage

```
drawbetaOR2(z, x, sigma, nu, tau2, theta, invB0, invB0b0)
```

Arguments

z	continuous latent values, vector of size $(nx1)$.
x	covariate matrix of size (nxk) including a column of ones with or without column names.
sigma	σ, a scalar value.
nu	modified latent weight, column vector of size $(nx1)$.
tau2	2/(p(1-p)).
theta	(1-2p)/(p(1-p)).
invB0	inverse of prior covariance matrix of normal distribution.
invB0b0	prior mean pre-multiplied by invB0.

Details

This function samples β, a vector, from its conditional posterior distribution which is an updated multivariate normal distribution.

Value

Returns a list with components

beta:	β, a column vector of size $(kx1)$, sampled from its condtional posterior distribution.
Btilde:	variance parameter for the posterior multivariate normal distribution.
btilde:	mean parameter for the posterior multivariate normal distribution.

References

Rahman, M. A. (2016). ‘"Bayesian Quantile Regression for Ordinal Models."‘ Bayesian Analysis, 11(1): 1-24. DOI: 10.1214/15-BA939

See Also

Gibbs sampling, normal distribution , rgig, inv

Examples

```
set.seed(101)
z <- c(21.01744, 33.54702, 33.09195, -3.677646,
 21.06553, 1.490476, 0.9618205, -6.743081, 21.02186, 0.6950479)
x <- matrix(c(
     1, -0.3010490, 0.8012506,
     1,  1.2764036, 0.4658184,
     1,  0.6595495, 1.7563655,
     1, -1.5024607, -0.8251381,
     1, -0.9733585, 0.2980610,
     1, -0.2869895, -1.0130274,
     1,  0.3101613, -1.6260663,
     1, -0.7736152, -1.4987616,
     1,  0.9961420, 1.2965952,
     1, -1.1372480, 1.7537353),
     nrow = 10, ncol = 3, byrow = TRUE)
sigma <- 1.809417
n <- dim(x)[1]
nu <- array(5 * rep(1,n), dim = c(n, 1))
tau2 <- 10.6667
theta <- 2.6667
invB0 <- matrix(c(
     1, 0, 0,
     0, 1, 0,
     0, 0, 1),
     nrow = 3, ncol = 3, byrow = TRUE)
invB0b0 <- c(0, 0, 0)

output <- drawbetaOR2(z, x, sigma, nu, tau2, theta, invB0, invB0b0)

# output$beta
#   -0.74441 1.364846 0.7159231
```

---

drawdeltaOR1 *Samples δ in the OR1 model*

Description

This function samples the cut-point vector δ using a random-walk Metropolis-Hastings algorithm in the OR1 model (ordinal quantile model with 3 or more outcomes).

Usage

```
drawdeltaOR1(y, x, beta, delta0, d0, D0, tune, Dhat, p)
```

Arguments

`y`	observed ordinal outcomes, column vector of size $(nx1)$.
`x`	covariate matrix of size (nxk) including a column of ones with or without column names.
`beta`	Gibbs draw of β, column vector of size $(kx1)$.
`delta0`	initial value for δ.
`d0`	prior mean for δ.
`D0`	prior covariance matrix for δ.
`tune`	tuning parameter to adjust MH acceptance rate.
`Dhat`	negative inverse Hessian from maximization of log-likelihood.
`p`	quantile level or skewness parameter, p in (0,1).

Details

Samples the cut-point vector δ using a random-walk Metropolis-Hastings algorithm.

Value

Returns a list with components

`deltaReturn:`	δ, a column vector of size $((J-2)x1)$, sampled using MH algorithm.
`accept:`	indicator for acceptance of the proposed value of δ.

References

Rahman, M. A. (2016). ‘"Bayesian Quantile Regression for Ordinal Models."‘ Bayesian Analysis, 11(1): 1-24. DOI: 10.1214/15-BA939

Chib, S., and Greenberg, E. (1995). ‘"Understanding the Metropolis-Hastings Algorithm."‘ The American Statistician, 49(4): 327-335. DOI: 10.2307/2684568

Jeliazkov, I., and Rahman, M. A. (2012). ‘"Binary and Ordinal Data Analysis in Economics: Modeling and Estimation"‘ in Mathematical Modeling with Multidisciplinary Applications, edited by X.S. Yang, 123-150. John Wiley & Sons Inc, Hoboken, New Jersey. DOI: 10.1002/9781118462706.ch6

See Also

NPflow, Gibbs sampling, mvnpdf

Examples

```
set.seed(101)
data("data25j4")
y <- data25j4$y
xMat <- data25j4$x
p <- 0.25
beta <- c(0.3990094, 0.8168991, 2.8034963)
delta0 <- c(-0.9026915, -2.2488833)
d0 <- matrix(c(0, 0),
                 nrow = 2, ncol = 1, byrow = TRUE)
D0 <- matrix(c(0.25, 0.00, 0.00, 0.25),
                    nrow = 2, ncol = 2, byrow = TRUE)
tune <- 0.1
Dhat <- matrix(c(0.046612180, -0.001954257, -0.001954257, 0.083066204),
             nrow = 2, ncol = 2, byrow = TRUE)
p <- 0.25
output <- drawdeltaOR1(y, xMat, beta, delta0, d0, D0, tune, Dhat, p)

# deltareturn
#   -0.9025802 -2.229514
# accept
#   1
```

`drawlatentOR1` *Samples latent variable z in the OR1 model*

Description

This function samples the latent variable z from a univariate truncated normal distribution in the OR1 model (ordinal quantile model with 3 or more outcomes).

Usage

```
drawlatentOR1(y, x, beta, w, theta, tau2, delta)
```

Arguments

`y` observed ordinal outcomes, column vector of size $(nx1)$.

`x` covariate matrix of size (nxk) including a column of ones with or without column names.

`beta` Gibbs draw of β, a column vector of size $(kx1)$.

`w` latent weights, column vector of size $(nx1)$.

`theta` (1-2p)/(p(1-p)).

`tau2` 2/(p(1-p)).

`delta` column vector of cutpoints including (-Inf, Inf).

Details

This function samples the latent variable z from a univariate truncated normal distribution.

Value

latent variable z of size $(nx1)$ sampled from a univariate truncated distribution.

