Title

bayes: qreg — Bayesian quantile regression+

+This command is part of StataNow.

Description

`bayes: qreg` fits a Bayesian quantile regression to a continuous outcome; see [BAYES] **bayes** and [R] **qreg** for details.

Quick start

Bayesian median regression of `y` on `x1` and `x2`, using default normal priors for regression coefficients

```
bayes: qreg y x1 x2
```

Same as above, and fix the scale σ equal to 1

```
bayes, sigma(1): qreg y x1 x2
```

Use a standard deviation of 10 instead of 100 for the default normal priors

```
bayes, normalprior(10): qreg y x1 x2
```

Use uniform priors for the slopes and a normal prior for the intercept

```
bayes, prior({y_q50: x1 x2}, uniform(-10,10)) ///
prior({y_q50: _cons}, normal(0,10)): qreg y x1 x2
```

Bayesian quantile regression of the 75th percentile of `y` conditional on `x1` and `x2`

```
bayes: qreg y x1 x2, quantile(0.75)
```

Same as above, but use uniform priors for the slopes and a normal prior for the intercept

```
bayes, prior({y_q75: x1 x2}, uniform(-10,10)) ///
prior({y_q75: _cons}, normal(0,10)): qreg y x1 x2, quantile(0.75)
```

Save simulation results to `simdata.dta`, and use a random-number seed for reproducibility

```
bayes, saving(simdata) rseed(123): qreg y x1 x2
```

Specify 20,000 Markov chain Monte Carlo (MCMC) samples, set length of the burn-in period to 5,000, and request that a dot be displayed every 500 simulations

```
bayes, mcmcsize(20000) burnin(5000) dots(500): qreg y x1 x2
```

In the above, request that the 90% highest posterior density (HPD) credible interval be displayed instead of the default 95% equal-tailed credible interval

```
bayes, clevel(90) hpd
```

Also see *Quick start* in [BAYES] **bayes** and *Quick start* in [R] **qreg**.

Menu

Statistics > Linear models and related > Bayesian regression > Quantile regression

Syntax

`bayes` [, *bayesopts*] : `qreg` *depvar* [*indepvars*] [*if*] [*in*] [*weight*] [, *options*]

options	Description
Model	
`quantile(#)`	estimate # quantile; default is `quantile(.5)`
`noconstant`	suppress constant term
Reporting	
display_options	control spacing, line width, and base and empty cells
`level(#)`	set credible level; default is `level(95)`

indepvars may contain factor variables; see [U] **11.4.3 Factor variables**.

`fweight`s are allowed; see [U] **11.1.6 weight**.

`bayes: qreg, level()` is equivalent to `bayes, clevel(): qreg`.

For a detailed description of options, see *Options* in [R] **qreg**.

bayesopts	Description
Priors	
* `sigma(#)`	specify a fixed scale σ; default is random σ parameter with inverse-gamma prior
* `normalprior(#)`	specify standard deviation of default normal priors for regression coefficients; default is `normalprior(100)`
* `igammaprior(# #)`	specify shape and scale of default inverse-gamma prior for scaling factor σ; default is `igammaprior(0.01 0.01)`
`prior(`*priorspec*`)`	prior for model parameters; this option may be repeated
`dryrun`	show model summary without estimation
Simulation	
`nchains(#)`	number of chains; default is to simulate one chain
`mcmcsize(#)`	MCMC sample size; default is `mcmcsize(10000)`
`burnin(#)`	burn-in period; default is `burnin(2500)`
`thinning(#)`	thinning interval; default is `thinning(1)`
`rseed(#)`	random-number seed
`exclude(`*paramref*`)`	specify model parameters to be excluded from the simulation results
Blocking	
* `blocksize(#)`	maximum block size; default is `blocksize(50)`
`block(`*paramref* [, *blockopts*]`)`	specify a block of model parameters; this option may be repeated
`blocksummary`	display block summary
* `noblocking`	do not block parameters by default

