Package: bayesQR (via r-universe)

October 29, 2026

Version 2.4

Date 2023-09-08

Title Bayesian Quantile Regression

Author Dries F. Benoit, Rahim Al-Hamzawi, Keming Yu, Dirk Van den Poel

Maintainer Dries F. Benoit <Dries.Benoit@UGent.be>

Depends R (>= 4.2), graphics, methods, stats, utils

Description Bayesian quantile regression using the asymmetric Laplace distribution, both continuous as well as binary dependent variables are supported. The package consists of implementations of the methods of Yu & Moyeed (2001) <doi:10.1016/S0167-7152(01)00124-9>, Benoit & Van den Poel (2012) <doi:10.1002/jae.1216> and Al-Hamzawi, Yu & Benoit (2012) <doi:10.1177/1471082X1101200304>. To speed up the calculations, the Markov Chain Monte Carlo core of all algorithms is programmed in Fortran and called from R.

License GPL (>= 2)

NeedsCompilation yes

Repository https://cran.r-universe.dev

Date/Publication 2023-09-08 18:30:31 UTC

RemoteUrl https://github.com/cran/bayesQR

RemoteRef HEAD

RemoteSha 9db6e4b7c7a0459da60f2ed2499ed530d5be5704

Contents

bayesQR 2
Churn 5
plot.bayesQR 6
predict.bayesQR 7
print.bayesQR 9
print.bayesQR.summary 10

prior 11
Prostate 13
summary.bayesQR 14

Index **16**

`bayesQR` *Bayesian quantile regression*

Description

`bayesQR` implements a Bayesian method for estimating quantile regression models (see references). To improve the speed of the routine, the Markov Chain Monte Carlo (MCMC) part of the algorithm is programmed in Fortran and is called from within the R function `bayesQR`.

Usage

```
bayesQR(formula, data, quantile, alasso, normal.approx, ndraw, keep, prior)
```

Arguments

`formula`	a symbolic description of the model to be fit.
`data`	an optional data frame containing the variables in the model.
`quantile`	numerical scalar or vector containing quantile(s) of interest (default=0.5).
`alasso`	logical flag for adaptive lasso variable selection (default=FALSE).
`normal.approx`	logical flag for normal approximation of posterior distribution (default=TRUE).
`ndraw`	number of MCMC draws.
`keep`	thinning parameter, i.e. keep every keepth draw (default=1).
`prior`	an S3 object of class `"prior"`. If omitted, a diffuse prior will be assumed (see `prior`).

Details

The function `bayesQR` can estimate four types of models, depending on whether the dependent variable is continuous or binary and whether adaptive lasso variable selection is used.

- Continuous dependent variable without adaptive lasso variable selection:

Model:	y = X*beta* + e
	e ~ ALD(location=0, scale=sigma, quantile=p)
Priors:	*beta ~ N(beta0, V0)*
	sigma ~ invGamma(shape0,scale0)

- Continuous dependent variable with adaptive lasso variable selection:

Model: y = X*beta* + e
e ~ ALD(location=0, scale=sigma, quantile=p)
Priors: *beta ~ ALD(location=0, scale=sigma/lambda, p=0.5)*
sigma ~ InvGamma(shape=sigma_shape, scale=sigma_scale)
(lambda/sigma)^2 ~ Gamma(shape=etasq_shape, scale=etasq_scale)

- Binary dependent variable without adaptive lasso variable selection:

Model: y* = X*beta* + e
e ~ ALD(location=0, scale=1, quantile=p)
y = 1, if (y > 0); y = 0, otherwise*
Priors: *beta ~ N(beta0, V0)*

- Binary dependent variable with adaptive lasso variable selection:

Model: y* = X*beta* + e
e ~ ALD(location=0, scale=1, quantile=p)
y = 1, if (y > 0); y = 0, otherwise*
Priors: *beta ~ ALD(location=0, scale=1/lambda, p=0.5)*
lambda^2 ~ Gamma(shape=lambdasq_shape, scale=lambdasq_scale)

Value

An object of class `bayesQR`, basically a list of lists. For every estimated quantile a list is created containing the following elements:

`method`	a string containing the method that was used, i.e. indicating whether the dependent variable was continuous or binary and whether adaptive lasso variable selection was used.
`normal.approx`	logical flag for normal approximation of posterior distribution.
`quantile`	the quantile that was estimated.
`names`	character vector containing the names of the independent variables in the model.
`betadraw`	ndraw/keep x nvar(X) matrix of beta draws.
`sigmadraw`	ndraw/keep vector of sigma draws (only in case of continuous dependent variable).

