Package ‘bayesQRsurvey’

May 7, 2026

Title Bayesian Quantile Regression Models for Complex Survey Data Analysis

Version 0.2.2

Description Provides Bayesian quantile regression models for complex survey data under informative sampling using survey-weighted estimators. Both single- and multiple-output models are supported. To accelerate computation, all algorithms are implemented in 'C++' using 'Rcpp', 'RcppArmadillo', and 'RcppEigen', and are called from 'R'. See Nascimento and Gonçalves (2024) <doi:10.1093/jssam/smae015> and Nascimento and Gonçalves (2025, in press) <`https://academic.oup.com/jssam`>.

License MIT + file LICENSE

LazyData true

LazyDataCompression gzip

Encoding UTF-8

RoxygenNote 7.3.3

LinkingTo Rcpp, RcppArmadillo, RcppEigen

Imports Rcpp, stats, graphics, methods, pracma, ggplot2, rlang, posterior

Suggests MASS, knitr, rmarkdown, grDevices, testthat (>= 3.0.0)

Config/testthat/edition 3

SystemRequirements BLAS, LAPACK

URL `https://github.com/torodriguezt/bayesQRsurvey`

BugReports `https://github.com/torodriguezt/bayesQRsurvey/issues`

NeedsCompilation yes

Author Tomás Rodríguez Taborda [aut, cre],
Johnatan Cardona Jiménez [aut],
Marcus L. Nascimento [aut],
Kelly Cristina Mota Gonçalves [aut]

Maintainer Tomás Rodríguez Taborda <`torodriguezt@unal.edu.co`>

Depends R (>= 3.5.0)

Repository CRAN

Date/Publication 2026-04-06 19:00:15 UTC

Contents

Anthro 2
bqr.svy 3
mo.bqr.svy 5
plot.bqr.svy 8
plotQuantileRegion 10
print.bayesQRsurvey 12
prior 13
summary.bayesQRsurvey 15

Index **17**

`Anthro` *Children anthropometric data*

Description

The dataset comprises 985 observations and 8 variables on children aged 0-60 months from the Central-West region of Brazil. The data are a cleaned subset of the 2006 Brazilian National Demographic Health Survey of Women and Children (NDHS), a complex survey in which sampling units were selected in two stages: primary sampling units (PSUs) were census tracts, and secondary sampling units (SSUs) were households. Tract selection within each stratum was designed to ensure a minimum number of blood samples, given the prevalence of vitamin A deficiency in children. Outlier weights (> 30 kg) and rows with missing values have been removed.

Usage

```
data("Anthro")
```

Format

The data frame has the following components:

- **wgt** : weight
- **hgt** : height
- **wgt_ind** : weight-for-age index
- **hgt_ind** : height-for-age index
- **ruc** : rural-urban classification (urban = 1 and rural = 2)
- **sex** : factor with levels `0` (girl) and `1` (boy)
- **age** : age in months
- **dweight** : sampling weights

Source

The 2006 Brazilian National Demographic Health Survey of Women and Children (NDHS) was published by the Brazilian Institute of Geography and Statistics.

`bqr.svy` *Bayesian quantile regression for complex survey data*

Description

`bqr.svy` implements Bayesian methods for estimating quantile regression models for complex survey data analysis regarding single (univariate) outputs. To improve computational efficiency, the Markov Chain Monte Carlo (MCMC) algorithms are implemented in 'C++'.

Usage

```
bqr.svy(
  formula,
  weights = NULL,
  data = NULL,
  quantile = 0.5,
  method = c("ald", "score", "approximate"),
  prior = NULL,
  niter = 20000,
  burnin = 0,
  thin = 1,
  verbose = TRUE,
  estimate_sigma = FALSE
)
```