Initialization	
initial(*initspec*)	specify initial values for model parameters with a single chain
init#(*initspec*)	specify initial values for #th chain; requires nchains()
initall(*initspec*)	specify initial values for all chains; requires nchains()
nomleinitial	suppress the use of linear programming estimates as starting values
initrandom	specify random initial values
initsummary	display initial values used for simulation
*noisily	display output from the estimation command during initialization
Adaptation	
adaptation(*adaptopts*)	control the adaptive MCMC procedure
scale(#)	initial multiplier for scale factor; default is scale(2.38)
covariance(*cov*)	initial proposal covariance; default is the identity matrix
Reporting	
clevel(#)	set credible interval level; default is clevel(95)
hpd	display HPD credible intervals instead of the default equal-tailed credible intervals
batch(#)	specify length of block for batch-means calculations; default is batch(0)
saving(*filename*[, replace])	save simulation results to *filename*.dta
nomodelsummary	suppress model summary
chainsdetail	display detailed simulation summary for each chain
[no]dots	suppress dots or display dots every 100 iterations and iteration numbers every 1,000 iterations; default is nodots
dots(#[, every(#)])	display dots as simulation is performed
[no]show(*paramref*)	specify model parameters to be excluded from or included in the output
notable	suppress estimation table
noheader	suppress output header
title(*string*)	display *string* as title above the table of parameter estimates
display_options	control spacing, line width, and base and empty cells
Advanced	
search(*search_options*)	control the search for feasible initial values
corrlag(#)	specify maximum autocorrelation lag; default varies
corrtol(#)	specify autocorrelation tolerance; default is corrtol(0.01)

*Starred options are specific to the bayes prefix; other options are common between bayes and bayesmh.

priorspec and *paramref* are defined in [BAYES] **bayesmh**.

paramref may contain factor variables; see [U] **11.4.3 Factor variables**.

collect is allowed; see [U] **11.1.10 Prefix commands**.

See [U] **20 Estimation and postestimation commands** for more capabilities of estimation commands.

Model parameters are regression coefficients {*depvar*_q#:*indepvars*} and scaling factor {sigma}. Use the dryrun option to see the definitions of model parameters prior to estimation.

For a detailed description of *bayesopts*, see *Options* in [BAYES] **bayes**.

Remarks and examples

stata.com

For a general introduction to Bayesian analysis, see [BAYES] **Intro**. For a general introduction to Bayesian estimation using an adaptive Metropolis–Hastings algorithm, see [BAYES] **bayesmh**. For remarks and examples specific to the `bayes` prefix, see [BAYES] **bayes**. For details about the estimation command, see [R] **qreg**.

For a simple example of the `bayes` prefix, see *Introductory example* in [BAYES] **bayes**.

▷ Example 1: Median regression

Consider the following dataset from budget surveys administered to European households in the 19th century, described in Koenker and Bassett (1982). The data are originally from Engel (1857), who argued that as household income increases, food expenditure takes up a smaller share. We have the households' annual income, `income`, and annual food expenditure, `foodexp`.

```
. use https://www.stata-press.com/data/r18/engel1857
(European household budget survey)

. describe

Contains data from https://www.stata-press.com/data/r18/engel1857.dta
 Observations:           235                  European household budget survey
    Variables:             2                  7 Dec 2023 11:11
                                              (_dta has notes)
-------------------------------------------------------------------------------
Variable      Storage   Display    Value
    name         type    format    label      Variable label
-------------------------------------------------------------------------------
income          float   %9.0g                 Annual household income (1,000s
                                                Belgian francs)
foodexp         float   %9.0g                 Annual household food expenditure
                                                (1,000s Belgian francs)
-------------------------------------------------------------------------------
Sorted by:
```