Author(s)

Dries F. Benoit, Rahim Al-Hamzawi, Keming Yu and Dirk Van den Poel

References

- Paper about the bayesQR package:
Benoit, D.F and Van den Poel, D. (2017). bayesQR: A Bayesian Approach to Quantile Regression, *Journal of Statistical Software*, 76(7), 1-32. <doi:10.18637/jss.v076.i07>

- Continuous dependent variable without adaptive lasso variable selection:
  Yu, K. and Moyeed, R. (2001). Bayesian quantile regression, *Statistics and Probability Letters*, 54, 437-447. <doi:10.1016/S0167-7152(01)00124-9>
  Also see:
  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>

- Continuous dependent variable with adaptive lasso variable selection:
  Alhamzawi, R., Yu, K. and Benoit, D.F. (2012). Bayesian adaptive LASSO quantile regression, *Statistical Modelling*, 12(3), 279-297. <doi:10.1177/1471082X1101200304>
  Also see:
  Li, Q., Xi, R. and Lin, N. (2010). Bayesian Regularized Quantile Regression. *Bayesian Analysis*, 5, 1-24. <doi:10.1214/10-BA521>

- Binary dependent variable without adaptive lasso variable selection:
  Benoit, D.F. and Van den Poel, D. (2012). Binary quantile regression: A Bayesian approach based on the asymmetric Laplace distribution, *Journal of Applied Econometrics*, 27(7), 1174-1188. <doi:10.1002/jae.1216>

- Binary dependent variable with adaptive lasso variable selection:
  Al-Hamzawi, R., Benoit, D.F. and Yu, K. (2012). Bayesian lasso binary quantile regression. *Computational Statistics*, 28(6), 2861-2873. <doi:10.1007/s00180-013-0439-0>

Examples

```
# Simulate data from heteroskedastic regression
set.seed(66)
n <- 200
X <- runif(n=n,min=0,max=10)
X <- X
y <- 1 + 2*X + rnorm(n=n, mean=0, sd=.6*X)

# Estimate series of quantile regressions with adaptive lasso
# NOTE: to limit execution time of the example, ndraw is set
#       to a very low value. Set value to 5000 for a better
#       approximation of the posterior distirubtion.
out <- bayesQR(y~X, quantile=c(.05,.25,.5,.75,.95), alasso=TRUE, ndraw=500)

# Initiate plot
## Plot datapoints
plot(X, y, main="", cex=.6, xlab="X")

## Add quantile regression lines to the plot (exclude first 500 burn-in draws)
sum <- summary(out, burnin=50)
for (i in 1:length(sum)){
  abline(a=sum[[i]]$betadraw[1,1],b=sum[[i]]$betadraw[2,1],lty=i,col=i)
}

# Estimate and plot OLS model
```

```
outOLS = lm(y~X)
abline(outOLS,lty=1,lwd=2,col=6)

# Add legend to plot
legend(x=0,y=max(y),legend=c(.05,.25,.50,.75,.95,"OLS"),lty=c(1,2,3,4,5,1),
       lwd=c(1,1,1,1,1,2),col=c(1:6),title="Quantile")
```

Churn *Customer Churn Data*

Description

This dataset is a stratified random sample from all active customers (at the end of June 2006) of a European financial services company. The dependent variable in this dataset is the churn behavior of the customers in the period from July 1st until December 31th 2006. Here a churned customer is defined as someone who closed all his/her bank accounts with the company. Note that all predictor variables are standardized. This dataset is a small subset of the dataset used in Benoit and Van den Poel (2013). The dataset is structured as a dataframe with 400 observations and 5 variables.

