Package: nlfh (via r-universe)

June 12, 2026

Title Nonlinear Fay-Herriot Models for Small Area Estimation

Version 0.1.0

Description Fits nonlinear Bayesian extensions of the Fay-Herriot model for small area estimation using area-level direct estimates and corresponding sampling variances. The package provides model fitting, prediction, uncertainty summaries, and diagnostic tools for nonlinear small area estimation workflows.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.3

Imports dbarts

Suggests knitr, posterior, rmarkdown, spelling, testthat (>= 3.0.0)

Config/testthat/edition 3

URL https://github.com/paparker/nlfh

BugReports https://github.com/paparker/nlfh/issues

VignetteBuilder knitr

Depends R (>= 3.5)

LazyData true

NeedsCompilation no

Author Paul Parker [aut, cre]

Maintainer Paul Parker <paulparker@ucsc.edu>

Repository https://cran.r-universe.dev

Date/Publication 2026-06-12 10:00:14 UTC

RemoteUrl https://github.com/cran/nlfh

RemoteRef HEAD

RemoteSha dcb5fa9ba5063c4c16bb2e85d846c23cd57728cd

Contents

`acs_dat` *Example ACS Small Area Data*

Description

A small example data set for fitting nonlinear Fay-Herriot models. `MedInc` is the direct estimate response and `MedIncSE` is its standard error. Use `MedIncSE^2` as the sampling variance when fitting models.

Usage

```
acs_dat
```

Format

A data frame with 1,617 rows and 8 variables:

MedInc Area-level direct estimate of median income.

MedIncSE Standard error of `MedInc`.

SNAPRate Area-level SNAP participation rate covariate.

PovRate Area-level poverty rate covariate.

White Area-level proportion identifying as White.

Black Area-level proportion identifying as Black.

Hispanic Area-level proportion identifying as Hispanic.

Asian Area-level proportion identifying as Asian.

Examples

```
data(acs_dat)
head(acs_dat)
```

fit_fh *Fit a Fay-Herriot Model*

Description

Primary user-facing model fitting function for linear and nonlinear Fay-Herriot models. Use `method` to choose the mean-function model and `control` to pass method-specific tuning parameters.

Usage

```
fit_fh(
  y = NULL,
  x = NULL,
  sampling_variance = NULL,
  method = c("linear", "rnn", "bart"),
  control = list(),
  data = NULL,
  formula = NULL,
  X = NULL
)
```

Arguments

`y`	Numeric vector of area-level direct estimates for the matrix interface.
`x`, `X`	Numeric matrix or data frame of area-level covariates for the matrix interface. Rows must correspond to entries of `y`. Include an intercept column if one is desired. For `method = "bart"`, the first column is treated as a baseline/intercept column and is excluded from the BART splitting variables and `variable_importance`.
`sampling_variance`	Numeric vector of known sampling variances for `y`. With the formula interface, this may also be an unquoted column name from `data` or a length-one character string naming a column in `data`.
`method`	Character string selecting the model. Options are `"linear"` for the linear Fay-Herriot model, `"rnn"` for the random-weight neural network Fay-Herriot model, and `"bart"` for the BART Fay-Herriot model.
`control`	Named list of control parameters. Common controls are `n_iter`, `burn_in`, `progress`, `scale`, `prior_shape`, and `prior_rate`. Linear-specific control: `prior_beta_variance`. RNN-specific controls: `n_hidden` and `prior_beta_variance`. BART-specific controls: `n_bart_samples` and `n_trees`.
`data`	Optional data frame containing variables used by `formula` and, optionally, `sampling_variance`.
`formula`	Optional model formula such as `y ~ x1 + x2`. If the first argument is a formula, it is treated as `formula`. For nonlinear methods, the formula specifies the predictors available to the model; it does not imply an additive linear mean structure. Nonlinearities and interactions may be learned implicitly by the selected model.

Details

Formula inputs are parsed with `stats::model.frame()` and `stats::model.matrix()`. Factors are expanded using R's standard contrast and dummy-variable rules. An intercept is included when the formula includes one, which is the default for formulas such as `y ~ x1 + x2`; use `0 +` or `- 1` in the formula to omit it. Matrix inputs are used as supplied, so include an intercept column manually if one is desired.

When `scale = TRUE`, non-intercept covariates are centered and scaled before fitting. The default is `TRUE` for `method = "rnn"` and `FALSE` for the linear and BART methods. For BART, the first baseline/intercept column is never scaled. The RNN method also standardizes the response and sampling variances internally, then transforms returned posterior quantities back to the original response scale.