References

Albert, J., and Chib, S. (1993). ‘"Bayesian Analysis of Binary and Polychotomous Response Data."‘ Journal of the American Statistical Association, 88(422): 669‘-‘679. DOI: 10.1080/01621459.1993.10476321

Robert, C. P. (1995). ‘"Simulation of truncated normal variables."‘ Statistics and Computing, 5: 121‘-‘125. DOI: 10.1007/BF00143942

See Also

Gibbs sampling, truncated normal distribution, rtruncnorm

Examples

```
set.seed(101)
data("data25j4")
y <- data25j4$y
xMat <- data25j4$x
p <- 0.25
beta <- c(0.3990094, 0.8168991, 2.8034963)
w <- 1.114347
theta <- 2.666667
tau2 <- 10.66667
delta <- c(-0.002570995,  1.044481071)
output <- drawlatentOR1(y, xMat, beta, w, theta, tau2, delta)

# output
#   0.6261896 3.129285 2.659578 8.680291
#   13.22584 2.545938 1.507739 2.167358
#   15.03059 -3.963201 9.237466 -1.813652
#   2.718623 -3.515609 8.352259 -0.3880043
#   -0.8917078 12.81702 -0.2009296 1.069133 ... soon
```

`drawlatentOR2` *Samples latent variable z in the OR2 model*

Description

This function samples the latent variable z from a univariate truncated normal distribution in the OR2 model (ordinal quantile model with exactly 3 outcomes).

Usage

```
drawlatentOR2(y, x, beta, sigma, nu, theta, tau2, gammacp)
```

Arguments

y	observed ordinal outcomes, column vector of size $(nx1)$.
x	covariate matrix of size (nxk) including a column of ones with or without column names.
beta	Gibbs draw of β, a column vector of size $(kx1)$.
sigma	σ, a scalar value.
nu	modified latent weight, column vector of size $(nx1)$.
theta	(1-2p)/(p(1-p)).
tau2	2/(p(1-p)).
gammacp	row vector of cut-points including -Inf and Inf.

Details

This function samples the latent variable z from a univariate truncated normal distribution.

Value

latent variable z of size $(nx1)$ from a univariate truncated distribution.

References

Albert, J., and Chib, S. (1993). ‘"Bayesian Analysis of Binary and Polychotomous Response Data."‘ Journal of the American Statistical Association, 88(422): 669‘-‘679. DOI: 10.1080/01621459.1993.10476321

Devroye, L. (2014). ‘"Random variate generation for the generalized inverse Gaussian distribution."‘ Statistics and Computing, 24(2): 239‘-‘246. DOI: 10.1007/s11222-012-9367-z

See Also

Gibbs sampling, truncated normal distribution, rtruncnorm

Examples

```
set.seed(101)
data("data25j3")
y <- data25j3$y
xMat <- data25j3$x
beta <- c(1.810504, 1.850332, 6.181163)
sigma <- 0.9684741
n <- dim(xMat)[1]
nu <- array(5 * rep(1,n), dim = c(n, 1))
theta <- 2.6667
tau2 <- 10.6667
gammacp <- c(-Inf, 0, 3, Inf)
output <- drawlatentOR2(y, xMat, beta, sigma, nu,
theta, tau2, gammacp)

# output
#   1.257096 10.46297 4.138694
#   28.06432 4.179275 19.21582
#   11.17549 13.79059 28.3650 .. soon
```

drawnuOR2 *Samples scale factor ν in the OR2 model*

Description

This function samples ν from a generalized inverse Gaussian (GIG) distribution in the OR2 model (ordinal quantile model with exactly 3 outcomes).

Usage

```
drawnuOR2(z, x, beta, sigma, tau2, theta, indexp)
```

Arguments

`z` Gibbs draw of continuous latent values, a column vector of size $(nx1)$.

`x` covariate matrix of size (nxk) including a column of ones.

`beta` Gibbs draw of β, a column vector of size $(kx1)$.

`sigma` σ, a scalar value.

`tau2` 2/(p(1-p)).

`theta` (1-2p)/(p(1-p)).

`indexp` index parameter of the GIG distribution which is equal to 0.5.

Details

This function samples ν from a GIG distribution.

Value

ν, a column vector of size $(nx1)$, sampled from a GIG distribution.

References

Rahman, M. A. (2016), ‘"Bayesian Quantile Regression for Ordinal Models."‘ Bayesian Analysis, 11(1), 1-24. DOI: 10.1214/15-BA939

Devroye, L. (2014). ‘"Random variate generation for the generalized inverse Gaussian distribution."‘ Statistics and Computing, 24(2): 239‘-‘246. DOI: 10.1007/s11222-012-9367-z

See Also

GIGrvg, Gibbs sampling, rgig

Examples

```
set.seed(101)
z <- c(21.01744, 33.54702, 33.09195, -3.677646,
 21.06553, 1.490476, 0.9618205, -6.743081, 21.02186, 0.6950479)
x <- matrix(c(
     1, -0.3010490, 0.8012506,
     1,  1.2764036, 0.4658184,
     1,  0.6595495, 1.7563655,
     1, -1.5024607, -0.8251381,
     1, -0.9733585, 0.2980610,
     1, -0.2869895, -1.0130274,
     1,  0.3101613, -1.6260663,
     1, -0.7736152, -1.4987616,
     1, 0.9961420, 1.2965952,
     1, -1.1372480, 1.7537353),
     nrow = 10, ncol = 3, byrow = TRUE)
beta <- c(-0.74441, 1.364846, 0.7159231)
sigma <- 3.749524
tau2 <- 10.6667
theta <- 2.6667
indexp <- 0.5
output <- drawnuOR2(z, x, beta, sigma, tau2, theta, indexp)

# output
#   5.177456 4.042261 8.950365
#   1.578122 6.968687 1.031987
#   4.13306 0.4681557 5.109653
#   0.1725333
```

drawsigmaOR2 *Samples σ in the OR2 model*

Description

This function samples σ from an inverse-gamma distribution in the OR2 model (ordinal quantile model with exactly 3 outcomes).

Usage

```
drawsigmaOR2(z, x, beta, nu, tau2, theta, n0, d0)
```

Arguments

`z` Gibbs draw of continuous latent values, a column vector of size $nx1$.

`x` covariate matrix of size (nxk) including a column of ones with or without column names.

`beta` Gibbs draw of β, a column vector of size $(kx1)$.

`nu` modified latent weight, column vector of size $(nx1)$.

`tau2` 2/(p(1-p)).

`theta` (1-2p)/(p(1-p)).

`n0` prior hyper-parameter for σ.

`d0` prior hyper-parameter for σ.

Details

This function samples σ from an inverse-gamma distribution.

Value

Returns a list with components

`sigma:` σ, a scalar, sampled from an inverse gamma distribution.

`dtilde:` scale parameter of the inverse-gamma distribution.