Usage

```
data("Churn")
```

Format

The data frame has the following components:

- **churn** : churn (yes/no)
- **gender** : gender of the customer (male = 1)
- **Social_Class_Score** : social class of the customer
- **lor** : length of relationship with the customer
- **recency** : number of days since last purchase

Source

Benoit, D.F. and Van den Poel, D. (2013). Quantile regression for database marketing: methods and applications. In: Coussement, K., De Bock, K.W. and Neslin, S.A. (eds.). *Advanced database marketing: Innovative methodologies and applications for managing customer relationships*. Gower Publishing: London (UK). <doi:10.4324/9781315565682>

References

Benoit, D.F. and Van den Poel, D. (2013). Quantile regression for database marketing: methods and applications. In: Coussement, K., De Bock, K.W. and Neslin, S.A. (eds.). *Advanced database marketing: Innovative methodologies and applications for managing customer relationships*. Gower Publishing: London (UK). <doi:10.4324/9781315565682>

Benoit, D.F and Van den Poel, D. (2017). bayesQR: A Bayesian Approach to Quantile Regression, *Journal of Statistical Software*, 76(7), 1-32. <doi:10.18637/jss.v076.i07>

plot.bayesQR *Produce quantile plots or traceplots with* plot.bayesQR

Description

plot.bayesQR is an S3 method that produces quantile plots, traceplots or posterior histograms based on the estimates obtained by the bayesQR function. For quantile plots, note that the more quantiles are estimated with bayesQR, the more detailed the quantile plot will be.

Usage

```
  ## S3 method for class 'bayesQR'
plot(x, var=NULL, quantile=NULL, burnin=0, credint=c(0.025,0.975), plottype=NULL,
main=NULL, xlab=NULL, ylab=NULL, xlim=NULL, ylim=NULL, ...)
```

Arguments

x	an output object of the bayesQR function, i.e. an S3 object of class bayesQR.
var	vector containing the index or name of the variable(s) that has to be plotted (default=all variables).
quantile	vector containing the quantile(s) that has to be plotted (default=all quantiles).
burnin	the number of burnin draws that should be discared (default=0, meaning all draws are included).
credint	the width of the credible interval (default=c(0.025, 0.975)).
plottype	should be 'quantile', 'trace' or 'hist'.
main	Main title of the plot (default="").
xlab	Label of the x-axis; if omitted, the value chosen based on the input data.
ylab	Label of the y-axis; if omitted, the value chosen based on the input data.
xlim	Plot region of the x-axis; if omitted, the value chosen based on the input data.
ylim	Plot region of the y-axis; if omitted, the value chosen based on the input data.
...	additional arguments that are passed to the generic plot function

Details

A quantile plot shows how the value of the regression parameter changes over a range of quantiles together with the associated credible interval. When the normal approximation was requested, the credible regions represent the adjusted credible intervals. Note that the more quantiles are estimated, the more detailed the quantile plot will be. The minimum number of quantiles to plot is 2. A posterior histogram provides a graphical representation of the marginal posterior distribution of the regression parameters. When the normal approximation was requested, the histogram will be overlaid with the adjusted credible intervals. A traceplot gives the evolution of the MCMC chain and can be used as graphical check of convergence. Note that more formal checks of convergence exist (see, for example, the **coda** package).

Value

A (series of) quantile plot(s) or a (series of) traceplot(s).