For `method = "rnn"` and `method = "bart"`, the formula identifies the predictors available to the nonlinear mean function. It does not impose an additive linear model. These methods estimate an unknown function `f(X)`, and nonlinearities or interactions may be learned implicitly by the model. `sampling_variance` is always supplied separately from the formula. When `sampling_variance` names a column in `data`, that column is excluded from `y ~ .` expansion.

For `method = "bart"`, the first column of the model matrix is treated as a baseline/intercept column. With the formula interface, this must be the default `(Intercept)` column; formulas that omit the intercept with `0 +` or `- 1` are rejected. With the matrix interface, put the baseline or intercept column first. BART variable importance is computed only for the remaining columns, so `fit$variable_importance` does not include the first column.

Value

An object inheriting from `nlfh_fit`. The first class identifies the fitted method: `nlfh_linear_fit`, `nlfh_rnn_fit`, or `nlfh_bart_fit`.

References

Parker, P. A. (2024). Nonlinear Fay-Herriot Models for Small Area Estimation Using Random Weight Neural Networks. *Journal of Official Statistics*, 40(2), 317-332. doi:10.1177/0282423X241244671

Parker, P. A. and Eideh, A. (2026). BART-FH: Flexible Nonlinear Modeling for Small Area Estimation. *Journal of Survey Statistics and Methodology*, 00, 1-18. doi:10.1093/jssam/smaf050

Examples

```
data(acs_dat)
acs_small <- as.data.frame(acs_dat[1:500, ])
example_control <- list(n_iter = 50, burn_in = 25, progress = FALSE)

fit_linear <- fit_fh(
  MedInc ~ SNAPRate + PovRate + White + Black + Hispanic + Asian,
  sampling_variance = MedIncSE^2,
  data = acs_small,
  method = "linear",
  control = example_control
)
```

```
X <- model.matrix(
  MedInc ~ SNAPRate + PovRate + White + Black + Hispanic + Asian,
  data = acs_small
)
fit_matrix <- fit_fh(
  y = acs_small$MedInc,
  X = X,
  sampling_variance = acs_small$MedIncSE^2,
  method = "linear",
  control = example_control
)

fit_rnn <- fit_fh(
  MedInc ~ .,
  sampling_variance = MedIncSE^2,
  data = acs_small,
  method = "rnn",
  control = example_control
)

fit_bart <- fit_fh(
  MedInc ~ SNAPRate + PovRate + White,
  sampling_variance = MedIncSE^2,
  data = acs_small,
  method = "bart",
  control = example_control
)

# The default formula intercept is the first model-matrix column and is not
# included in BART variable importance.
fit_bart$variable_importance
```

`fit_fh_bart` *Fit a BART Fay-Herriot Model*

Description

Fits a Bayesian Fay-Herriot model whose mean function is represented with Bayesian additive regression trees via the `dbarts` package.

Usage

```
fit_fh_bart(
  y = NULL,
  x = NULL,
  sampling_variance = NULL,
  formula = NULL,
  data = NULL,
  X = NULL,
```

```
  prior_shape = 0.01,
  prior_rate = 0.01,
  n_iter = 1000,
  burn_in = 500,
  n_bart_samples = 10,
  n_trees = 50,
  scale = FALSE,
  progress = TRUE
)
```

Arguments

y	Numeric vector of area-level direct estimates for the matrix interface. If the first argument is a formula, it is treated as `formula`.
x, X	Numeric matrix or data frame of area-level covariates for the matrix interface. Rows must correspond to entries of `y`. The first column is treated as a baseline/intercept column and excluded from the BART splitting variables and `variable_importance`.
sampling_variance	Numeric vector of known sampling variances for `y`. With the formula interface, this may also be an unquoted column name from `data` or a length-one character string naming a column in `data`.
formula	Optional model formula such as `y ~ x1 + x2`. For nonlinear models, the formula specifies the predictors available to the model; it does not imply an additive linear mean structure.
data	Optional data frame containing variables used by `formula` and, optionally, `sampling_variance`.
prior_shape	Non-negative scalar shape parameter for the inverse-gamma prior on the random-effect variance.
prior_rate	Non-negative scalar rate parameter for the inverse-gamma prior on the random-effect variance.
n_iter	Positive integer number of MCMC iterations.
burn_in	Positive integer number of initial MCMC iterations to discard.
n_bart_samples	Positive integer number of BART samples to draw per outer MCMC iteration.
n_trees	Positive integer number of trees used by `dbarts`.
scale	Logical; if `TRUE`, center and scale covariates after the first baseline/intercept column before fitting. The first column is never scaled.
progress	Logical; if `TRUE`, display a progress bar.