References

Rahman, M. A. (2016). ‘"Bayesian Quantile Regression for Ordinal Models."‘ Bayesian Analysis, 11(1): 1-24. DOI: 10.1214/15-BA939

Devroye, L. (2014). ‘"Random variate generation for the generalized inverse Gaussian distribution."‘ Statistics and Computing, 24(2): 239‘-‘246. DOI: 10.1007/s11222-012-9367-z

See Also

rgamma, Gibbs sampling

Examples

```
set.seed(101)
z <- c(21.01744, 33.54702, 33.09195, -3.677646,
 21.06553, 1.490476, 0.9618205, -6.743081, 21.02186, 0.6950479)
x <- matrix(c(
     1, -0.3010490, 0.8012506,
     1,  1.2764036, 0.4658184,
     1,  0.6595495, 1.7563655,
     1, -1.5024607, -0.8251381,
     1, -0.9733585, 0.2980610,
     1, -0.2869895, -1.0130274,
     1,  0.3101613, -1.6260663,
     1, -0.7736152, -1.4987616,
     1,  0.9961420, 1.2965952,
     1, -1.1372480, 1.7537353),
     nrow = 10, ncol = 3, byrow = TRUE)
beta <- c(-0.74441, 1.364846, 0.7159231)
n <- dim(x)[1]
nu <- array(5 * rep(1,n), dim = c(n, 1))
tau2 <- 10.6667
theta <- 2.6667
```

```
n0 <- 5
d0 <- 8
output <- drawsigmaOR2(z, x, beta, nu, tau2, theta, n0, d0)

# output$sigma
#   3.749524
```

`drawwOR1` *Samples latent weight w in the OR1 model*

Description

This function samples latent weight w from a generalized inverse-Gaussian distribution (GIG) in the OR1 model (ordinal quantile model with 3 or more outcomes).

Usage

```
drawwOR1(z, x, beta, tau2, theta, indexp)
```

Arguments

`z`	continuous latent values, vector of size $(nx1)$.
`x`	covariate matrix of size (nxk) including a column of ones with or without column names.
`beta`	Gibbs draw of β, a column vector of size $(kx1)$.
`tau2`	2/(p(1-p)).
`theta`	(1-2p)/(p(1-p)).
`indexp`	index parameter of GIG distribution which is equal to 0.5

Details

This function samples a vector of latent weight w from a GIG distribution.

Value

w, a column vector of size $(nx1)$, sampled from a GIG distribution.

References

Rahman, M. A. (2016). ‘"Bayesian Quantile Regression for Ordinal Models."‘ Bayesian Analysis, 11(1): 1-24. DOI: 10.1214/15-BA939

Devroye, L. (2014). ‘"Random variate generation for the generalized inverse Gaussian distribution."‘ Statistics and Computing, 24(2): 239‘-‘246. DOI: 10.1007/s11222-012-9367-z

See Also

GIGrvg, Gibbs sampling, rgig

Examples

```
set.seed(101)
z <- c(0.9812363, -1.09788, -0.9650175, 8.396556,
 1.39465, -0.8711435, -0.5836833, -2.792464,
 0.1540086, -2.590724, 0.06169976, -1.823058,
 0.06559151, 0.1612763, 0.161311, 4.908488,
 0.6512113, 0.1560708, -0.883636, -0.5531435)
x <- matrix(c(
     1, 1.4747905363, 0.167095186,
     1, -0.3817326861, 0.041879526,
     1, -0.1723095575, -1.414863777,
     1, 0.8266428137, 0.399722073,
     1, 0.0514888733, -0.105132425,
     1, -0.3159992662, -0.902003846,
     1, -0.4490888878, -0.070475600,
     1, -0.3671705251, -0.633396477,
     1, 1.7655601639, -0.702621934,
     1, -2.4543678120, -0.524068780,
     1,  0.3625025618,  0.698377504,
     1, -1.0339179063,  0.155746376,
     1,  1.2927374692, -0.155186911,
     1, -0.9125108094, -0.030513775,
     1,  0.8761233001,  0.988171587,
     1,  1.7379728231,  1.180760114,
     1,  0.7820635770, -0.338141095,
     1, -1.0212853209, -0.113765067,
     1,  0.6311364051, -0.061883874,
     1,  0.6756039688,  0.664490143),
     nrow = 20, ncol = 3, byrow = TRUE)
beta <- c(-1.583533, 1.407158, 2.259338)
tau2 <- 10.66667
theta <- 2.666667
indexp <- 0.5
output <- drawwOR1(z, x, beta, tau2, theta, indexp)

# output
#   0.16135732
#   0.39333080
#   0.80187227
#   2.27442898
#   0.90358310
#   0.99886987
#   0.41515947 ... soon
```

`Educational_Attainment` *Educational Attainment study based on data from the National Longitudinal Study of Youth (NLSY, 1979) survey.*

Description

Educational Attainment study based on data from the National Longitudinal Study of Youth (NLSY, 1979) survey.

Usage

```
data(Educational_Attainment)
```

Details

This data is taken from the National Longitudinal Study of Youth (NLSY, 1979) survey and corresponds to 3,923 individuals. The objective is to study the effect of family background, individual, and school level variables on the quantiles of educational attainment conditional on the covariates. The dependent variable i.e. the educational degree, has four categories given as less than high school, high school degree, some college or associate's degree, and college or graduate degree. The independent variables include intercept, square root of family income, mother's education, father's education, mother's working status, gender, race, and whether the youth lived in an urban area at the age of 14, and indicator variables to control for age-cohort effects.

Value

Returns data with components

`mother_work:` Indicator for working female at the age of 14.

`urban:` Indicator for the youth living in urban area at the age of 14.

`south:` Indicator for the youth living in South at the age of 14.

`father_educ:` Number of years of father's education.

`mother_educ:` Number of years of mother's education.

`fam_income:` Family income of the household in $1000.

`female:` Indicator for individual's gender.

`black:` Indicator for black race.

`age_cohort_2:` Indicator variable for age 15.

`age_cohort_3:` Indicator variable for age 16.

`age_cohort_4:` Indicator variable for age 17.

`dep_edu_level:` Four categories of educational attainment: less than high school, high school degree, some college or associate's degree, and college or graduate degree.