Author(s)

Dries F. Benoit

Examples

```
# Simulate data from heteroskedastic regression
set.seed(66)
n <- 200
X <- runif(n=n,min=0,max=10)
X <- X
y <- 1 + 2*X + rnorm(n=n, mean=0, sd=.6*X)

# Analyze 5 quantiles using default prior
# NOTE: to limit execution time of the example, ndraw is set
#       to a very low value. Set value to 5000 for a better
#       approximation of the posterior distirubtion.
out <- bayesQR(y ~ X, quantile=c(.05,.25,.5,.75,.95), ndraw=500)

# Check traceplot of first variable of .75 quantile regression
plot(out, var=1, quantile=.75, plottype="trace")

# Check posterior histogram of first variable of .5 quantile regression
plot(out, var=1, quantile=.5, plottype="hist")

# Create default quantile plot of first variable
plot(out, var=1, plottype="quantile")

# Create quantile plot of second variable with 90% credible interval
plot(out, var="X", credint=c(.05, .95), plottype="quantile", main="This is an example")
```

---

predict.bayesQR *Calculate predicted probabilities for binary quantile regression*

Description

`predict.bayesQR` is an S3 method that calculates predicted probabilities for binary quantile regression, both with or without adaptive lasso.

Usage

```
  ## S3 method for class 'bayesQR'
predict(object, X, burnin=0, ...)
```

Arguments

object	an output object of the bayesQR function, S3 class bayesQR, with binary dependent variable, with or without adaptive lasso. At leas 9 estimated quantiles are required.
X	matrix of predictors (should be of the same type as the variables used to estimate the model).
burnin	the number of burnin draws that should be discared (default=0, meaning all draws are included).
...	additional parameters passed to the generic predict function.

Details

predict.bayesQR is an S3 method that calculates the predicted probabilities based on a matrix of predictors *X* and an object containing the parameter estimates of a sequence of binary quantile regressions. The rationale behind the approach is described in Kordas (2006) and applied in Migueis, V.L., Benoit, D.F. and Van den Poel, D. (2012). Note that the more quantiles are estimated, the more fine-grained the predicted probabilities will be.

Value

A vector containing the predicted probabilities.

Author(s)

Dries F. Benoit

References

Kordas, G. (2006). Binary regression quantiles, *Journal of Applied Econometrics*, **21**(3), 387-407.

Migueis, V.L., Benoit, D.F. and Van den Poel, D. (2012). Enhanced decision support in credit scoring using Bayesian binary quantile regression, *Journal of the Operational Research Society*, (*in press*).

Examples

```
# Simulate data from binary regression model
set.seed(1234)
n <- 200
X <- matrix(runif(n=n*2, min=-5, max=5),ncol=2)
ystar <- cbind(1,X)%*% c(1,1.5,-.5) + rnorm(n=n, mean=0, sd=abs(2*X[,1]))
y <- as.numeric(ystar>0)

# Estimate a sequence of binary quantile regression models
# NOTE: to limit execution time of the example, ndraw is set
#       to a very low value. Set value to 4000 for a better
#       approximation of the posterior distirubtion.
out <- bayesQR(y ~ X, quantile=seq(.1,.9,.1), ndraw=400)
```

```
# Calculate predicted probabilities
pred <- predict(object=out, X=cbind(1,X), burnin=20)

# Make histogram of predicted probabilities
hist(pred,breaks=10)

# Calculate Percentage Correclty Classified (PCC)
mean(y==as.numeric(pred>.5))

# Compare with logit model
mylogit <- glm(y ~ X, family=binomial(logit))

# Make histogram of predicted probabilities
hist(mylogit$fit,breaks=10)

# Calculate Percentage Correclty Classified (PCC)
mean(y==as.numeric(mylogit$fit>.5))
```

`print.bayesQR` *Prints the contents of* `bayesQR` *object to the console*

Description

`print.bayesQR` is an S3 method that prints the content of an S3 object of class `bayesQR` to the console.

Usage

```
  ## S3 method for class 'bayesQR'
print(x, digits=3, ...)
```

Arguments

`x`	an output object of the `bayesQR` function, i.e. an S3 object of class `bayesQR`.
`digits`	the number of printed digits of the estimates (default=3).
`...`	additional arguments that are passed to the generic `print` function

Value

Formatted output of a `bayesQR` object.