Below, we fit a Bayesian quantile regression model with outcome variable `foodexp` and predictor variable `income`. By default, `bayes:qreg` fits a median regression model; in other words, we model the 50th percentile of `foodexp`.

```
. bayes, rseed(19): qreg foodexp income
Burn-in ...
Simulation ...

Model summary
------------------------------------------------------------------------------
Likelihood:
  foodexp ~ asymlaplaceq(xb_foodexp_q50,{sigma},.5)

Priors:
  {foodexp_q50:income _cons} ~ normal(0,10000)                             (1)
                     {sigma} ~ igamma(0.01,0.01)
------------------------------------------------------------------------------
(1) Parameters are elements of the linear form xb_foodexp_q50.

Bayesian quantile regression                     MCMC iterations  =     12,500
Random-walk Metropolis-Hastings sampling         Burn-in          =      2,500
                                                 MCMC sample size =     10,000
Quantile = .5                                    Number of obs    =        235
                                                 Acceptance rate  =      .3603
                                                 Efficiency:  min =     .09896
                                                              avg =       .151
Log marginal-likelihood =  186.43947                          max =      .2268

------------------------------------------------------------------------------
             |                                                Equal-tailed
             |      Mean   Std. dev.     MCSE     Median  [95% cred. interval]
-------------+----------------------------------------------------------------
foodexp_q50  |
      income |  .5567276   .0159401   .000507   .5562547   .5248025    .587735
       _cons |   .084986   .0143782   .000403   .0851108   .0575581   .1134264
-------------+----------------------------------------------------------------
       sigma |  .0377533   .0024907   .000052   .0376511   .0331066   .0430957
------------------------------------------------------------------------------
```

Using the mean posterior estimates for coefficients, we can express the relationship between the households' annual income and the annual food expenditure can be expressed as

$$\texttt{foodexp}_{\text{median}} = 0.56 \times \texttt{income} + 0.08$$

The median food expenditure is 640 Belgian francs for a household with an income of 1,000 Belgian francs ($0.56 + 0.08 = 0.64$); note that both `income` and `foodexp` are measured in 1,000s of Belgian francs. For this household, food expenditure comprises 64% of income ($640/1000 = 0.64$). However, the median food expenditure is 2,320 for a household with an income of 4,000 Belgian francs ($0.56 \times 4 + 0.08 = 2.32$); the median food expenditure comprises 58% of household income, as opposed to 64% for a household making 1,000 annually.

◁

▷ Example 2: Estimating other quantiles

We can check whether the effect of `income` varies across different quantiles of `foodexp` by comparing the median regression model from our last example with models for the 25th and 75th percentiles; we will use the `quantile()` option to specify the quantile levels of the outcome.

We use the `collect` prefix to collect results from each model, to be displayed in a table, and we store regression coefficients as scalars for later use.

```
. collect get: bayes, rseed(19): qreg foodexp income, quantile(0.25)
Burn-in ...
Simulation ...

Model summary
------------------------------------------------------------------------------
Likelihood:
  foodexp ~ asymlaplaceq(xb_foodexp_q25,{sigma},.25)

Priors:
  {foodexp_q25:income _cons} ~ normal(0,10000)                             (1)
                     {sigma} ~ igamma(0.01,0.01)
------------------------------------------------------------------------------
(1) Parameters are elements of the linear form xb_foodexp_q25.

Bayesian quantile regression                     MCMC iterations  =     12,500
Random-walk Metropolis-Hastings sampling         Burn-in          =      2,500
                                                 MCMC sample size =     10,000
Quantile = .25                                   Number of obs    =        235
                                                 Acceptance rate  =      .3423
                                                 Efficiency:  min =      .1436
                                                              avg =      .1765
Log marginal-likelihood =  169.18624                          max =      .2421
------------------------------------------------------------------------------
             |                                                Equal-tailed
             |      Mean   Std. dev.     MCSE     Median  [95% cred. interval]
-------------+----------------------------------------------------------------
foodexp_q25  |
      income |  .4718604   .0140225    .00037   .4735463   .4414884   .4948657
       _cons |  .0962851   .0116976   .000308   .0957929   .0742573   .1196877
-------------+----------------------------------------------------------------
       sigma |  .0304463   .0020364   .000041   .0303373   .0266857   .0347907
------------------------------------------------------------------------------

. scalar bqr1_b1 = e(mean)[1,1]

. scalar bqr1_b0 = e(mean)[1,2]
```