References

Rahman, M. A. (2016). ‘"Bayesian Quantile Regression for Ordinal Models."‘ Bayesian Analysis, 11(1): 1-24. DOI: 10.1214/15-BA939

Jeliazkov, I., Graves, J., and Kutzbach, M. (2008). ‘"Fitting and Comparison of Models for Multivariate Ordinal Outcomes."‘ Advances in Econometrics: Bayesian Econometrics, 23: 115‘-‘156. DOI: 10.1016/S0731-9053(08)23004-5

Jeliazkov, I., and Rahman, M. A. (2012). ‘"Binary and Ordinal Data Analysis in Economics: Modeling and Estimation"‘ in Mathematical Modeling with Multidisciplinary Applications, edited by X.S. Yang, 123-150. John Wiley ‘&‘ Sons Inc, Hoboken, New Jersey. DOI: 10.1002/9781118462706.ch6

See Also

Survey Process.

`ineffactorOR1` *Inefficiency factor in the OR1 model*

Description

This function calculates the inefficiency factor from the MCMC draws of (β, δ) in the OR1 model (ordinal quantile model with 3 or more outcomes). The inefficiency factor is calculated using the batch-means method.

Usage

```
ineffactorOR1(x, betadraws, deltadraws, accutoff, maxlags, verbose)
```

Arguments

`x`	covariate matrix of size (nxk) including a column of ones with or without column names. This input is used to extract column names, if available, but not used in calculation.
`betadraws`	dataframe of the MCMC draws of β, size $(kxnsim)$.
`deltadraws`	dataframe of the MCMC draws of δ, size $((J-2)xnsim)$.
`accutoff`	cut-off to identify the number of lags and form batches, default is 0.05.
`maxlags`	maximum lag at which to calculate the acf, default is 400.
`verbose`	whether to print the final output and provide additional information or not, default is TRUE.

Details

Calculates the inefficiency factor of (β, δ) using the batch-means method based on MCMC draws. Inefficiency factor can be interpreted as the cost of working with correlated draws. A low inefficiency factor indicates better mixing and an efficient algorithm.

Value

Returns a column vector of inefficiency factors for each component of β and δ.

References

Greenberg, E. (2012). ‘"Introduction to Bayesian Econometrics."‘ Cambridge University Press, Cambridge. DOI: 10.1017/CBO9780511808920

Chib, S. (2012), ‘"Introduction to simulation and MCMC methods."‘ In Geweke J., Koop G., and Dijk, H.V., editors, ‘"The Oxford Handbook of Bayesian Econometrics"‘, pages 183–218. Oxford University Press, Oxford. DOI: 10.1093/oxfordhb/9780199559084.013.0006

See Also

pracma, acf

Examples

```
set.seed(101)
data("data25j4")
y <- data25j4$y
xMat <- data25j4$x
k <- dim(xMat)[2]
J <- dim(as.array(unique(y)))[1]
b0 <- array(rep(0, k), dim = c(k, 1))
B0 <- 10*diag(k)
d0 <- array(0, dim = c(J-2, 1))
D0 <- 0.25*diag(J - 2)
output <- quantregOR1(y = y, x = xMat, b0, B0, d0, D0,
burn = 10, mcmc = 40, p = 0.25, tune = 1, accutoff = 0.5, maxlags = 400, verbose = FALSE)
betadraws <- output$betadraws
deltadraws <- output$deltadraws
inefficiency <- ineffactorOR1(xMat, betadraws, deltadraws, 0.5, 400, TRUE)

# Summary of Inefficiency Factor:

#              Inef Factor
# beta_1         1.1008
# beta_2         3.0024
# beta_3         2.8543
# delta_1        3.6507
# delta_2        3.1784
```

ineffactorOR2 *Inefficiency factor in the OR2 model*

Description

This function calculates the inefficiency factor from the MCMC draws of (β, σ) in the OR2 model (ordinal quantile model with exactly 3 outcomes). The inefficiency factor is calculated using the batch-means method.

Usage

```
ineffactorOR2(x, betadraws, sigmadraws, accutoff, maxlags, verbose)
```

Arguments

`x`	covariate matrix of size (nxk) including a column of ones with or without column names. This input is used to extract column names, if available, but not used in calculation.
`betadraws`	dataframe of the Gibbs draws of β, size $(kxnsim)$.
`sigmadraws`	dataframe of the Gibbs draws of σ, size $(1xnsim)$.
`accutoff`	cut-off to identify the number of lags and form batches, default is 0.05.
`maxlags`	maximum lag at which to calculate the acf, default is 400.
`verbose`	whether to print the final output and provide additional information or not, default is TRUE.

Details

Calculates the inefficiency factor of (β, σ) using the batch-means method based on the Gibbs draws. Inefficiency factor can be interpreted as the cost of working with correlated draws. A low inefficiency factor indicates better mixing and an efficient algorithm.

Value

Returns a column vector of inefficiency factors for each component of β and σ.

References

Greenberg, E. (2012). ‘"Introduction to Bayesian Econometrics."‘ Cambridge University Press, Cambridge. DOI: 10.1017/CBO9780511808920

Chib, S. (2012), ‘"Introduction to simulation and MCMC methods."‘ In Geweke J., Koop G., and Dijk, H.V., editors, ‘"The Oxford Handbook of Bayesian Econometrics"‘, pages 183–218. Oxford University Press, Oxford. DOI: 10.1093/oxfordhb/9780199559084.013.0006

See Also

pracma, acf

Examples

```
set.seed(101)
data("data25j3")
y <- data25j3$y
xMat <- data25j3$x
k <- dim(xMat)[2]
b0 <- array(rep(0, k), dim = c(k, 1))
B0 <- 10*diag(k)
n0 <- 5
d0 <- 8
output <- quantregOR2(y = y, x = xMat, b0, B0, n0, d0, gammacp2 = 3,
burn = 10, mcmc = 40, p = 0.25, accutoff = 0.5, maxlags = 400, verbose = FALSE)
betadraws <- output$betadraws
sigmadraws <- output$sigmadraws

inefficiency <- ineffactorOR2(xMat, betadraws, sigmadraws, 0.5, 400, TRUE)

# Summary of Inefficiency Factor:
#            Inef Factor
# beta_1       1.5686
# beta_2       1.5240
# beta_3       1.4807
# sigma        2.4228
```

---

logMargLikeOR1 *Marginal likelihood in the OR1 model*

Description

This function computes the logarithm of marginal likelihood in the OR1 model (ordinal quantile model with 3 or more outcomes) using the MCMC outputs from the complete and reduced runs.

Usage

```
logMargLikeOR1(y, x, b0, B0, d0, D0, postMeanbeta,
postMeandelta, betadraws, deltadraws, tune, Dhat, p, verbose)
```

Arguments

`y` observed ordinal outcomes, column vector of size $(nx1)$.

