Package 'refundBayes'

April 2, 2026

Encoding UTF-8

Type Package

Title Bayesian Regression with Functional Data

Version 0.5

Depends R (>= 3.6.0)

Imports brms, dplyr, ggplot2 (>= 2.0.0), mgcv, refund, rstan (>= 2.29.0), splines2, stats

Description Bayesian regression with functional data, including regression with scalar, survival, or functional outcomes. The package allows regression with scalar and functional predictors. Methods are described in Jiang et al. (2025) ``Tutorial on Bayesian Functional Regression Using Stan'' <doi:10.1002/sim.70265>.

NeedsCompilation no

BugReports https://github.com/ZirenJiang/refundBayes/issues

License GPL-2

URL https://github.com/ZirenJiang/refundBayes

RoxygenNote 7.3.3

Suggests knitr, rmarkdown

VignetteBuilder knitr

Author Ziren Jiang [aut, cre],
Erjia Cui [aut],
Ciprian Crainiceanu [ctb]

Maintainer Ziren Jiang <jian0746@umn.edu>

Repository CRAN

Date/Publication 2026-04-02 07:40:09 UTC

Contents

`fcox_bayes` *Bayesian Functional Cox Regression*

Description

Fit the Bayesian Functional Cox Regression model using Stan.

Usage

```
fcox_bayes(
  formula,
  data,
  cens,
  joint_FPCA = NULL,
  intercept = FALSE,
  runStan = TRUE,
  niter = 3000,
  nwarmup = 1000,
  nchain = 3,
  ncores = 1
)
```

Arguments

`formula`	Functional regression formula, with the same syntax as that in the R mgcv package.
`data`	A data frame containing data of all scalar and functional variables used in the model.
`cens`	A vector indicating censoring status (1 = event observed, 0 = censored). Must be the same length as the number of observations.
`joint_FPCA`	A True/False vector of the same length of the number of functional predictors, indicating whether jointly modeling FPCA for the functional predictors. Default to NULL.
`intercept`	True/False variable for whether include an intercept term in the linear predictor. Default to FALSE.
`runStan`	True/False variable for whether to run the Stan program. If False, the function only generates the Stan code and data.

niter	Total number of Bayesian iterations. Default to 3000.
nwarmup	Number of warmup (burnin) iterations for posterior sampling. Default to 1000.
nchain	Number of chains for posterior sampling. Default to 3.
ncores	Number of cores to use when executing the chains in parallel. Default to 1.

Details

The Bayesian Functional Cox model extends the scalar-on-function regression framework to survival outcomes with right censoring. The model is specified using similar syntax as in the R mgcv package.

Value

A list containing:

stanfit	The Stan fit object.
spline_basis	Basis functions used to reconstruct the functional coefficients from posterior samples.
stancode	A character string containing the code to fit the Stan model.
standata	A list containing the data to fit the Stan model.
int	A vector containing posterior samples of the intercept term (NULL for Cox models by default).
scalar_coef	A matrix containing posterior samples of scalar coefficients, where each row is one sample and each column is one variable.
func_coef	A list containing posterior samples of functional coefficients. Each element is a matrix, where each row is one sample and each column is one location of the functional domain.
baseline_hazard	Posterior samples of baseline hazard parameters.
family	Family type: "Cox".

Author(s)

Erjia Cui <ecui@umn.edu>, Ziren Jiang <jian0746@umn.edu>

References

Jiang, Z., Crainiceanu, C., and Cui, E. (2025). Tutorial on Bayesian Functional Regression Using Stan. *Statistics in Medicine*, 44(20-22), e70265.