Author(s)

Dries F. Benoit

Examples

```
# Simulate data from heteroskedastic regression
set.seed(666)
n <- 200
X <- runif(n=n,min=0,max=10)
y <- 1 + 2*X + rnorm(n=n, mean=0, sd=.6*X)

# Analyze 0.5 quantile using default prior
out <- bayesQR(y ~ X, ndraw=5000)

# Print the bayesQR object
print(out)
```

---

print.bayesQR.summary *Prints the contents of* bayesQR.summary *object to the console*

Description

print.bayesQR.summary is an S3 method that prints the content of an S3 object of class bayesQR.summary to the console.

Usage

```
  ## S3 method for class 'bayesQR.summary'
print(x, digits=3, ...)
```

Arguments

x	an output object of the summary.bayesQR function, i.e. an S3 object of class bayesQR.summary.
digits	the number of printed digits of the estimates (default=3).
...	additional arguments that are passed to the generic print function

Value

Formatted output of a bayesQR.summary object.

Author(s)

Dries F. Benoit

Examples

```
# Simulate data from heteroskedastic regression
set.seed(666)
n <- 200
X <- runif(n=n,min=0,max=10)
y <- 1 + 2*X + rnorm(n=n, mean=0, sd=.6*X)

# Analyze 0.5 quantile using default prior and adaptive lasso
out <- bayesQR(y ~ X, alasso=TRUE, ndraw=5000)

# Return Bayes estimates and credible intervals
sum <- summary(out, burnin=1000)

# Print the bayesQR.summary object
sum
```

---

prior *Create prior for Bayesian quantile regression*

Description

`prior` creates an S3 object of class `bayesQR.prior` that contains all necessary prior information to estimate a Bayesian quantile regression model.

Usage

```
prior(formula, data, alasso, ...)
```

Arguments

`formula`	a symbolic description of the model to be fit.
`data`	an optional data frame containing the variables in the model.
`alasso`	logical flag for adaptive lasso variable selection (default=FALSE).
`...`	the prior parameters that are dependent on the method that is used. If omitted, a standard diffuse prior will be used (see details section).

Details

The function `prior` builds the prior for four types of models, depending on whether the dependent variable is continuous or binary and whether adaptive lasso variable selection is used. Every non-specified prior parameter will get the default value.

Continuous dependent variable without adaptive lasso variable selection:

- **beta0** : nvar(X) x 1 vector of prior means (default: 0)
- **V0** : nvar(X) x nvar(X) prior covariance matrix (default: 100*diag(ncol(X)))
- **shape0** : shape parameter for inverse Gamma prior for sigma (default: 0.01)

- **scale0** : scale parameter for inverse Gamma prior for sigma (default: 0.01)

Continuous dependent variable with adaptive lasso variable selection:

- **sigma_shape** : shape parameter for the inverse gamma prior on sigma (default: 0.01)
- **sigma_scale** : scale parameter for the inverse gamma prior on sigma (default: 0.01)
- **etasq_shape** : shape parameter for the gamma prior on etasq (default: 0.01)
- **etasq_scale** : scale parameter for the gamma prior on etasq (default: 0.01)

Binary dependent variable without adaptive lasso variable selection:

- **beta0** : nvar(X) x 1 vector of prior means (default: 0)
- **V0** : nvar(X) x nvar(X) prior covariance matrix (default: 100*diag(ncol(X)))

Binary dependent variable with adaptive lasso variable selection:

- **lambdasq_shape** : shape parameter for the gamma prior on lambdasq (default: 0.01)
- **lambdasq_scale** : scale parameter for the gamma prior on lambdasq (default: 0.01)

Value

An object of class `bayesQR`, basically a list containing the following elements:

`method`	a string containing the method that was used, i.e. indicating whether the dependent variable was continuous or binary and whether adaptive lasso variable selection was used.
`...`	the prior parameters that are dependent on the method that is used (see details section).

Author(s)

Dries F. Benoit

Examples

```
# Load the Prostate cancer dataset
data(Prostate)

# Create informative prior object
prior <- prior(lpsa~., data=Prostate, beta0=rep(5,9), V0=diag(9))

# Investigate structure of bayesQR.prior object
str(prior)

# Estimate the model parameters with informative prior
out <- bayesQR(lpsa~., data=Prostate, prior=prior, ndraw=5000)

# Print results
summary(out)
```

`Prostate` *Prostate Cancer Data*

Description

These data come from a study that examined the correlation between the level of prostate specific antigen and a number of clinical measures in men who were about to receive a radical prostatectomy. It is a data frame with 97 rows and 9 columns.