`x` covariate matrix of size (nxk) including a column of ones with or without column names.

`b0` prior mean for β.

`B0` prior covariance matrix for β

`d0` prior mean for δ.

`D0` prior covariance matrix for δ.

`postMeanbeta`	posterior mean of β from the complete MCMC run.
`postMeandelta`	posterior mean of δ from the complete MCMC run.
`betadraws`	a dataframe with all the sampled values for β from the complete MCMC run.
`deltadraws`	a dataframe with all the sampled values for δ from the complete MCMC run.
`tune`	tuning parameter to adjust the MH acceptance rate.
`Dhat`	negative inverse Hessian from the maximization of log-likelihood.
`p`	quantile level or skewness parameter, p in (0,1).
`verbose`	whether to print the final output and provide additional information or not, default is TRUE.

Details

This function computes the logarithm of marginal likelihood in the OR1 model using the MCMC outputs from complete and reduced runs.

Value

Returns an estimate of log marginal likelihood

References

Chib, S. (1995). ‘"Marginal likelihood from the Gibbs output."‘ Journal of the American Statistical Association, 90(432):1313 to 1321, 1995. DOI: 10.1080/01621459.1995.10476635

Chib, S., and Jeliazkov, I. (2001). ‘"Marginal likelihood from the Metropolis-Hastings output."‘ Journal of the American Statistical Association, 96(453):270‘-‘281, 2001. DOI: 10.1198/016214501750332848

See Also

mvnpdf, dnorm, Gibbs sampling, Metropolis-Hastings algorithm

Examples

```
set.seed(101)
data("data25j4")
y <- data25j4$y
xMat <- data25j4$x
k <- dim(xMat)[2]
J <- dim(as.array(unique(y)))[1]
b0 <- array(rep(0, k), dim = c(k, 1))
B0 <- 10*diag(k)
d0 <- array(0, dim = c(J-2, 1))
D0 <- 0.25*diag(J - 2)
output <- quantregOR1(y = y, x = xMat, b0, B0, d0, D0,
burn = 10, mcmc = 40, p = 0.25, tune = 1, accutoff = 0.5, maxlags = 400, verbose = FALSE)
# output$logMargLike
#   -559.73
```

logMargLikeOR2 *Marginal likelihood in the OR2 model*

Description

This function computes the logarithm of marginal likelihood in the OR2 model (ordinal quantile model with exactly 3 outcomes) using the Gibbs output from the complete and reduced runs.

Usage

```
logMargLikeOR2(y, x, b0, B0, n0, d0, postMeanbeta, postMeansigma,
btildeStore, BtildeStore, gammacp2, p, verbose)
```

Arguments

`y`	observed ordinal outcomes, column vector of size $(nx1)$.
`x`	covariate matrix of size (nxk) including a column of ones with or without column names.
`b0`	prior mean for β.
`B0`	prior covariance matrix for β.
`n0`	prior shape parameter of inverse-gamma distribution for σ.
`d0`	prior scale parameter of inverse-gamma distribution for σ.
`postMeanbeta`	posterior mean of β from the complete Gibbs run.
`postMeansigma`	posterior mean of δ from the complete Gibbs run.
`btildeStore`	a storage matrix for btilde from the complete Gibbs run.
`BtildeStore`	a storage matrix for Btilde from the complete Gibbs run.
`gammacp2`	one and only cut-point other than 0.
`p`	quantile level or skewness parameter, p in (0,1).
`verbose`	whether to print the final output and provide additional information or not, default is TRUE.

Details

This function computes the logarithm of marginal likelihood in the OR2 model using the Gibbs output from the complete and reduced runs.

Value

Returns an estimate of log marginal likelihood

References

Chib, S. (1995). ‘"Marginal likelihood from the Gibbs output."‘ Journal of the American Statistical Association, 90(432):1313‘-‘1321, 1995. DOI: 10.1080/01621459.1995.10476635

See Also

dinvgamma, mvnpdf, dnorm, Gibbs sampling

Examples

```
set.seed(101)
data("data25j3")
y <- data25j3$y
xMat <- data25j3$x
k <- dim(xMat)[2]
b0 <- array(rep(0, k), dim = c(k, 1))
B0 <- 10*diag(k)
n0 <- 5
d0 <- 8
output <- quantregOR2(y = y, x = xMat, b0, B0, n0, d0, gammacp2 = 3,
burn = 10, mcmc = 40, p = 0.25, accutoff = 0.5, maxlags = 400, verbose = FALSE)
# output$logMargLike
#   -404.57
```

Policy_Opinion *Data contains public opinion on the proposal to raise federal income taxes for couples (individuals) earning more than $250,000 ($200,000) per year and a host of other covariates. The data is taken from the 2010-2012 American National Election Studies (ANES) on the Evaluation of Government and Society Study I (EGSS 1)*

Description

Data contains public opinion on the proposal to raise federal income taxes for couples (individuals) earning more than $250,000 ($200,000) per year and a host of other covariates. The data is taken from the 2010-2012 American National Election Studies (ANES) on the Evaluation of Government and Society Study I (EGSS 1)

Usage

```
data(Policy_Opinion)
```

Details

The data consists of 1,164 observations taken from the 2010-2012 American National Election Studies (ANES) on the Evaluations of Government and Society Study 1 (EGSS 1). The objective is to analyze public opinion on the proposal to raise federal income taxes for couples (individuals) earning more than $250,000 ($200,000) per year. The responses were recorded as oppose, neither favor nor oppose, or favor the tax increase, and forms the dependent variable in the study. The independent variables include indicator variables (or dummy) for employment, income above $75,000, bachelor's and post-bachelor's degree, computer ownership, cellphone ownership, and white race.

Value

Returns data with components

`Intercept:`	Column of ones.
`EmpCat:`	Indicator for employment status.
`IncomeCat:`	Indicator for household income > $75,000.
`Bachelors:`	Individual’s highest degree is Bachelors.
`Post.Bachelors:`	Indicator for highest degree is Masters, Professional or Doctorate.
`Computers:`	Indicator for computer ownership by individual or household.
`CellPhone:`	Indicator for cellphone ownership by individual or household.
`White:`	Indicator for White race.
`y:`	Public opinion on the proposal to raise federal income taxes. The three categories are: oppose, neither favor nor oppose, or favor the tax increase.