Arguments

`formula`	a symbolic description of the model to be fit.
`weights`	an optional numerical vector containing the survey weights. If `NULL`, equal weights are used.
`data`	an optional data frame containing the variables in the model.
`quantile`	numerical scalar or vector containing quantile(s) of interest (default=0.5).
`method`	one of `"ald"`, `"score"` and `"approximate"` (default=`"ald"`).
`prior`	a `bqr_prior` object of class "prior". If omitted, a vague prior is assumed (see `prior`).
`niter`	number of MCMC draws.
`burnin`	number of initial MCMC draws to be discarded.(default = 0)
`thin`	thinning parameter, i.e., keep every keepth draw (default=1).
`verbose`	logical flag indicating whether to print progress messages (default=TRUE).
`estimate_sigma`	logical flag indicating whether to estimate the scale parameter when method = "ald" (default=FALSE and σ^2 is set to 1)

Details

The bqr.svy function can estimate three types of models, where the quantile regression coefficients are defined at the super-population level, and their estimators are built upon the survey weights.

- `"ald"` – The asymmetric Laplace distribution as working likelihood.
- `"score"` – A score based likelihood function.
- `"approximate"` – A pseudolikelihood function based on a Gaussian approximation.

Value

An object of class `"bqr.svy"`, containing:

`beta`	Posterior mean estimates of regression coefficients.
`draws`	Posterior draws from the MCMC sampler.
`accept_rate`	Average acceptance rate (if available).
`warmup`, `thin`	MCMC control parameters used during sampling.
`quantile`	The quantile(s) fitted.
`prior`	Prior specification used.
`formula`, `terms`, `model`	Model specification details.
`runtime`	Elapsed runtime in seconds.
`method`	Estimation method
`estimate_sigma`	Logical flag indicating whether the scale parameter σ^2 was estimated (`TRUE`) or fixed at 1 (`FALSE`).

References

Nascimento, M. L. & Gonçalves, K. C. M. (2024). Bayesian Quantile Regression Models for Complex Survey Data Under Informative Sampling. *Journal of Survey Statistics and Methodology*, 12(4), 1105–1130. doi:10.1093/jssam/smae015

Examples

```
# Generate population data
set.seed(123)
N    <- 10000
x1_p <- runif(N, -1, 1)
x2_p <- runif(N, -1, 1)
y_p  <- 2 + 1.5 * x1_p - 0.8 * x2_p + rnorm(N)

# Generate sample data
n <- 500
z_aux <- rnorm(N, mean = 1 + y_p, sd = .5)
p_aux <- 1 / (1 + exp(2.5 - 0.5 * z_aux))
s_ind <- sample(1:N, n, replace = FALSE, prob = p_aux)
y_s   <- y_p[s_ind]
x1_s  <- x1_p[s_ind]
```

```
x2_s  <- x2_p[s_ind]
w     <- 1 / p_aux[s_ind]
data  <- data.frame(y = y_s, x1 = x1_s, x2 = x2_s, w = w)

# Basic usage with default method ('ald') and priors (vague)
fit1 <- bqr.svy(y ~ x1 + x2, weights = w, data = data)

# Specify informative priors
prior <- prior(
  beta_x_mean = c(2, 1.5, -0.8),
  beta_x_cov  = diag(c(0.25, 0.25, 0.25)),
  sigma_shape = 1,
  sigma_rate  = 1
)
fit2 <- bqr.svy(y ~ x1 + x2, weights = w, data = data, prior = prior)

# Specify different methods
fit_score  <- bqr.svy(y ~ x1 + x2, weights = w, data = data, method = "score")
fit_approx <- bqr.svy(y ~ x1 + x2, weights = w, data = data, method = "approximate")
```

`mo.bqr.svy` *Multiple-Output Bayesian quantile regression for complex survey data*

Description

mo.bqr.svy implements a Bayesian approach to multiple-output quantile regression for complex survey data analysis. The method builds a quantile region based on a directional approach. To improve computational efficiency, an Expectation-Maximization (EM) algorithm is implemented instead of the usual Markov Chain Monte Carlo (MCMC).