Examples

```
## Not run:
## Not run: Stan requires a working C++ toolchain
# Functional Cox regression with right-censored survival outcome
set.seed(1)
n <- 150; L <- 50
```

```
Lindex    <- seq(0, 1, length.out = L)
X_func    <- matrix(rnorm(n * L), nrow = n)
age       <- rnorm(n)
beta_true <- cos(2 * pi * Lindex)
lp        <- X_func * beta_true / L + 0.3 * age
time      <- rexp(n, rate = exp(lp))
cens_time <- runif(n, min = 0.5, max = 3)
obs_time  <- pmin(time, cens_time)
cens_ind  <- as.integer(time <= cens_time)
dat        <- data.frame(obs_time = obs_time, age = age)
dat$X_func <- X_func
dat$Lindex <- matrix(rep(Lindex, n), nrow = n, byrow = TRUE)
fit <- fcox_bayes(
  formula = obs_time ~ age + s(Lindex, by = X_func, bs = "cr", k = 10),
  data = dat, cens = cens_ind, niter = 2000, nchain = 3
)
summary(fit)
plot(fit)

## End(Not run)
```

`fosr_bayes` *Bayesian Function-on-Scalar Regression*

Description

Fit the Bayesian Function-on-Scalar Regression (FOSR) model using Stan.

Usage

```
fosr_bayes(
  formula,
  data,
  joint_FPCA = NULL,
  runStan = TRUE,
  niter = 3000,
  nwarmup = 1000,
  nchain = 3,
  ncores = 1,
  spline_type = "bs",
  spline_df = 10
)
```

Arguments

`formula`	Functional regression formula, with the same syntax as that in the R mgcv package.
`data`	A data frame containing data of all scalar and functional variables used in the model.

joint_FPCA	A True/False vector of the same length of the number of functional predictors, indicating whether jointly modeling FPCA for the functional predictors. Default to NULL.
runStan	True/False variable for whether to run the Stan program. If False, the function only generates the Stan code and data.
niter	Total number of Bayesian iterations.
nwarmup	Number of warmup (burnin) iterations for posterior sampling.
nchain	Number of chains for posterior sampling. Default to 3.
ncores	Number of cores to use when executing the chains in parallel. Default to 1.
spline_type	Type of spline basis for modelling the residual process.
spline_df	Degrees of freedom for the spline basis for modelling the residual process.

Details

The Bayesian FOSR model is implemented following the tutorial by Jiang et al., 2025. The model is specified using the same syntax as in the R mgcv package.

Value

A list containing:

stanfit	The Stan fit object.
spline_basis	Basis functions used to reconstruct the functional coefficients from posterior samples.
stancode	A character string containing the code to fit the Stan model.
standate	A list containing the data to fit the Stan model.
int	A vector containing posterior samples of the intercept term.
scalar_coef	A matrix containing posterior samples of scalar coefficients, where each row is one sample and each column is one variable.
func_coef	A list containing posterior samples of functional coefficients. Each element is a matrix, where each row is one sample and each column is one location of the functional domain.
family	Distribution of the outcome variable.

Author(s)

Erjia Cui <ecui@umn.edu>, Ziren Jiang <jian0746@umn.edu>

References

Jiang, Z., Crainiceanu, C., and Cui, E. (2025). Tutorial on Bayesian Functional Regression Using Stan. *Statistics in Medicine*, 44(20-22), e70265.

Examples

```
## Not run:
## Not run: Stan requires a working C++ toolchain
# Function-on-scalar regression
set.seed(1)
n <- 100; M <- 50
tindex   <- seq(0, 1, length.out = M)
age      <- rnorm(n)
sex      <- rbinom(n, 1, 0.5)
Y_mat    <- outer(age, sin(2 * pi * tindex)) +
            outer(sex, cos(2 * pi * tindex)) +
            matrix(rnorm(n * M, sd = 0.3), nrow = n)
dat       <- data.frame(age = age, sex = sex)
dat$Y_mat <- Y_mat
fit <- fosr_bayes(
  formula = Y_mat ~ age + sex,
  data = dat, spline_type = "bs", spline_df = 10, niter = 2000
)
plot(fit)

## End(Not run)
```

---

plot.refundBayes *Plot the estimated functional coefficients with the corresponding credible interval(s).*

Description

Plot the estimated functional coefficients with the corresponding credible interval(s).