References

Rahman, M. A. (2016). ‘"Bayesian Quantile Regression for Ordinal Models."‘ Bayesian Analysis, 11(1): 1-24. DOI: 10.1214/15-BA939

See Also

ANES, Tax Policy

`qrminfundtheorem` *Minimizes the negative of log-likelihood in the OR1 model*

Description

This function minimizes the negative of log-likelihood in the OR1 model with respect to the cut-points δ using the fundamental theorem of calculus.

Usage

```
qrminfundtheorem(deltaIn, y, x, beta, cri0, cri1, stepsize, maxiter, h, dh, sw, p)
```

Arguments

`deltaIn`	initialization of cut-points.
`y`	observed ordinal outcomes, column vector of size $(nx1)$.
`x`	covariate matrix of size (nxk) including a column of ones with or without column names.
`beta`	β, a column vector of size $(kx1)$.
`cri0`	initial criterion, $cri0 = 1$.

`cri1` criterion lies between (0.001 to 0.0001).

`stepsize` learning rate lies between (0.1, 1).

`maxiter` maximum number of iteration.

`h` change in each value of δ, holding other δ constant for first derivatives.

`dh` change in each value of δ, holding other δ constant for second derivaties.

`sw` iteration to switch from BHHH to inv(-H) algorithm.

`p` quantile level or skewness parameter, p in (0,1).

Details

First derivative from first principle

$$dy/dx = [f(x+h) - f(x-h)]/2h$$

Second derivative from first principle

$$\begin{aligned} f'(x-h) &= (f(x) - f(x-h))/h \\ f''(x) &= [(f(x+h) - f(x))/h - (f(x) - f(x-h))/h]/h \\ &= [(f(x+h) + f(x-h) - 2f(x))]/h^2 \end{aligned}$$

cross partial derivatives

$$\begin{aligned} f(x) &= [f(x+dh,y) - f(x-dh,y)]/2dh \\ f(x,y) &= [(f(x+dh,y+dh) - f(x+dh,y-dh))/2dh - (f(x-dh,y+dh) - f(x-dh,y-dh))/2dh]/2dh \\ &= 0.25*[(f(x+dh,y+dh) - f(x+dh,y-dh)) - (f(x-dh,y+dh) - f(x-dh,y-dh))]/dh2 \end{aligned}$$

Value

Returns a list with components

`deltamin:` cutpoint vector that minimizes the log-likelihood function.

`negsum:` negative sum of log-likelihood.

`logl:` log-likelihood values.

`G:` gradient vector, (nxk) matrix with i-th row as the score for the i-th unit.

`H:` Hessian matrix.

See Also

differential calculus, functional maximization, mldivide

Examples

```
set.seed(101)
deltaIn <- c(-0.002570995,  1.044481071)
data("data25j4")
y <- data25j4$y
xMat <- data25j4$x
p <- 0.25
beta <- c(0.3990094, 0.8168991, 2.8034963)
cri0     <- 1
cri1     <- 0.001
stepsize <- 1
maxiter  <- 10
h        <- 0.002
dh       <- 0.0002
sw       <- 20
output <- qrminfundtheorem(deltaIn, y, xMat, beta, cri0, cri1, stepsize, maxiter, h, dh, sw, p)

# deltamin
#   0.8266967 0.3635708
# negsum
#   645.4911
# logl
#    -0.7136999
#    -1.5340787
#    -1.1072447
#    -1.4423124
#    -1.3944677
#    -0.7941271
#    -1.6544072
#    -0.3246632
#    -1.8582422
#    -0.9220822
#    -2.1117739 .. soon
# G
#    0.803892784  0.00000000
#   -0.420190546  0.72908381
#   -0.421776117  0.72908341
#   -0.421776117 -0.60184063
#   -0.421776117 -0.60184063
#    0.151489598  0.86175120
#    0.296995920  0.96329114
#   -0.421776117  0.72908341
#   -0.340103190 -0.48530164
#    0.000000000  0.00000000
#   -0.421776117 -0.60184063.. soon
# H
#   -338.21243  -41.10775
#   -41.10775 -106.32758
```

qrnegLogLikensumOR1 *Negative log-likelihood in the OR1 model*

Description

This function computes the negative of log-likelihood for each individual and negative sum of log-likelihood in the OR1 model.

Usage

```
qrnegLogLikensumOR1(y, x, betaOne, deltaOne, p)
```

Arguments

`y`	observed ordinal outcomes, column vector of size $(nx1)$.
`x`	covariate matrix of size (nxk) including a column of ones with or without column names.
`betaOne`	a sample draw of β of size $(kx1)$.
`deltaOne`	a sample draw of δ of size $((J-2)x1)$.
`p`	quantile level or skewness parameter, p in (0,1).

Details

This function computes the negative of log-likelihood for each individual and negative sum of log-likelihood in the OR1 model.

The latter when evaluated at postMeanbeta and postMeandelta is used to calculate the DIC and may also be utilized to calculate the Akaike information criterion (AIC) and the Bayesian information criterion (BIC).

Value

Returns a list with components

`nlogl:`	vector of negative log-likelihood values.
`negsumlogl:`	negative sum of log-likelihood.

References

Rahman, M. A. (2016). ‘"Bayesian Quantile Regression for Ordinal Models."‘ Bayesian Analysis, 11(1): 1-24. DOI: 10.1214/15-BA939

See Also

likelihood maximization

Examples

```
set.seed(101)
deltaOne <- c(-0.002570995, 1.044481071)
data("data25j4")
y <- data25j4$y
xMat <- data25j4$x
p <- 0.25
betaOne <- c(0.3990094, 0.8168991, 2.8034963)
output <- qrnegLogLikensumOR1(y, xMat, betaOne, deltaOne, p)

# nlogl
#   0.7424858
#   1.1649645
#   2.1344390
#   0.9881085
#   2.7677386
#   0.8229129
#   0.8854911
#   0.3534490
#   1.8582422
#   0.9508680 .. soon

# negsumlogl
#   663.5475
```

qrnegLogLikeOR2 *Negative sum of log-likelihood in the OR2 model*

Description

This function computes the negative sum of log-likelihood in the OR2 model (ordinal quantile model with exactly 3 outcomes).

Usage

```
qrnegLogLikeOR2(y, x, gammacp, betaOne, sigmaOne, p)
```

Arguments

y	observed ordinal outcomes, column vector of size $(nx1)$.
x	covariate matrix of size (nxk) including a column of ones with or without column names.
gammacp	a row vector of cutpoints including (-Inf, Inf).
betaOne	a sample draw of β of size $(kx1)$.
sigmaOne	a sample draw of σ, a scalar value.
p	quantile level or skewness parameter, p in (0,1).