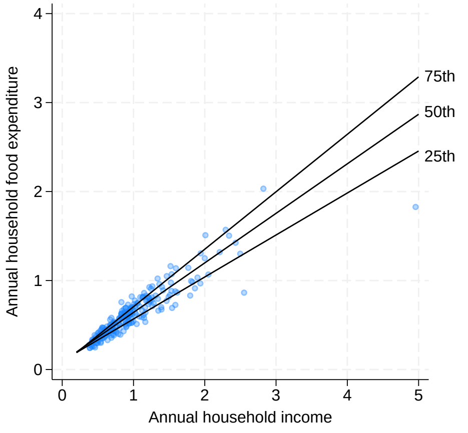
```
. collect get: bayes, rseed(19): qreg foodexp income, quantile(0.5)
Burn-in ...
Simulation ...

Model summary
------------------------------------------------------------------------------
Likelihood:
  foodexp ~ asymlaplaceq(xb_foodexp_q50,{sigma},.5)

Priors:
  {foodexp_q50:income _cons} ~ normal(0,10000)                             (1)
                     {sigma} ~ igamma(0.01,0.01)
------------------------------------------------------------------------------
(1) Parameters are elements of the linear form xb_foodexp_q50.

Bayesian quantile regression                     MCMC iterations  =     12,500
Random-walk Metropolis-Hastings sampling         Burn-in          =      2,500
                                                 MCMC sample size =     10,000
Quantile = .5                                    Number of obs    =        235
                                                 Acceptance rate  =      .3603
                                                 Efficiency:  min =     .09896
                                                              avg =       .151
Log marginal-likelihood =  186.43947                          max =      .2268

------------------------------------------------------------------------------
             |                                                Equal-tailed
             |      Mean   Std. dev.     MCSE     Median  [95% cred. interval]
-------------+----------------------------------------------------------------
foodexp_q50  |
      income |  .5567276   .0159401   .000507   .5562547   .5248025    .587735
       _cons |   .084986   .0143782   .000403   .0851108   .0575581   .1134264
-------------+----------------------------------------------------------------
       sigma |  .0377533   .0024907   .000052   .0376511   .0331066   .0430957
------------------------------------------------------------------------------

. scalar bqr2_b1 = e(mean)[1,1]

. scalar bqr2_b0 = e(mean)[1,2]
```

```
. collect get: bayes, rseed(19): qreg foodexp income, quantile(0.75)
Burn-in ...
Simulation ...

Model summary
------------------------------------------------------------------------------
Likelihood:
  foodexp ~ asymlaplaceq(xb_foodexp_q75,{sigma},.75)

Priors:
  {foodexp_q75:income _cons} ~ normal(0,10000)                             (1)
                     {sigma} ~ igamma(0.01,0.01)
------------------------------------------------------------------------------
(1) Parameters are elements of the linear form xb_foodexp_q75.

Bayesian quantile regression                      MCMC iterations  =     12,500
Random-walk Metropolis-Hastings sampling          Burn-in          =      2,500
                                                  MCMC sample size =     10,000
Quantile = .75                                    Number of obs    =        235
                                                  Acceptance rate  =      .3103
                                                  Efficiency:  min =      .1421
                                                               avg =      .1704
Log marginal-likelihood =  188.25668                           max =      .2262

------------------------------------------------------------------------------
             |                                                Equal-tailed
             |      Mean   Std. dev.     MCSE     Median  [95% cred. interval]
-------------+----------------------------------------------------------------
foodexp_q75  |
      income |  .6456717   .0170002   .000451   .6461026   .6089782   .6757706
       _cons |  .0606789    .014418   .000381    .060412    .034519   .0924086
-------------+----------------------------------------------------------------
       sigma |  .0280768   .0018942    .00004   .0279643   .0245888   .0321131
------------------------------------------------------------------------------

. scalar bqr3_b1 = e(mean)[1,1]

. scalar bqr3_b0 = e(mean)[1,2]

. collect label levels colname income "Annual household income", modify

. collect label levels cmdset 1 "25th" 2 "50th" 3 "75th"

. collect layout (colname[income]#result[mean sd]) (cmdset)

Collection: default
      Rows: colname[income]#result[mean sd]
   Columns: cmdset
   Table 1: 3 x 3

-----------------------------------------------------
                        |     25th     50th     75th
------------------------+----------------------------
Annual household income |
  Posterior means       | .4718604 .5567276 .6456717
  Std. dev.             | .0140225 .0159401 .0170002
-----------------------------------------------------
```