Usage

```
mo.bqr.svy(
  formula,
  weights = NULL,
  data = NULL,
  quantile = 0.5,
  prior = NULL,
  U = NULL,
  gamma_U = NULL,
  n_dir = NULL,
  epsilon = 1e-06,
  max_iter = 1000,
  verbose = FALSE,
  estimate_sigma = FALSE
)
```

Arguments

`formula`	a symbolic description of the model to be fit.
`weights`	an optional numerical vector containing the survey weights. If `NULL`, equal weights are used.
`data`	an optional data frame containing the variables in the model.
`quantile`	numerical scalar or vector containing quantile(s) of interest (default=0.5).
`prior`	a `bqr_prior` object of class "prior". If omitted, a vague prior is assumed (see `prior`).
`U`	an optional $d \times K$-matrix of directions, where d indicates the response variable dimension and K indicates indicates the number of directions.
`gamma_U`	an optional list with length equal to K for which each element corresponds to $d \times (d-1)$-matrix of ortoghonal basis for each row of `U`.
`n_dir`	numerical scalar corresponding to the number of directions (if `U` and `gamma_U` are not supplied).
`epsilon`	numerical scalar indicating the convergence tolerance for the EM algorithm (default = 1e-6).
`max_iter`	numerical scalar indicating maximum number of EM iterations (default = 1000).
`verbose`	logical flag indicating whether to print progress messages (default=FALSE).
`estimate_sigma`	logical flag indicating whether to estimate the scale parameter when method = "ald" (default=FALSE and σ^2 is set to 1)

Value

An object of class `"mo.bqr.svy"` containing:

`call`	The matched call
`formula`	The model formula
`terms`	The terms object
`quantile`	Vector of fitted quantiles
`prior`	List of priors used for each quantile
`fit`	List of fitted results for each quantile, each containing one sub-list per direction
`coefficients`	Coefficients organized by quantile
`sigma`	List of scale parameters by quantile and direction. If `estimate_sigma = FALSE`, all entries are fixed at 1. If `estimate_sigma = TRUE`, each entry contains the estimated value of σ (posterior mode from EM).
`n_dir`	Number of directions
`U`	Matrix of projection directions ($d \times K$)
`Gamma_list`	List of orthogonal complement bases, one per direction
`n_obs`	Number of observations
`n_vars`	Number of covariates
`response_dim`	Dimension of the response d
`estimate_sigma`	Logical flag indicating whether the scale parameter σ^2 was estimated (`TRUE`) or fixed at 1 (`FALSE`).

References

Nascimento, M. L. & Gonçalves, K. C. M. (2024). Bayesian Quantile Regression Models for Complex Survey Data Under Informative Sampling. *Journal of Survey Statistics and Methodology*, 12(4), 1105–1130. doi:10.1093/jssam/smae015

Examples

```
library(MASS)

# Generate population data
set.seed(123)
N    <- 10000
data <- mvrnorm(N, rep(0, 3),
                matrix(c(4, 0, 2,
                         0, 1, 1.5,
                         2, 1.5, 9), 3, 3))
x_p  <- as.matrix(data[, 1])
y_p  <- data[, 2:3] + cbind(rep(0, N), x_p)

# Generate sample data
n <- 500
z_aux <- rnorm(N, mean = 1 + y_p, sd = 0.5)
p_aux <- 1 / (1 + exp(2.5 - 0.5 * z_aux))
s_ind <- sample(1:N, n, replace = FALSE, prob = p_aux)
y_s   <- y_p[s_ind, ]
x_s   <- x_p[s_ind, ]
w     <- 1 / p_aux[s_ind]
data_s <- data.frame(y1 = y_s[, 1],
                     y2 = y_s[, 2],
                     x1 = x_s,
                     w  = w)

# Basic usage with default priors when U and gamma_U are given
fit1 <- mo.bqr.svy(
  cbind(y1, y2) ~ x1,
  weights = w,
  data = data_s,
  quantile = c(0.1, 0.2),
  U = matrix(c(0, 1, 1/sqrt(2), 1/sqrt(2)), 2),
  gamma_U = list(c(1, 0), c(1/sqrt(2), -1/sqrt(2)))
)

# Basic usage with default priors when n_dir is given
fit2 <- mo.bqr.svy(
  cbind(y1, y2) ~ x1,
  weights = w,
  data = data_s,
  quantile = c(0.1, 0.2),
  n_dir = 2
)
```

`plot.bqr.svy` *Plot Method for Bayesian Weighted Quantile Regression*

Description

Plot method for objects of class `bqr.svy` produced by `bqr.svy()`. It can display fitted quantile curves, coefficient–quantile profiles, MCMC trace plots, and posterior densities.

