

Package ‘BeQut’

January 20, 2025

Title Bayesian Estimation for Quantile Regression Mixed Models

Version 0.1.0

Maintainer Antoine Barbieri <antoine.barbieri@u-bordeaux.fr>

Description Using a Bayesian estimation procedure, this package fits linear quantile regression models such as linear quantile models, linear quantile mixed models, quantile regression joint models for time-to-event and longitudinal data. The estimation procedure is based on the asymmetric Laplace distribution and the 'JAGS' software is used to get posterior samples (Yang, Luo, DeSantis (2019) <[doi:10.1177/0962280218784757](https://doi.org/10.1177/0962280218784757)>).

Depends R (>= 3.5.0), survival

Imports jagsUI, lqmm, MASS

SystemRequirements JAGS 4.x.y

License GPL (>= 2.0)

Encoding UTF-8

RoxygenNote 7.2.3

NeedsCompilation no

Author Antoine Barbieri [aut, cre],
Hélène Jacqmin-Gadda [aut]

Repository CRAN

Date/Publication 2023-11-09 10:30:06 UTC

Contents

dataLong	2
deviance	2
lqm	4
lqmm	6
qtjm	8
wave	11
Index	12

`dataLong`*dataLong*

Description

'dataLong' is a dataset simulated from a joint model for longitudinal and time-to-event data. This dataset is used to illustrate both 'lqmm' and 'qrjm' functions.

Usage

```
data(dataLong)
```

Format

A data.frame with 1562 observations from 300 subjects. The columns are:

ID integer: number for patient identification.

visit numeric: measurement times for the repeated blood pressure measurements.

y numeric: longitudinal measurements.

time numeric: time to event (or censoring).

event integer: event indicator. Coded as 0 = right-censoring, and 1 = event.

X1 integer: time-independent binary explanatory variable.

X2 numeric: time-independent continuous explanatory variable.

Examples

```
data(dataLong)
```

`deviance`*deviance returns the deviance based on the conditional likelihood associated with the survival part.*

Description

deviance returns the deviance based on the conditional likelihood associated with the survival part.

Usage

```
deviance(object, M = 1000, conditional = "survival", verbose = TRUE)
```

Arguments

object	an object inheriting from class 'Bqrm'.
M	an integer indicating the number of draws used for the approximation of the integral with respect to random effects, M=1000 by default.
conditional	is "survival" by default (only this one is implemented until now).
verbose	A logical indicating if information about method's progress (included progress bars for each step) must be printed (default to TRUE). Adds a small extra over-load.

Value

An object which is a list with the following elements:

deviance Numerical object returning the deviance

likelihood (Conditional) likelihood

sims.list list of individual quantities as likelihood, draws of random effects, hazard and survival functions

control list of arguments giving details about the deviance

Author(s)

Antoine Barbieri and Baptiste Courrèges

Examples

```
#---- load data
data(dataLong)

#---- Fit quantile regression joint model for the median
qrjm_50 <- qrjm(formFixed = y ~ visit,
               formRandom = ~ visit,
               formGroup = ~ ID,
               formSurv = survival::Surv(time, event) ~ X1 + X2,
               survMod = "weibull",
               param = "value",
               timeVar = "visit",
               data = dataLong,
               save_va = TRUE,
               parallel = FALSE,
               tau = 0.5)

deviance(qrjm_50, M=200)
```

lqm

lqm fits linear quantile regression model

Description

Function using 'JAGS' software to estimate the linear quantile regression model assuming asymmetric Laplace distribution for residual error.

Usage

```
lqm(
  formula,
  data,
  tau = 0.5,
  n.chains = 3,
  n.iter = 10000,
  n.burnin = 5000,
  n.thin = 1,
  n.adapt = NULL,
  save_jagsUI = TRUE,
  parallel = FALSE
)
```

Arguments

formula	formula for the quantile regression including response variable
data	dataset of observed variables
tau	the quantile(s) to be estimated. This must be a number between 0 and 1, otherwise the execution is stopped. If more than one quantile is specified, rounding off to the 4th decimal must give non-duplicated values of tau, otherwise the execution is stopped.
n.chains	the number of parallel chains for the model; default is 1.
n.iter	integer specifying the total number of iterations; default is 10000
n.burnin	integer specifying how many of n.iter to discard as burn-in ; default is 5000
n.thin	integer specifying the thinning of the chains; default is 1
n.adapt	integer specifying the number of iterations to use for adaptation; default is NULL
save_jagsUI	If TRUE (is TRUE by default), the output of jagsUI package is return by the function
parallel	see jagsUI::jags() function

Value

A Blqm object which is a list with the following elements:

mean list of posterior mean for each parameter

median list of posterior median for each parameter

modes list of posterior mode for each parameter

StErr list of standard error for each parameter

StDev list of standard deviation for each parameter

Rhat Gelman and Rubin diagnostic for all parameters

ICs list of the credibility interval at 0.95 for each parameters excepted for covariance parameters in covariance matrix of random effects. Otherwise, use save_jagsUI=TRUE to have the associated quantiles.

