

[Description](#)[Remarks and examples](#)[Also see](#)[Quick start](#)[Stored results](#)[Menu](#)[Methods and formulas](#)[Syntax](#)[References](#)

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 x_1 and x_2 , 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 x_1 and x_2

```
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, mcmcs(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 ]
```

<i>options</i>	Description
Model	
<u>quantile</u> (#)	estimate # quantile; default is <code>quantile(.5)</code>
<u>noconstant</u>	suppress constant term
Reporting	
<u>display_options</u>	control spacing, line width, and base and empty cells
<u>level</u> (#)	set credible level; default is <code>level(95)</code>
<i>indepvars</i> may contain factor variables; see [U] 11.4.3 Factor variables.	
<i>fwweights</i> are allowed; see [U] 11.1.6 weight.	
<code>bayes: qreg, level()</code> is equivalent to <code>bayes, clevel(): qreg</code> .	
For a detailed description of options, see <i>Options</i> in [R] <code>qreg</code> .	
<i>bayesopts</i>	Description
Priors	
* <u>sigma</u> (#)	specify a fixed scale σ ; default is random σ parameter with inverse-gamma prior
* <u>normalprior</u> (#)	specify standard deviation of default normal priors for regression coefficients; default is <code>normalprior(100)</code>
* <u>igammaprior</u> (# #)	specify shape and scale of default inverse-gamma prior for scaling factor σ ; default is <code>igammaprior(0.01 0.01)</code>
<u>prior</u> (<i>priorspec</i>)	prior for model parameters; this option may be repeated
<u>dryrun</u>	show model summary without estimation
Simulation	
<u>nchains</u> (#)	number of chains; default is to simulate one chain
<u>mcmcsize</u> (#)	MCMC sample size; default is <code>mcmcsize(10000)</code>
<u>burnin</u> (#)	burn-in period; default is <code>burnin(2500)</code>
<u>thinning</u> (#)	thinning interval; default is <code>thinning(1)</code>
<u>rseed</u> (#)	random-number seed
<u>exclude</u> (<i>paramref</i>)	specify model parameters to be excluded from the simulation results
Blocking	
* <u>blocksize</u> (#)	maximum block size; default is <code>blocksize(50)</code>
<u>block</u> (<i>paramref</i> [, <i>blockopts</i>])	specify a block of model parameters; this option may be repeated
<u>blocksummary</u>	display block summary
* <u>noblocking</u>	do not block parameters by default

Initialization

<code>initial(<i>initspec</i>)</code>	specify initial values for model parameters with a single chain
<code>init#(<i>initspec</i>)</code>	specify initial values for #th chain; requires <code>nchains()</code>
<code>initall(<i>initspec</i>)</code>	specify initial values for all chains; requires <code>nchains()</code>
<code>nomleinitial</code>	suppress the use of linear programming estimates as starting values
<code>initrandom</code>	specify random initial values
<code>initsummary</code>	display initial values used for simulation
* <code>noisily</code>	display output from the estimation command during initialization

Reporting

<code>clevel(#)</code>	set credible interval level; default is <code>clevel(95)</code>
<code>hpd</code>	display HPD credible intervals instead of the default equal-tailed credible intervals
<code>batch(#)</code>	specify length of block for batch-means calculations; default is <code>batch(0)</code>
<code>saving(filename[, replace])</code>	save simulation results to <i>filename.dta</i>
<code>nomodelsummary</code>	suppress model summary
<code>chainsdetail</code>	display detailed simulation summary for each chain
<code>[no]dots</code>	suppress dots or display dots every 100 iterations and iteration numbers every 1,000 iterations; default is <code>nodots</code>
<code>dots(#[, every(#)])</code>	display dots as simulation is performed
<code>[no]show(paramref)</code>	specify model parameters to be excluded from or included in the output
<code>notable</code>	suppress estimation table
<code>noheader</code>	suppress output header
<code>title(string)</code>	display <i>string</i> as title above the table of parameter estimates
<code>display_options</code>	control spacing, line width, and base and empty cells

Advanced

<code>search(search_options)</code>	control the search for feasible initial values
<code>corrlag(#)</code>	specify maximum autocorrelation lag; default varies
<code>corrtol(#)</code>	specify autocorrelation tolerance; default is <code>corrtol(0.01)</code>

* Starred options are specific to the `bayer` prefix; other options are common between `bayer` and `bayermh`.

priorspec and *paramref* are defined in [BAYES] `bayermh`.

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 *bayeropts*, see *Options* in [BAYES] `bayer`.

Remarks and examples

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] `bayermh`. For remarks and examples specific to the `bayer` prefix, see [BAYES] `bayer`. For details about the estimation command, see [R] `qreg`.

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

➤ 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/r19/engel1857
(European household budget survey)

. describe
Contains data from https://www.stata-press.com/data/r19/engel1857.dta
Observations:      235      European household budget survey
Variables:         2       7 Dec 2024 11:11
                        (_dta has notes)
```

Variable name	Storage type	Display format	Value 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)
                                {sigma} ~ igamma(0.01,0.01)                                     (1)
```

```
(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
                                           max              =     .2268

Log marginal-likelihood = 186.43947
```

	Mean	Std. dev.	MCSE	Median	Equal-tailed [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 as

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

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)
                                {sigma} ~ igamma(0.01,0.01)
(1)
```

```
(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
                                           max            =    .2421

Log marginal-likelihood =  169.18624
```

	Mean	Std. dev.	MCSE	Median	Equal-tailed [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)
                                {sigma} ~ igamma(0.01,0.01) (1)
```

(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
```

	Mean	Std. dev.	MCSE	Median	Equal-tailed [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

	Mean	Std. dev.	MCSE	Median	Equal-tailed [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

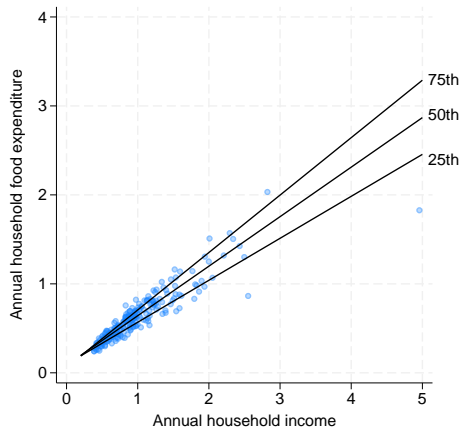
Table 1: 3 x 3

	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

<code>e(q)</code>	quantile requested
<code>e(q_v)</code>	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 i th observation, let \mathbf{x}_i be the vector of independent variables and y_i be the outcome value. The i th 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:

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

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 , $i = 1, \dots, n$, 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](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 after Bayesian estimation
- [BAYES] **Bayesian estimation** — Bayesian estimation commands
- [BAYES] **Bayesian commands** — Introduction to commands for Bayesian analysis
- [BAYES] **Intro** — Introduction to Bayesian analysis
- [BAYES] **Glossary**

Stata, Stata Press, Mata, NetCourse, and NetCourseNow are registered trademarks of StataCorp LLC. Stata and Stata Press are registered trademarks with the World Intellectual Property Organization of the United Nations. StataNow is a trademark of StataCorp LLC. Other brand and product names are registered trademarks or trademarks of their respective companies. Copyright © 1985–2025 StataCorp LLC, College Station, TX, USA. All rights reserved.



For suggested citations, see the FAQ on [citing Stata documentation](#).