Usage

```
## S3 method for class 'bqr.svy'
plot(
  x,
  y = NULL,
  type = c("fit", "quantile", "trace", "density"),
  tau = NULL,
  which = NULL,
  add_points = TRUE,
  combine = TRUE,
  show_ci = FALSE,
  ci_probs = c(0.1, 0.9),
  at = NULL,
  grid_length = 200,
  points_alpha = 0.4,
  point_size = 2.5,
  line_size = 1.2,
  main = NULL,
  use_ggplot = TRUE,
  theme_style = c("minimal", "classic", "bw", "light"),
  color_palette = c("viridis", "plasma", "set2", "dark2"),
  add_h0 = FALSE,
  add_ols = FALSE,
  ols_fit = NULL,
  ols_weights = NULL,
  ...
)

## S3 method for class 'bwqr_fit'
plot(x, ...)

## S3 method for class 'bwqr_fit_multi'
plot(x, ...)
```

Arguments

`x`	Object of class `bqr.svy`.
`y`	Ignored (S3 signature).

`type`	One of `"fit"`, `"quantile"`, `"trace"`, `"density"`.
`tau`	Quantile(s) to plot; must appear in `x$quantile`. If `NULL`, all available are used.
`which`	Variable name(s) or coefficient index(es) to display. For `type = "fit"`, the name of a numeric predictor to plot on the x-axis (if `NULL`, the first numeric predictor is used). For `type = "quantile"`, a character vector of coefficient names or integer vector of indices; when more than one is given the plot uses `facet_wrap` to show all coefficients in a single figure. For `type = "trace"` and `type = "density"`, a single coefficient name or index (default: first coefficient in the model).
`add_points`	(fit) Logical; overlay observed data points.
`combine`	(fit) Logical; if multiple `tau`: `TRUE` overlays curves in one panel; `FALSE` uses one panel per quantile.
`show_ci`	(fit) Logical; draw credible bands.
`ci_probs`	(fit) Length-2 numeric vector with lower/upper probabilities for credible bands.
`at`	(fit) Named list of fixed values for non-plotted covariates (see Details).
`grid_length`	(fit) Integer; number of points in the predictor grid.
`points_alpha`	(fit) Point transparency in `[0,1]`.
`point_size`	(fit) Point size.
`line_size`	(fit/quantile) Line width for fitted/summary lines.
`main`	Optional main title.
`use_ggplot`	Logical; if `TRUE`, return a ggplot object.
`theme_style`	(ggplot) One of `"minimal"`, `"classic"`, `"bw"`, `"light"`.
`color_palette`	(ggplot) One of `"viridis"`, `"plasma"`, `"set2"`, `"dark2"`.
`add_h0`	(quantile) Logical; add a horizontal reference at $y = 0$.
`add_ols`	(quantile) Logical; add the OLS estimate (dotted line) for the selected coefficient.
`ols_fit`	(quantile) Optional precomputed `lm` object; if `NULL`, an `lm()` is fitted internally using `x$model` and `x$terms`.
`ols_weights`	(quantile) Optional numeric vector of weights when fitting OLS internally (length must match `nrow(x$model)`).
`...`	Accepted for compatibility; ignored by internal plotting code.

Details

Supported plot types:

- `type = "fit"`: Fitted quantile curves versus a single numeric predictor (selected via `which`). Optionally overlay observed points and credible bands. Other covariates can be held fixed via `at`.
- `type = "quantile"`: A single coefficient as a function of the quantile τ. Optionally add a reference line at 0 and the corresponding OLS estimate.
- `type = "trace"`: MCMC trace for one selected coefficient at a chosen τ.

- `type = "density"`: Posterior density for one selected coefficient at a chosen τ.