data data included in argument

sims.list list of the MCMC chains of the parameters and random effects

control list of arguments giving details about the estimation

W list including both posterior mean and posterior standard deviation of subject-specific random variable W

out_jagsUI only if save_jagsUI=TRUE in argument: list including posterior mean, median, quantiles (2.5%, 25%, 50%, 75%, 97.5%), standard deviation for each parameter and each random effect. Moreover, this list also returns the MCMC draws, the Gelman and Rubin diagnostics (see output of jagsUI objects)

Author(s)

Antoine Barbieri

Examples

```
#---- Use data
data(wave)

#---- Fit regression model for the first quartile
lqm_025 <- lqm(formula = h110d~vent_vit_moy,
               data = wave,
               n.iter = 1000,
               n.burnin = 500,
               tau = 0.25)

#---- Get the posterior mean of parameters
lqm_025$mean

#---- Visualize the trace for beta parameters
jagsUI::traceplot(lqm_025$out_jagsUI, parameters = "beta" )

#---- Summary of output
summary(lqm_025)
```

lqmm

lqmm fits linear quantile mixed model

Description

Function using 'JAGS' software to estimate the linear quantile mixed model assuming asymmetric Laplace distribution for residual error.

Usage

```
lqmm(
  formFixed,
  formRandom,
  formGroup,
  data,
  tau,
  RE_ind = FALSE,
  n.chains = 3,
  n.iter = 10000,
  n.burnin = 5000,
  n.thin = 1,
  n.adapt = NULL,
  precision = 10,
  save_jagsUI = TRUE,
  parallel = FALSE
)
```

Arguments

formFixed	formula for fixed part of longitudinal submodel with response variable
formRandom	formula for random part of longitudinal submodel without response variable
formGroup	formula specifying the cluster variable (e.g. = ~ subject)
data	dataset of observed variables
tau	the quantile(s) to be estimated. This must be a number between 0 and 1, otherwise the execution is stopped. If more than one quantile is specified, rounding off to the 4th decimal must give non-duplicated values of tau, otherwise the execution is stopped.
RE_ind	Boolean denoting if the random effects are assumed independent ; default is FALSE
n.chains	the number of parallel chains for the model; default is 1.
n.iter	integer specifying the total number of iterations; default is 10000
n.burnin	integer specifying how many of n.iter to discard as burn-in ; default is 5000

n.thin	integer specifying the thinning of the chains; default is 1
n.adapt	integer specifying the number of iterations to use for adaptation; default is NULL
precision	variance by default for vague prior distribution
save_jagsUI	If TRUE (by default), the output of jagsUI package is returned by the function. Warning, if TRUE, the output can be large.
parallel	see jagsUI::jags() function

Value

A Blqmm object is a list with the following elements:

mean list of posterior mean for each parameter
 median list of posterior median for each parameter
 modes list of posterior mode for each parameter
 StErr list of standard error for each parameter
 StDev list of standard deviation for each parameter
 ICs list of the credibility interval at 0.95 for each parameters excepted for covariance parameters in covariance matrix of random effects. Otherwise, use save_jagsUI=TRUE to have the associated quantiles.
 data data included in argument
 sims.list list of the MCMC chains of the parameters and random effects
 control list of arguments giving details about the estimation
 random_effect list for each quantile including both posterior mean and posterior standard deviation of subject-specific random effects
 out_jagsUI only if save_jagsUI=TRUE in argument: list including posterior mean, median, quantiles (2.5%, 25%, 50%, 75%, 97.5%), standart deviation for each parameter and each random effect. Moreover, this list also returns the MCMC draws, the Gelman and Rubin diagnostics (see output of jagsUI objects)

Author(s)

Antoine Barbieri

References

Marco Geraci and Matteo Bottai (2014). *Linear quantile mixed models*. Statistics and Computing, 24(3):461-479. doi: 10.1007/s11222-013-9381-9.

Examples

```
#---- Use dataLong dataset
data(dataLong)

#---- Fit regression model for the first quartile
lqmm_075 <- lqmm(formFixed = y ~ visit,
```

```

        formRandom = ~ visit,
        formGroup = ~ ID,
        data = dataLong,
        tau = 0.75,
        n.iter = 10000,
        n.burnin = 1000)

#---- Get the posterior means
lqmm_075$mean

#---- Visualize the trace for beta parameters
jagsUI::traceplot(lqmm_075$out_jagsUI, parameters = "beta")

#---- Summary of output
summary(lqmm_075)

```

qrjm

qrjm fits quantile regression joint model

Description

Function using 'JAGS' software via jagsUI package to estimate the quantile regression joint model assuming asymmetric Laplace distribution for residual error. Joint modeling concerns longitudinal data and time-to-event

Usage

```

qrjm(
  formFixed,
  formRandom,
  formGroup,
  formSurv,
  survMod = "weibull",
  param = "value",
  timeVar,
  data,
  tau,
  RE_ind = FALSE,
  n.chains = 3,
  n.iter = 10000,
  n.burnin = 5000,
  n.thin = 1,
  n.adapt = 5000,
  precision = 10,
  C = 1000,
  save_jagsUI = TRUE,