Details

This function computes the negative sum of log-likelihood in the OR2 model where the error is assumed to follow an AL distribution.

Value

Returns the negative sum of log-likelihood.

References

Rahman, M. A. (2016). ‘"Bayesian Quantile Regression for Ordinal Models."‘ Bayesian Analysis, 11(1): 1-24. DOI: 10.1214/15-BA939

See Also

likelihood maximization

Examples

```
set.seed(101)
data("data25j3")
y <- data25j3$y
xMat <- data25j3$x
p <- 0.25
gammacp <- c(-Inf, 0, 3, Inf)
betaOne <- c(1.810504, 1.850332, 6.18116)
sigmaOne <- 0.9684741
output <- qrnegLogLikeOR2(y, xMat, gammacp, betaOne, sigmaOne, p)

# output
#   902.4045
```

---

quantregOR1 *Bayesian quantile regression in the OR1 model*

Description

This function estimates Bayesian quantile regression in the OR1 model (ordinal quantile model with 3 or more outcomes) and reports the posterior mean, posterior standard deviation, 95 percent posterior credible intervals, and inefficiency factor of (β, δ). The output also displays the log of marginal likelihood and the DIC.

Usage

```
quantregOR1(y, x, b0, B0, d0, D0, burn, mcmc, p, tune, accutoff, maxlags, verbose)
```

Arguments

`y` observed ordinal outcomes, column vector of size $(nx1)$.

`x` covariate matrix of size (nxk) including a column of ones with or without column names.

`b0` prior mean for β.

`B0` prior covariance matrix for β.

`d0` prior mean for δ.

`D0` prior covariance matrix for δ.

`burn` number of burn-in MCMC iterations.

`mcmc` number of MCMC iterations, post burn-in.

`p` quantile level or skewness parameter, p in (0,1).

`tune` tuning parameter to adjust MH acceptance rate, default is 0.1.

`accutoff` autocorrelation cut-off to identify the number of lags and form batches to compute the inefficiency factor, default is 0.05.

`maxlags` maximum lag at which to calculate the acf in inefficiency factor calculation, default is 400.

`verbose` whether to print the final output and provide additional information or not, default is TRUE.

Details

This function estimates Bayesian quantile regression for the OR1 model using a combination of Gibbs sampling and Metropolis-Hastings algorithm. The function takes the prior distributions and other information as inputs and then iteratively samples β, latent weight w, δ, and latent variable z from their respective conditional distributions.

The function also provides the logarithm of marginal likelihood and the DIC. These quantities can be utilized to compare two or more competing models at the same quantile. The model with a higher (lower) log marginal likelihood (DIC) provides a better model fit.

Value

Returns a bqrorOR1 object with components:

`summary:` summary of the MCMC draws.

`postMeanbeta:` posterior mean of β from the complete MCMC run.

`postMeandelta:` posterior mean of δ from the complete MCMC run.

`postStdbeta:` posterior standard deviation of β from the complete MCMC run.

`postStddelta:` posterior standard deviation of δ from the complete MCMC run.

`gammacp:` vector of cut points including (Inf, -Inf).

`catprob:` probability for each category, calculated at the posterior mean and the mean of x.

`acceptancerate:` acceptance rate of the proposed draws of δ.

`dicQuant:` all quantities of DIC.

`logMargLike:` an estimate of the log marginal likelihood.

`ineffactor:` inefficiency factor for each component of β and δ.

`betadraws:` dataframe of the β draws from the complete MCMC run, size is $(kxnsim)$.

`deltadraws:` dataframe of the δ draws from the complete MCMC run, size is $((J-2)xnsim)$.

References

Rahman, M. A. (2016). ‘"Bayesian Quantile Regression for Ordinal Models."‘ Bayesian Analysis, 11(1): 1-24. DOI: 10.1214/15-BA939

See Also

rnorm, qnorm, Gibbs sampler, Metropolis-Hastings algorithm

Examples

```
set.seed(101)
data("data25j4")
y <- data25j4$y
xMat <- data25j4$x
k <- dim(xMat)[2]
J <- dim(as.array(unique(y)))[1]
b0 <- array(rep(0, k), dim = c(k, 1))
B0 <- 10*diag(k)
d0 <- array(0, dim = c(J-2, 1))
D0 <- 0.25*diag(J - 2)
output <- quantregOR1(y = y, x = xMat, b0 ,B0, d0, D0,
burn = 10, mcmc = 40, p = 0.25, tune = 1, accutoff = 0.5, maxlags = 400, verbose = TRUE)

# Summary of MCMC draws:


#             Post Mean  Post Std   Upper Credible Lower Credible Inef Factor
# beta_1       -2.6202   0.3588        -2.0560        -3.3243       1.1008
# beta_2        3.1670   0.5894         4.1713         2.1423       3.0024
# beta_3        4.2800   0.9141         5.7142         2.8625       2.8534
# delta_1       0.2188   0.4043         0.6541        -0.4384       3.6507
# delta_2       0.4567   0.3055         0.7518        -0.2234       3.1784

# MH acceptance rate: 50%
# Log of Marginal Likelihood: -559.73
# DIC: 1133.11
```

quantregOR2 *Bayesian quantile regression in the OR2 model*

Description

This function estimates Bayesian quantile regression in the OR2 model (ordinal quantile model with exactly 3 outcomes) and reports the posterior mean, posterior standard deviation, 95 percent posterior credible intervals and inefficiency factor of (β, σ). The output also displays the log of marginal likelihood and the DIC.

Usage

```
quantregOR2(y, x, b0, B0 , n0, d0, gammacp2, burn, mcmc, p, accutoff, maxlags, verbose)
```

Arguments

y	observed ordinal outcomes, column vector of size $(nx1)$.
x	covariate matrix of size (nxk) including a column of ones with or without column names.
b0	prior mean for β.
B0	prior covariance matrix for β.
n0	prior shape parameter of the inverse-gamma distribution for σ, default is 5.
d0	prior scale parameter of the inverse-gamma distribution for σ, default is 8.
gammacp2	one and only cut-point other than 0, default is 3.
burn	number of burn-in MCMC iterations.
mcmc	number of MCMC iterations, post burn-in.
p	quantile level or skewness parameter, p in (0,1).
accutoff	autocorrelation cut-off to identify the number of lags and form batches to compute the inefficiency factor, default is 0.05.
maxlags	maximum lag at which to calculate the acf in inefficiency factor calculation, default is 400.
verbose	whether to print the final output and provide additional information or not, default is TRUE.