Notes:

- `tau` must be included in `x$quantile`. If `NULL`, all available quantiles in the object are used.
- For `type = "fit"`, `which` must name a numeric column in the original model. If `NULL`, the first numeric predictor (different from the response) is chosen automatically.
- For `type = "fit"`, `at` is a named list (`list(var = value, ...)`) used to fix other covariates while plotting versus the selected predictor. Provide valid levels for factors.
- When `use_ggplot = TRUE`, a ggplot object is returned and the appearance is controlled by `theme_style` and `color_palette`. Otherwise, base graphics are used and the function returns `invisible(NULL)`.

Value

`invisible(NULL)` for base R graphics, or a ggplot object if `use_ggplot = TRUE`.

Examples

```
data(mtcars)
fit <- bqr.svy(mpg ~ wt + hp + cyl, data = mtcars,
               quantile = c(0.5, 0.75), method = "ald",
               niter = 20000, burnin = 10000, thin = 5)

plot(fit, type = "fit", which = "wt", show_ci = TRUE)
plot(fit, type = "quantile", which = "wt", add_h0 = TRUE, add_ols = TRUE)
plot(fit, type = "quantile", which = c("(Intercept)", "wt", "hp", "cyl"),
     add_h0 = TRUE, add_ols = TRUE)
plot(fit, type = "trace", which = "wt", tau = 0.5)
plot(fit, type = "density", which = "wt", tau = 0.5)
```

`plotQuantileRegion` *Plot Bivariate Quantile Regions for Multiple-Output Models*

Description

Draws bivariate quantile regions from a fitted `mo.bqr.svy` object. The function projects data onto a grid, determines which points lie inside each quantile region, and visualises the result using **ggplot2**.

Usage

```
plotQuantileRegion(
  model,
  response,
  datafile,
  ngridpoints = 200,
```

```
  xValue = 1,
  paintedArea = TRUE,
  range_y = NULL,
  main = NULL,
  point_alpha = 0.3,
  point_size = 1.2,
  line_size = 0.8,
  verbose = FALSE
)
```

Arguments

`model`	An object of class `"mo.bqr.svy"` produced by `mo.bqr.svy`.
`response`	Character vector of length 2 giving the names of the response columns in `datafile`.
`datafile`	A data frame containing at least the two columns named in `response`.
`ngridpoints`	Integer; number of grid points per axis (default = 200).
`xValue`	Numeric vector of covariate values at which to evaluate the regression. For an intercept-only model use `xValue = 1`.
`paintedArea`	Logical; if `TRUE` (default) the regions are drawn as filled ribbons (see Details).
`range_y`	An optional 2×2 matrix whose rows give the `[min, max]` range for the first and second response, respectively. If `NULL`, the ranges are computed from the data with a 15 percent margin.
`main`	Optional character string for the plot title.
`point_alpha`	Numeric in `[0, 1]`; transparency for the observed-data point cloud (default = 0.3).
`point_size`	Numeric; size of the data points (default = 1.2).
`line_size`	Numeric; line width for contour outlines (default = 0.8).
`verbose`	Logical; if `TRUE`, print per-quantile progress messages (default = `FALSE`).

Details

Two display modes are available:

- `paintedArea = TRUE` (default): Filled ribbons with contour outlines, using a sequential `"Blues"` palette. An in-plot text legend shows the quantile level and approximate coverage.
- `paintedArea = FALSE`: Contour-only lines coloured by quantile, with a standard ggplot2 legend at the bottom.

The quantile regions are built from the directional approach described in Nascimento & Gonçalves (2025). For each quantile τ, the function evaluates every grid point against the half-space constraints defined by the fitted directions and orthogonal bases.

Value

Invisibly, a list with components:

`plot`	A `ggplot` object.
`data`	A data frame with columns `y1`, `min`, `max` and `tau`, one block per quantile level.