Details

Formula inputs are parsed with `stats::model.frame()` and `stats::model.matrix()`. Factors are expanded using R's standard contrast and dummy-variable rules. Formula inputs must include an intercept, which is the default. For this nonlinear method, the formula specifies the available predictors and does not impose an additive linear mean structure. The BART mean component estimates an unknown function `f(X)`.

The first model-matrix column is treated as a baseline/intercept column and is excluded from BART splitting variables. With the formula interface this is the default `(Intercept)` column; formulas

that omit the intercept with `0 +` or `- 1` are rejected. With the matrix interface, put the baseline or intercept column first. BART variable importance is computed only for the remaining columns.

Value

An object of class `nlfh_bart_fit` and `nlfh_fit`, a list with posterior draws for `predictions`, the BART mean function `mean`, random effects `random_effects`, random-effect variance `random_effect_variance`, `variable_importance`, the scalar `dic`, and MCMC metadata.

References

Parker, P. A. and Eideh, A. (2026). BART-FH: Flexible Nonlinear Modeling for Small Area Estimation. *Journal of Survey Statistics and Methodology*, 00, 1-18. doi:10.1093/jssam/smaf050

Examples

```
data(acs_dat)
acs_small <- as.data.frame(acs_dat[1:500, ])
fit <- fit_fh_bart(
  MedInc ~ SNAPRate + PovRate + White,
  sampling_variance = MedIncSE^2,
  data = acs_small,
  n_iter = 50,
  burn_in = 25,
  progress = FALSE
)
summary(fit)
fit$variable_importance
```

---

fit_fh_linear *Fit a Linear Fay-Herriot Model*

Description

Fits the basic Bayesian Fay-Herriot model with a linear mean function and area-level random effects.

Usage

```
fit_fh_linear(
  y = NULL,
  x = NULL,
  sampling_variance = NULL,
  formula = NULL,
  data = NULL,
  X = NULL,
  prior_beta_variance = 10000^2,
  prior_shape = 0.1,
  prior_rate = 0.1,
```

```
  n_iter = 1000,
  burn_in = 500,
  scale = FALSE,
  progress = TRUE
)
```

Arguments

y	Numeric vector of area-level direct estimates for the matrix interface. If the first argument is a formula, it is treated as formula.
x, X	Numeric matrix or data frame of area-level covariates for the matrix interface. Rows must correspond to entries of y. Include an intercept column if one is desired.
sampling_variance	Numeric vector of known sampling variances for y. With the formula interface, this may also be an unquoted column name from data or a length-one character string naming a column in data.
formula	Optional model formula such as y ~ x1 + x2. The formula interface requires data.
data	Optional data frame containing variables used by formula and, optionally, sampling_variance.
prior_beta_variance	Positive scalar prior variance for the regression coefficients.
prior_shape	Non-negative scalar shape parameter for the inverse-gamma prior on the random-effect variance.
prior_rate	Non-negative scalar rate parameter for the inverse-gamma prior on the random-effect variance.
n_iter	Positive integer number of MCMC iterations.
burn_in	Positive integer number of initial MCMC iterations to discard.
scale	Logical; if TRUE, center and scale non-intercept covariates before fitting. Intercept columns named (Intercept), Intercept, or intercept are not scaled.
progress	Logical; if TRUE, display a progress bar.

Details

Formula inputs are parsed with stats::model.frame() and stats::model.matrix(). Factors are expanded using R's standard contrast and dummy-variable rules. An intercept is included when the formula includes one, which is the default; matrix inputs are used as supplied.

Value

An object of class nlfh_linear_fit and nlfh_fit, a list with posterior draws for predictions, random_effect_variance, coefficients, mean, the scalar dic, and MCMC metadata.

Examples

```
data(acs_dat)
acs_small <- as.data.frame(acs_dat[1:500, ])
fit <- fit_fh_linear(
  MedInc ~ SNAPRate + PovRate + White + Black + Hispanic + Asian,
  sampling_variance = MedIncSE^2,
  data = acs_small,
  n_iter = 50,
  burn_in = 25,
  progress = FALSE
)
summary(fit)
```

---

fit_fh_rnn *Fit a Random-Weight Neural Network Fay-Herriot Model*

Description

Fits a Bayesian Fay-Herriot model whose mean function is represented by a fixed random hidden layer with logistic activation and sampled output-layer coefficients.