Before we lay out our table, we shorten the label for `income`, and we label the results with the quantile being estimated. To learn more about modifying labels in a collection and laying out a table, see [TABLES] **collect label** and [TABLES] **collect layout**. The table shows that the coefficient of `income` increases across the quantiles, from 0.472 for the 25th quantile to 0.646 for the 75th quantile.

Below, we plot the posterior mean quantile lines corresponding to the three models.

```
. twoway (scatter foodexp income, mcolor(%30)) ||
> (function y = bqr3_b1 * x + bqr3_b0,  range(0.2 5) lcolor(black)) ||
> (function y = bqr2_b1 * x + bqr2_b0,  range(0.2 5) lcolor(black)) ||
> (function y = bqr1_b1 * x + bqr1_b0,  range(0.2 5) lcolor(black)),
> legend(off) xtitle("Annual household income")
> ytitle("Annual household food expenditure") aspect(1)
> text(3.3 5.3 "75th" 2.9 5.3  "50th" 2.4 5.3  "25th")
```



The above plot of `foodexp` versus `income` (and the fitted quantile lines) indicates the potential presence of heteroskedasticity, although this inference may require further verification.

In contrast to quantile regression, the linear regression model assumes homoskedasticity of the outcome with respect to each predictor variable, meaning that the residual variance is uniform throughout the range of predicted values. A formal comparison between quantile and linear regression models will show which one provides a better fit for the data.

We first run the linear and the median regression models and store the estimation results in memory with `estimates store`. Then, we use the `bayestest model` command to compute and compare the posterior model probabilities.

```
. quietly bayes, rseed(19) saving(meanreg_sim, replace): regress foodexp income
. estimates store meanreg
. quietly bayes, rseed(19) saving(medianreg_sim, replace): qreg foodexp income
. estimates store medianreg
. bayestest model meanreg medianreg
Bayesian model tests
```

	log(ML)	P(M)	P(M|y)
meanreg	152.5311	0.5000	0.0000
medianreg	186.4395	0.5000	1.0000

```
Note: Marginal likelihood (ML) is computed using
      Laplace-Metropolis approximation.
```

The median regression model, with an estimated posterior model probability of 1, provides an overwhelmingly better fit than the simple linear regression, which is consistent with the noted heteroskedasticity of the outcome `foodexp`.

◁

Stored results

See *Stored results* in [BAYES] **bayes**. In addition, `bayes: qreg` stores the following in `e()`:

Scalars

`e(q)`	quantile requested
`e(q_v)`	value of the quantile

Methods and formulas

In the context of quantile regression, it is instructive to consider the optimization process as outlined in *Methods and formulas* of [R] **qreg**.

Let τ be the target estimation quantile of the outcome. For the ith observation, let $\mathbf{x}_i$ be the vector of independent variables and y_i be the outcome value. The ith residual is $\varepsilon_i = y_i - \mathbf{x}_i'\boldsymbol{\beta}_\tau$, where $\boldsymbol{\beta}_\tau$ is a quantile-specific vector of coefficients that is subject to estimation.