See Also

`mo.bqr.svy`, `prior`

Examples

```
# Load the Anthro dataset (children anthropometric data, already cleaned)
data(Anthro, package = "bayesQRsurvey")

# --- Directions ---
n_dir  <- 12
angles <- (0:(n_dir - 1)) * 2 * pi / n_dir
U_mat  <- rbind(cos(angles), sin(angles))
gamma_list <- lapply(seq_len(n_dir), function(k)
   matrix(c(-sin(angles[k]), cos(angles[k])), ncol = 1))

# --- Fit ---
fit <- mo.bqr.svy(
   cbind(wgt, hgt) ~ 1,
   data     = Anthro,
   quantile = c(0.05, 0.10, 0.25),
   U        = U_mat,
   gamma_U  = gamma_list,
   max_iter = 2000,
   verbose  = FALSE
)

# --- Plot quantile regions (filled ribbons) ---
plotQuantileRegion(fit, response = c("wgt", "hgt"), datafile = Anthro)

# Contour-only style
plotQuantileRegion(fit, response = c("wgt", "hgt"), datafile = Anthro,
                   paintedArea = FALSE)
```

`print.bayesQRsurvey` *Print methods for bayesQRsurvey model objects*

Description

`print.bayesQRsurvey` is an S3 method that prints the content of an S3 object of class `bqr.svy` or `mo.bqr.svy` to the console.

Usage

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

## S3 method for class 'mo.bqr.svy'
print(x, ...)
```

Arguments

`x`	An object of class `"bqr.svy"` or `"mo.bqr.svy"`, returned by `bqr.svy` or `mo.bqr.svy`.
`digits`	Integer specifying the number of decimal places to print. Defaults to `3`.
`...`	Additional arguments that are passed to the generic `print()` function.

Examples

```
set.seed(123)
N    <- 10000
x1_p <- runif(N, -1, 1)
x2_p <- runif(N, -1, 1)
y_p  <- 2 + 1.5 * x1_p - 0.8 * x2_p + rnorm(N)

# Generate sample data
n <- 500
z_aux <- rnorm(N, mean = 1 + y_p, sd = .5)
p_aux <- 1 / (1 + exp(2.5 - 0.5 * z_aux))
s_ind <- sample(1:N, n, replace = FALSE, prob = p_aux)
y_s   <- y_p[s_ind]
x1_s  <- x1_p[s_ind]
x2_s  <- x2_p[s_ind]
w     <- 1 / p_aux[s_ind]
data  <- data.frame(y = y_s, x1 = x1_s, x2 = x2_s, w = w)

# Fit a model
fit1 <- bqr.svy(y ~ x1 + x2, weights = w, data = data,
                niter = 2000, burnin = 500, thin = 2)

print(fit1)
```

`prior` *Create prior for Bayesian quantile regression models for complex survey data*

Description

`prior` creates prior distributions for both single (`bqr.svy`) and multiple-output (`mo.bqr.svy`) Bayesian quantile regression models for complex survey data.

Usage

```
prior(
  beta_x_mean = NULL,
  beta_x_cov = NULL,
  sigma_shape = 0.001,
  sigma_rate = 0.001,
  beta_y_mean = NULL,
```

```
  beta_y_cov = NULL
)
```

Arguments

`beta_x_mean`	vector of prior means for the regression coefficients. (default = NULL).
`beta_x_cov`	prior covariance matrix for the regression coefficients. (default = NULL).
`sigma_shape`	shape parameter for inverse Gamma prior for σ^2. (default = 0.001).
`sigma_rate`	rate parameter for inverse Gamma prior for σ^2. (default = 0.001).
`beta_y_mean`	prior means for the coefficients related to the variables that emerge from the product between the orthogonal basis and the outputs (default = NULL).
`beta_y_cov`	prior covariance matrix for the coefficients related to the variables that emerge from the product between the orthogonal basis and the outputs. (default = NULL).

Details

The function `prior` builds prior distributions for the three methods implemented in the function `bqr.svy` and for the multiple-output quantile regression implemented in the function `mo.bqr.svy`. Every nonspecified prior parameter will get the default value.

- `method` = `"ald"` in function `bqr.svy` allow the specification of hyperparameters `beta_x_mean`, `beta_x_cov`, `sigma_shape`, and `sigma_rate`.
- `method` = `"score"` in function `bqr.svy` allow the specification of hyperparameters `beta_x_mean` and `beta_x_cov`.
- `method` = `"approximate"` in function `bqr.svy` allow the specification of hyperparameters `beta_x_mean` and `beta_x_cov`.
- In function `mo.bqr.svy`, the specification of hyperparameters `beta_x_mean`,`beta_x_cov`, `sigma_shape`, `sigma_rate`, `beta_y_mean`, and `beta_y_cov` are allowed.

Value

An object of class `"prior"`.

See Also

`bqr.svy`, `mo.bqr.svy`, `summary`

Examples