Usage

```
data("Prostate")
```

Format

The data frame has the following components:

- **lcavol** : log(cancer volume)
- **lweight** : log(prostate weight)
- **age** : age
- **lbph** : log(benign prostatic hyperplasia amount)
- **svi** : seminal vesicle invasion
- **lcp** : log(capsular penetration)
- **gleason** : Gleason score
- **pgg45** : percentage Gleason scores 4 or 5
- **lpsa** : log(prostate specific antigen)

Source

Stamey, T.A., Kabalin, J.N., McNeal, J.E., Johnstone, I.M., Freiha, F., Redwine, E.A. and Yang, N. (1989). Prostate specific antigen in the diagnosis and treatment of adenocarcinoma of the prostate: II. radical prostatectomy treated patients, *Journal of Urology* **141**(5), 1076–1083.

References

Stamey, T.A., Kabalin, J.N., McNeal, J.E., Johnstone, I.M., Freiha, F., Redwine, E.A. and Yang, N. (1989). Prostate specific antigen in the diagnosis and treatment of adenocarcinoma of the prostate: II. radical prostatectomy treated patients, *Journal of Urology*, **141**(5), 1076–1083.

This dataset also appears in the R-package: *lasso2*.
Lokhorst, J., Venables, B. and Turlach, B. (2011). lasso2: L1 contstrained estimation aka 'lasso', published on R CRAN website: *http://cran.r-project.org/web/packages/lasso2/index.html*.

summary.bayesQR *Summarize the output of the* bayesQR *function*

Description

summary.bayesQR is an S3 method that summarizes the output of the bayesQR function in an object of class bayesQR.summary. For every estimated beta and sigma, this object contains the Bayes estimate and the posterior credible interval is calculated. The object also contains other relevant information about the estimation procedure, such as the quantile, the variable names, etc.

Usage

```
  ## S3 method for class 'bayesQR'
summary(object, burnin=0, credint=c(.025,.975), quantile=NULL, ...)
```

Arguments

object	an output object of the bayesQR function, i.e. an S3 object of class bayesQR.
burnin	the number of burnin draws that should be discarded (default=0, meaning all draws are included).
credint	the width of the credible interval (default=c(0.025, 0.975)).
quantile	the quantile(s) of the quantile regressions that have to be summarized (default: all estimated quantiles in QRobj).
...	additional arguments passed to the generic summary function.

Value

An object of class bayesQR.summary, basically a list including elements:

method	a string containing the method that was used, i.e. indicating whether the dependent variable was continuous or binary and whether adaptive lasso variable selection was used.
normal.approx	logical flag for normal approximation of posterior distribution.
quantile	the quantile that was estimated.
names	character vector containing the names of the independent variables in the model.
burnin	the number of burnin draws that were discarded.
retained	the number of draws that were retained and used to calculate the summary statistics.
credint	the width of the credible interval.
betadraw	the Bayes estimate, credible interval and, if normal.approx=TRUE, the adjusted credible intervals of the beta draws.
sigmadraw	the Bayes estimate and credible interval of the sigma draws.

Author(s)

Dries F. Benoit

Examples

```
# Load the Prostate cancer dataset
data(Churn)

# Estimate the model parameters with default prior
out <- bayesQR(churn~gender+recency, data=Churn, ndraw=2000)

# Return Bayes estimates and credible intervals
sum <- summary(out, burnin=1000)

# Inspect structure of bayesQR.summary object
str(sum)

# Print bayesQR.summary object
sum
```

Index

∗ **datasets**
    Churn, 5
    Prostate, 13

bayesQR, 2

Churn, 5

plot.bayesQR, 6
predict.bayesQR, 7
print.bayesQR, 9
print.bayesQR.summary, 10
prior, 2, 11
Prostate, 13

summary.bayesQR, 14