Usage

```
fit_fh_rnn(
  y = NULL,
  x = NULL,
  sampling_variance = NULL,
  formula = NULL,
  data = NULL,
  X = NULL,
  n_hidden = 200,
  prior_beta_variance = NULL,
  prior_shape = 0.1,
  prior_rate = 0.1,
  n_iter = 1000,
  burn_in = 500,
  scale = TRUE,
  progress = TRUE
)
```

Arguments

y	Numeric vector of area-level direct estimates for the matrix interface. If the first argument is a formula, it is treated as `formula`.
x, X	Numeric matrix or data frame of area-level covariates for the matrix interface. Rows must correspond to entries of `y`. Include an intercept column if one is desired.

`sampling_variance`
: Numeric vector of known sampling variances for `y`. With the formula interface, this may also be an unquoted column name from `data` or a length-one character string naming a column in `data`.

`formula`
: Optional model formula such as `y ~ x1 + x2`. For nonlinear models, the formula specifies the predictors available to the model; it does not imply an additive linear mean structure.

`data`
: Optional data frame containing variables used by `formula` and, optionally, `sampling_variance`.

`n_hidden`
: Positive integer number of hidden nodes in the random-weight neural network.

`prior_beta_variance`
: Optional positive scalar prior variance for the output-layer coefficients. When `NULL`, the coefficient variance is sampled with the original inverse-gamma update.

`prior_shape`
: Non-negative scalar shape parameter for the inverse-gamma prior on the random-effect variance.

`prior_rate`
: Non-negative scalar rate parameter for the inverse-gamma prior on the random-effect variance.

`n_iter`
: Positive integer number of MCMC iterations.

`burn_in`
: Positive integer number of initial MCMC iterations to discard.

`scale`
: Logical; if `TRUE`, center and scale non-intercept covariates before fitting. Intercept columns named `(Intercept)`, `Intercept`, or `intercept` are not scaled.

`progress`
: Logical; if `TRUE`, display a progress bar.

Details

Formula inputs are parsed with `stats::model.frame()` and `stats::model.matrix()`. Factors are expanded using R's standard contrast and dummy-variable rules. An intercept is included when the formula includes one, which is the default; matrix inputs are used as supplied. For this nonlinear method, the formula specifies the available predictors and does not impose an additive linear mean structure. The model estimates an unknown function `f(X)`.

The response and sampling variances are standardized internally before fitting the RNN. Posterior predictions, mean function draws, coefficients, random-effect variances, and DIC are transformed back to the original response scale before being returned.

Value

An object of class `nlfh_rnn_fit` and `nlfh_fit`, a list with posterior draws for `predictions`, `random_effect_variance`, `coefficient_variance`, hidden-layer `coefficients`, `mean`, the scalar `dic`, and MCMC metadata.

References

Parker, P. A. (2024). Nonlinear Fay-Herriot Models for Small Area Estimation Using Random Weight Neural Networks. *Journal of Official Statistics*, 40(2), 317-332. doi:10.1177/0282423X241244671

Examples

```
data(acs_dat)
acs_small <- as.data.frame(acs_dat[1:500, ])
fit <- fit_fh_rnn(
  MedInc ~ .,
  sampling_variance = MedIncSE^2,
  data = acs_small,
  n_iter = 50,
  burn_in = 25,
  progress = FALSE
)
summary(fit)
```

`fitted.nlfh_fit` *Extract Fitted Values from a Nonlinear Fay-Herriot Model*

Description

For fitted `nlfh_fit` objects, fitted values are posterior summaries of the area-level quantities `theta_i`. The posterior draws are stored in the fitted object as an `n_areas` by `n_draws` matrix.

Usage

```
## S3 method for class 'nlfh_fit'
fitted(
  object,
  statistic = c("mean", "median"),
  summary = TRUE,
  full = FALSE,
  ...
)
```

Arguments

`object`	An object inheriting from class `nlfh_fit`.
`statistic`	Character string selecting the posterior point summary to return when `summary = TRUE`. Options are `"mean"` and `"median"`.
`summary`	Logical. If `TRUE`, return posterior summaries. If `FALSE`, return the posterior draw matrix for `theta_i`.
`full`	Logical. If `TRUE` and `summary = TRUE`, return a data frame with posterior mean, standard deviation, median, 2.5% quantile, and 97.5% quantile for each area.
`...`	Additional arguments passed to methods. Currently unused.