```



```

    save_va = FALSE,
    parallel = FALSE
  )

```

Arguments

<code>formFixed</code>	formula for fixed part of longitudinal submodel with response variable
<code>formRandom</code>	formula for random part of longitudinal submodel without response variable
<code>formGroup</code>	formula specifying the cluster variable (e.g. = ~ subject)
<code>formSurv</code>	survival formula as formula in survival package for latency submodel
<code>survMod</code>	specifying the baseline risk function for Cox proportional hazard model (only "weibull" is available until now)
<code>param</code>	shared association including in joint modeling: the classical shared random effects or the current value denoting by "sharedRE" (default) or "value", respectively.
<code>timeVar</code>	string specify the names of time variable (time of repeated measurements)
<code>data</code>	dataset of observed variables
<code>tau</code>	the quantile(s) to be estimated. This must be a number between 0 and 1, otherwise the execution is stopped. If more than one quantile is specified, rounding off to the 4th decimal must give non-duplicated values of tau, otherwise the execution is stopped.
<code>RE_ind</code>	Boolean denoting if the random effects are assumed independent ; default is FALSE
<code>n.chains</code>	the number of parallel chains for the model; default is 1.
<code>n.iter</code>	integer specifying the total number of iterations; default is 10000
<code>n.burnin</code>	integer specifying how many of <code>n.iter</code> to discard as burn-in ; default is 5000
<code>n.thin</code>	integer specifying the thinning of the chains; default is 1
<code>n.adapt</code>	integer specifying the number of iterations to use for adaptation; default is 5000
<code>precision</code>	variance by default for vague prior distribution
<code>C</code>	value used in the zero trick; default is 1000.
<code>save_jagsUI</code>	If TRUE (by default), the output of <code>jagsUI</code> package is returned by the function
<code>save_va</code>	If TRUE (is FALSE by default), the draws of auxiliary variable <code>W</code> is returned by the function
<code>parallel</code>	see <code>jagsUI::jags()</code> function

Value

A `Bqrjm` object is a list with the following elements:

`mean` list of posterior mean for each parameter
`median` list of posterior median for each parameter
`modes` list of posterior mode for each parameter

StErr list of standard error for each parameter
 StDev list of standard deviation for each parameter
 ICs list of the credibility interval at 0.95 for each parameters excepted for covariance parameters in covariance matrix of random effects. Otherwise, use save_jagsUI=TRUE to have the associated quantiles.
 data data included in argument
 sims.list list of the MCMC chains of the parameters and random effects
 control list of arguments giving details about the estimation
 random_effect list for each quantile including both posterior mean and posterior standard deviation of subject-specific random effects
 out_jagsUI only if save_jagsUI=TRUE in argument: list including posterior mean, median, quantiles (2.5%, 25%, 50%, 75%, 97.5%), standart deviation for each parameter and each random effect. Moreover, this list also returns the MCMC draws, the Gelman and Rubin diagnostics (see output of jagsUI objects)

Author(s)

Antoine Barbieri

References

Ming Yang, Sheng Luo, and Stacia DeSantis (2019). *Bayesian quantile regression joint models: Inference and dynamic predictions*. *Statistical Methods in Medical Research*, 28(8):2524-2537. doi: 10.1177/0962280218784757.

Examples

```
#---- load data
data(dataLong)

#---- Fit quantile regression joint model for the first quartile
qrjm_75 <- qrjm(formFixed = y ~ visit,
               formRandom = ~ visit,
               formGroup = ~ ID,
               formSurv = Surv(time, event) ~ X1 + X2,
               survMod = "weibull",
               param = "value",
               timeVar= "visit",
               data = dataLong,
               tau = 0.75)

#---- Visualize the trace for beta parameters
jagsUI::traceplot(qrjm_75$out_jagsUI, parameters = "beta")

#---- Get the estimated coefficients: posterior means
qrjm_75$mean

#---- Summary of output
```

```
summary(qrjm_75)
```

wave

Data of wave

Description

Data including environmental measurements around Bordeaux from CANDHIS database and data from InfoClimat website. CANDHIS national in situ wave measurement database. The measurements were carried out within the framework of a collaboration between the Grand Port Maritime de Nantes St-Nazaire, the École Centrale de Nantes and CEREMA. In addition, data from the InfoClimat site over the same period are used.

Usage

```
data(wave)
```

Format

A data.frame with 453 observations (rows) and 25 variables with explicit names. The first variables are:

date date of measure

temperature temperature

pression pressure

humidite_relative Humidity

point2rose Dew point temperature

visibilite_horiz visibility

vent_cite_moy wind speed average

vent_vit_rafale maximum of wind speed

precipitation_cum cumulative rainfall by day

References

InfoClimat (https://www.infoclimat.fr/climatologie/stations_principales.php?)

Examples

```
data(wave)
```

Index

dataLong, [2](#)
deviance, [2](#)

lqm, [4](#)
lqmm, [6](#)

qrjm, [8](#)

wave, [11](#)