Usage

```
## S3 method for class 'refundBayes'
plot(x = NULL, ..., prob = 0.95, include = "both")
```

Arguments

x	A fitted object returned by `sofr_bayes()`.
...	Other parameters
prob	Coverage probability for the credible interval(s). Defaults to 0.95.
include	Type of interval to include. `"pointwise"` produces pointwise credible intervals; `"CMA"` produces the CMA credible band; `"both"` produces both. Defaults to `"both"`.

Value

A list of ggplot objects, one for each functional coefficient.

`sofr_bayes` *Bayesian Scalar-on-Function Regression*

Description

Fit the Bayesian Scalar-on-Function Regression (SoFR) model using Stan.

Usage

```
sofr_bayes(
  formula,
  data,
  family = gaussian(),
  joint_FPCA = NULL,
  intercept = TRUE,
  runStan = TRUE,
  niter = 3000,
  nwarmup = 1000,
  nchain = 3,
  ncores = 1
)
```

Arguments

`formula`	Functional regression formula, with the same syntax as that in the R mgcv package.
`data`	A data frame containing data of all scalar and functional variables used in the model.
`family`	Distribution of the outcome variable. Currently support "gaussian" and "binomial".
`joint_FPCA`	A True/False vector of the same length of the number of functional predictors, indicating whether jointly modeling FPCA for the functional predictors. Default to NULL.
`intercept`	True/False variable for whether include an intercept term in the linear predictor. Default to TRUE.
`runStan`	True/False variable for whether to run the Stan program. If False, the function only generates the Stan code and data.
`niter`	Total number of Bayesian iterations.
`nwarmup`	Number of warmup (burnin) iterations for posterior sampling.
`nchain`	Number of chains for posterior sampling. Default to 3.
`ncores`	Number of cores to use when executing the chains in parallel. Default to 1.

Details

The Bayesian SoFR model is implemented following the tutorial by Jiang et al., 2025. The model is specified using the same syntax as in the R mgcv package.

Value

A list containing:

`stanfit`	The Stan fit object.
`spline_basis`	Basis functions used to reconstruct the functional coefficients from posterior samples.
`stancode`	A character string containing the code to fit the Stan model.
`standate`	A list containing the data to fit the Stan model.
`int`	A vector containing posterior samples of the intercept term.
`scalar_coef`	A matrix containing posterior samples of scalar coefficients, where each row is one sample and each column is one variable.
`func_coef`	A list containing posterior samples of functional coefficients. Each element is a matrix, where each row is one sample and each column is one location of the functional domain.
`family`	Distribution of the outcome variable.

Author(s)

Erjia Cui <ecui@umn.edu>, Ziren Jiang <jian0746@umn.edu>

References

Jiang, Z., Crainiceanu, C., and Cui, E. (2025). Tutorial on Bayesian Functional Regression Using Stan. *Statistics in Medicine*, 44(20-22), e70265.

Examples

```
## Not run:
## Not run: Stan requires a working C++ toolchain
# Gaussian scalar-on-function regression
set.seed(1)
n <- 100; L <- 50
Lindex    <- seq(0, 1, length.out = L)
X_func    <- matrix(rnorm(n * L), nrow = n)
age       <- rnorm(n)
beta_true <- sin(2 * pi * Lindex)
Y  <- X_func
dat        <- data.frame(Y = Y, age = age)
dat$X_func <- X_func
dat$Lindex <- matrix(rep(Lindex, n), nrow = n, byrow = TRUE)
fit <- sofr_bayes(
  formula = Y ~ age + s(Lindex, by = X_func, bs = "cr", k = 10),
  data = dat, family = "gaussian", niter = 2000, nchain = 3
)
summary(fit)
plot(fit)

## End(Not run)
```

summary.refundBayes *Generate the summary table for the Bayesian model*

Description

Generate the summary table for the Bayesian model

Usage

```
## S3 method for class 'refundBayes'
summary(object = NULL, ..., prob = 0.95)
```

Arguments

object	A fitted object returned by sofr_bayes().
...	Other parameters
prob	Coverage probability for the reported confidence intervals. Defaults to 0.95.

Value

A list of two objects, the first is the summary table for the estimated scalar coefficients, the second is the plots for the estimated functional coefficients.

Index