Value

If `summary = TRUE` and `full = FALSE`, a named numeric vector of posterior means or medians. If `summary = TRUE` and `full = TRUE`, a data frame of posterior summaries. If `summary = FALSE`, the posterior draw matrix for `theta_i`.

posterior_draws *Extract Posterior Draws from a Fay-Herriot Model*

Description

Extract posterior draws stored in fitted Bayesian Fay-Herriot model objects. Draws are returned with one row per retained MCMC draw and one column per requested parameter or area-level quantity.

Usage

```
posterior_draws(object, variable = c("theta", "beta", "u", "A"), ...)

## S3 method for class 'nlfh_fit'
posterior_draws(object, variable = c("theta", "beta", "u", "A"), ...)
```

Arguments

`object`	A fitted model object.
`variable`	Character string naming the posterior quantity to extract. Supported values are `"theta"` for area-level estimates, `"beta"` for regression or hidden-layer coefficients when available, `"u"` for area-level random effects when available, and `"A"` for the random-effect variance parameter.
`...`	Additional arguments passed to methods.

Value

A `posterior::draws_df` object when the `posterior` package is installed. Otherwise, a data frame with one row per posterior draw.

Examples

```
data(acs_dat)
acs_small <- as.data.frame(acs_dat[1:500, ])
fit <- fit_fh(
  MedInc ~ SNAPRate + PovRate + White + Black + Hispanic + Asian,
  sampling_variance = MedIncSE^2,
  data = acs_small,
  method = "linear",
  control = list(n_iter = 50, burn_in = 25, progress = FALSE)
)

theta_draws <- posterior_draws(fit, variable = "theta")
beta_draws <- posterior_draws(fit, variable = "beta")
```

print.nlfh_fit *Print a Fitted Nonlinear Fay-Herriot Model*

Description

Provides a compact overview of a fitted `nlfh_fit` object. The print method reports the model type, formula or input interface, number of areas, number of posterior draws, MCMC settings, a short variance-component summary, and DIC when available. It intentionally avoids printing posterior draw matrices, coefficient tables, or other large objects.

Usage

```
## S3 method for class 'nlfh_fit'
print(x, ...)
```

Arguments

`x`	An object inheriting from class `nlfh_fit`.
`...`	Additional arguments passed to methods. Currently unused.

Value

The input object `x`, invisibly.

print.summary.nlfh_fit *Print a Summary of a Fitted Nonlinear Fay-Herriot Model*

Description

Prints a compact view of a `summary.nlfh_fit` object. Large area-level and coefficient summaries are truncated to their first rows.

Usage

```
## S3 method for class 'summary.nlfh_fit'
print(x, digits = 4, max_rows = 6, ...)
```

Arguments

`x`	An object returned by `summary.nlfh_fit()`.
`digits`	Number of significant digits to print.
`max_rows`	Maximum number of rows to print from coefficient and area-level summaries.
`...`	Additional arguments passed to methods. Currently unused.

Value

The input object `x`, invisibly.

`summary.nlfh_fit` *Summarize a Fitted Nonlinear Fay-Herriot Model*

Description

Computes posterior summaries from the MCMC draws stored in a fitted `nlfh_fit` object. The returned object is structured for downstream use and is not printed directly by `summary()`.

Usage

```
## S3 method for class 'nlfh_fit'
summary(object, ...)
```

Arguments

`object`	An object inheriting from class `nlfh_fit`.
`...`	Additional arguments passed to methods. Currently unused.

Details

Posterior summaries include the mean, standard deviation, median, 2.5% quantile, and 97.5% quantile. Summaries are computed for area-level estimates `theta_i`, variance parameters, and regression or hidden-layer coefficients when those draws are available.

Value

An object with classes `summary.nlfh_fit` and `summary`, containing:

call Original model call.

method Model fitting method.

formula Formula, if the formula interface was used.

model_type Human-readable model type.

mcmc Stored MCMC settings and draw counts.

diagnostics Available MCMC diagnostics and metadata.

coefficients Posterior summaries of coefficient draws, if present.

variance Posterior summaries of variance parameters.

areas Posterior summaries of area-level estimates `theta_i`.

dic DIC, if available.

variable_importance BART variable-importance proportions, if available. The first model-matrix column is treated as a baseline/intercept column and is excluded.

Index