Details

This function estimates Bayesian quantile regression for the OR2 model using a Gibbs sampling procedure. The function takes the prior distributions and other information as inputs and then iteratively samples β, σ, latent weight ν, and latent variable z from their respective conditional distributions.

The function also provides the logarithm of marginal likelihood and the DIC. These quantities can be utilized to compare two or more competing models at the same quantile. The model with a higher (lower) log marginal likelihood (DIC) provides a better model fit.

Value

Returns a bqrorOR2 object with components

`summary:` summary of the MCMC draws.

`postMeanbeta:` posterior mean of β from the complete Gibbs run.

`postMeansigma:` posterior mean of σ from the complete Gibbs run.

`postStdbeta:` posterior standard deviation of β from the complete Gibbs run.

`postStdsigma:` posterior standard deviation of σ from the complete Gibbs run.

`dicQuant:` all quantities of DIC.

`logMargLike:` an estimate of log marginal likelihood.

`ineffactor:` inefficiency factor for each component of β and σ.

`betadraws:` dataframe of the β draws from the complete Gibbs run, size is $(kxnsim)$.

`sigmadraws:` dataframe of the σ draws from the complete Gibbs run, size is $(1xnsim)$.

References

Rahman, M. A. (2016). ‘"Bayesian Quantile Regression for Ordinal Models."‘ Bayesian Analysis, 11(1): 1-24. DOI: 10.1214/15-BA939

See Also

rnorm, qnorm, Gibbs sampling

Examples

```
set.seed(101)
data("data25j3")
y <- data25j3$y
xMat <- data25j3$x
k <- dim(xMat)[2]
b0 <- array(rep(0, k), dim = c(k, 1))
B0 <- 10*diag(k)
n0 <- 5
d0 <- 8
output <- quantregOR2(y = y, x = xMat, b0, B0, n0, d0, gammacp2 = 3,
burn = 10, mcmc = 40, p = 0.25, accutoff = 0.5, maxlags = 400, verbose = TRUE)

# Summary of MCMC draws :

#             Post Mean Post Std Upper Credible Lower Credible Inef Factor
#    beta_1   -4.5185   0.9837        -3.1726        -6.2000     1.5686
#    beta_2    6.1825   0.9166         7.6179         4.8619     1.5240
#    beta_3    5.2984   0.9653         6.9954         4.1619     1.4807
#    sigma     1.0879   0.2073         1.5670         0.8436     2.4228

# Log of Marginal Likelihood: -404.57
# DIC: 801.82
```

rndald *Generates random numbers from an AL distribution*

Description

This function generates a vector of random numbers from an AL distribution at quantile p.

Usage

```
rndald(sigma, p, n)
```

Arguments

`sigma`	scale factor, a scalar value.
`p`	quantile or skewness parameter, p in (0,1).
`n`	number of observations

Details

Generates a vector of random numbers from an AL distribution as a mixture of normal‘–‘exponential distributions.

Value

Returns a vector $(nx1)$ of random numbers from an AL(0, σ, p)

References

Kozumi, H., and Kobayashi, G. (2011). ‘"Gibbs Sampling Methods for Bayesian Quantile Regression."‘ Journal of Statistical Computation and Simulation, 81(11): 1565‘-‘1578. DOI: 10.1080/00949655.2010.496117

Yu, K., and Zhang, J. (2005). ‘"A Three-Parameter Asymmetric Laplace Distribution."‘ Communications in Statistics - Theory and Methods, 34(9-10), 1867‘-‘1879. DOI: 10.1080/03610920500199018

See Also

asymmetric Laplace distribution

Examples

```
set.seed(101)
sigma <- 2.503306
p <- 0.25
n <- 1
output <- rndald(sigma, p, n)

# output
#   1.07328
```

summary.bqrorOR1 *Extractor function for summary*

Description

This function extracts the summary from the bqrorOR1 object

Usage

```
## S3 method for class 'bqrorOR1'
summary(object, digits, ...)
```

Arguments

object	bqrorOR1 object from which the summary is extracted.
digits	controls the number of digits after the decimal.
...	extra arguments

Details

This function is an extractor function for the summary

Value

the summarized information object

Examples

```
set.seed(101)
data("data25j4")
y <- data25j4$y
xMat <- data25j4$x
k <- dim(xMat)[2]
J <- dim(as.array(unique(y)))[1]
b0 <- array(rep(0, k), dim = c(k, 1))
B0 <- 10*diag(k)
d0 <- array(0, dim = c(J-2, 1))
D0 <- 0.25*diag(J - 2)
output <- quantregOR1(y = y, x = xMat, b0, B0, d0, D0,
burn = 10, mcmc = 40, p = 0.25, tune = 1, accutoff = 0.5, maxlags = 400, verbose = FALSE)
summary(output, 4)

 #            Post Mean  Post Std   Upper Credible Lower Credible Inef Factor
 # beta_1       -2.6202   0.3588        -2.0560        -3.3243       1.1008
 # beta_2        3.1670   0.5894         4.1713         2.1423       3.0024
 # beta_3        4.2800   0.9141         5.7142         2.8625       2.8534
 # delta_1       0.2188   0.4043         0.6541        -0.4384       3.6507
 # delta_2       0.4567   0.3055         0.7518        -0.2234       3.1784
```

`summary.bqrorOR2` *Extractor function for summary*

Description

This function extracts the summary from the bqrorOR2 object

Usage

```
## S3 method for class 'bqrorOR2'
summary(object, digits, ...)
```

Arguments

`object`	bqrorOR2 object from which the summary is extracted.
`digits`	controls the number of digits after the decimal
`...`	extra arguments

Details

This function is an extractor function for the summary

Value

the summarized information object

Examples

```
set.seed(101)
data("data25j3")
y <- data25j3$y
xMat <- data25j3$x
k <- dim(xMat)[2]
b0 <- array(rep(0, k), dim = c(k, 1))
B0 <- 10*diag(k)
n0 <- 5
d0 <- 8
output <- quantregOR2(y = y, x = xMat, b0, B0, n0, d0, gammacp2 = 3,
burn = 10, mcmc = 40, p = 0.25, accutoff = 0.5, maxlags = 400, FALSE)
summary(output, 4)

#            Post Mean Post Std Upper Credible Lower Credible Inef Factor
#    beta_1   -4.5185   0.9837        -3.1726        -6.2000     1.5686
#    beta_2    6.1825   0.9166         7.6179         4.8619     1.5240
#    beta_3    5.2984   0.9653         6.9954         4.1619     1.4807
#    sigma     1.0879   0.2073         1.5670         0.8436     2.4228
```

Index