```
# Simulate data
set.seed(123)
n  <- 200
x1 <- rnorm(n, 0, 1)
x2 <- runif(n, -1, 1)
w  <- runif(n, 0.5, 2)   # survey weights
```

```
y1 <- 2 + 1.5*x1 - 0.8*x2 + rnorm(n, 0, 1)
y2 <- 1 + 0.5*x1 - 0.2*x2 + rnorm(n, 0, 1)

data <- data.frame(y1 = y1, y2 = y2, x1 = x1, x2 = x2, w = w)

# Define a general informative prior
prior_general <- prior(
  beta_x_mean = c(2, 1.5, -0.8),
  beta_x_cov  = diag(c(0.25, 0.25, 0.25)),
  sigma_shape = 3,
  sigma_rate  = 2,
  beta_y_mean = 1,
  beta_y_cov  = 0.25
)

# Estimate the model parameters with informative prior
fit_ald <- bqr.svy(y1 ~ x1 + x2, weights = w, data = data,
                   prior = prior_general, method = "ald")

fit_scr <- bqr.svy(y1 ~ x1 + x2, weights = w, data = data,
                   prior = prior_general, method = "score")

fit_apx <- bqr.svy(y1 ~ x1 + x2, weights = w, data = data,
                   prior = prior_general, method = "approximate")

# Multiple-output method
fit_mo <- mo.bqr.svy(cbind(y1, y2) ~ x1 + x2, weights = w,
                     data = data, prior = prior_general, n_dir = 10)

plot(fit_ald, type = "trace", which = "x1", tau = 0.5)
plot(fit_ald, type = "trace", which = "x2", tau = 0.5)

print(fit_mo)
```

`summary.bayesQRsurvey` *Summary methods for bayesQRsurvey*

Description

summary.bayesQRsurvey is an S3 method that summarizes the output of the `bqr.svy` or `mo.bqr.svy` function. For the `bqr.svy` the posterior mean, posterior credible interval and convergence diagnostics are calculated. For the `mo.bqr.svy` the iterations for convergence, the MAP and the direction are calculated.

Usage

```
## S3 method for class 'bqr.svy'
summary(object, probs = c(0.025, 0.975), digits = 3, ...)
```

```
## S3 method for class 'mo.bqr.svy'
summary(object, digits = 4, ...)
```

Arguments

`object`	An object of class `mo.bqr.svy`.
`probs`	Two-element numeric vector with credible interval probabilities. Default `c(0.025, 0.975)`.
`digits`	Integer; number of decimals used by printing helpers. Default `4`.
`...`	Unused.

Value

An object of class `summary.bqr.svy` with one block per τ.

An object of class `summary.mo.bqr.svy`.

Index

∗ **datasets**
    Anthro, 2

Anthro, 2

bqr.svy, 3, *13, 14*

mo.bqr.svy, 5, *11–14*

plot (plot.bqr.svy), 8
plot.bqr.svy, 8
plotQuantileRegion, 10
print.bayesQRsurvey, 12
print.bqr.svy (print.bayesQRsurvey), 12
print.mo.bqr.svy (print.bayesQRsurvey), 12
prior, *3*, *6*, *12*, 13

summary, *14*
summary.bayesQRsurvey, 15
summary.bqr.svy (summary.bayesQRsurvey), 15
summary.mo.bqr.svy (summary.bayesQRsurvey), 15