The objective function under consideration seeks to minimize a specific criterion:

$$\min_{\boldsymbol{\beta}_\tau} \sum_i c_\tau(\varepsilon_i) \tag{1}$$

Here $c_\tau(\varepsilon_i)$ is defined as $c_\tau(\varepsilon_i) = \{\tau - \mathbf{1}(\varepsilon_i < 0)\}\varepsilon_i$, where $\mathbf{1}(\cdot)$ is an indicator function.

Yu and Moyeed (2001) proposed an alternative representation of (1), wherein the optimization problem was reformulated as the maximization of a likelihood function employing the asymmetric Laplace distribution (ALD).

The probability density function of ALD can be defined as

$$f_\tau(x;\mu,\sigma) = \frac{\tau(1-\tau)}{\sigma}\exp\left\{-c_\tau\left(\frac{x-\mu}{\sigma}\right)\right\},\quad \sigma > 0$$

where μ is a location parameter and σ is a scale parameter.

The likelihood function of a quantile regression with outcome observations y_i and covariates $\mathbf{x}_i$, $i = 1,\ldots,n$, is a product of ALDs with location parameters $\mu_i = \mathbf{x}_i'\boldsymbol{\beta}_\tau$,

$$L(\mathbf{y}|\boldsymbol{\beta}_\tau,\sigma) = \prod_{i=1}^n f_\tau(y_i;\mathbf{x}_i'\boldsymbol{\beta}_\tau,\sigma) = \frac{\tau^n(1-\tau)^n}{\sigma^n}\exp\left\{-\sum_i c_\tau\left(\frac{y_i-\mathbf{x}_i'\boldsymbol{\beta}_\tau}{\sigma}\right)\right\}$$

Bayesian quantile regression considers a posterior distribution of $\boldsymbol{\beta}_\tau$ and σ, denoted as $p(\boldsymbol{\beta}_\tau,\sigma|\mathbf{y})$, which is proportional to the product of the likelihood function and a prior distribution for $\boldsymbol{\beta}_\tau$ and σ, $\pi(\boldsymbol{\beta}_\tau,\sigma)$,

$$p(\boldsymbol{\beta}_\tau,\sigma|\mathbf{y}) \propto L(\mathbf{y}|\boldsymbol{\beta}_\tau,\sigma)\pi(\boldsymbol{\beta}_\tau,\sigma)$$

The default prior distribution choices are independent normal with mean 0 and variance of 10,000 for $\boldsymbol{\beta}_\tau$ and inverse-gamma with shape 0.01 and scale of 0.01 for σ. The `bayes: qreg` command performs estimation using adaptive Metropolis–Hastings sampling.

See *Methods and formulas* in [BAYES] **bayesmh**.

References

Engel, E. 1857. Die productions-und consumtionsverhältnisse des königreichs sachsen. *Zeitschrift des Statistischen Bureaus des Königlich Sächsischen Ministeriums des Innern* 8: 1–54.

Koenker, R., and G. Bassett, Jr. 1982. Robust tests for heteroscedasticity based on regression quantiles. *Econometrica* 50: 43–61. https://doi.org/10.2307/1912528.

Yu, K., and R. A. Moyeed. 2001. Bayesian quantile regression. *Statistics and Probability Letters* 54: 437–447. https://doi.org/10.1016/S0167-7152(01)00124-9.

Also see

[BAYES] **bayes** — Bayesian regression models using the bayes prefix+

[R] **qreg** — Quantile regression

[BAYES] **Bayesian postestimation** — Postestimation tools for bayesmh and the bayes prefix

[BAYES] **Bayesian estimation** — Bayesian estimation commands

[BAYES] **Bayesian commands** — Introduction to commands for Bayesian analysis

[BAYES] **Intro** — Introduction to Bayesian analysis

[BAYES] **Glossary**

For suggested citations, see the FAQ on citing Stata documentation.