Package 'smoothbp'

June 16, 2026

Title Hierarchical Piecewise Regression with Smoothed Change-Points

Version 0.2.7

Date 2026-06-15

Description Fits Bayesian hierarchical piecewise regression models with multiple logistic-smoothed change-points. Non-linear parameters (change-point locations and transition sharpness) and linear parameters can each be conditioned on covariates and factors via flexible design matrices. A random-intercept structure is supported for any parameter. Spike-and-slab regularization is supported for selecting the number of breakpoints. Posterior inference uses a Metropolis-within-Gibbs sampler implemented in 'Rust' for speed. Methods are based on the smooth transition piecewise regression model of Bacon and Watts (1971) <doi:10.2307/2334389> and variable selection spike-and-slab priors of Kuo and Mallick (1998) <https://www.jstor.org/stable/25053023>.

License MIT + file LICENSE

Encoding UTF-8

Language en-GB

RoxygenNote 7.3.3

Config/rextendr/version 0.5.0

SystemRequirements Cargo (Rust's package manager), rustc >= 1.65.0

Depends R (>= 4.2)

Imports posterior, ggplot2, stats, bridgesampling, loo, bayesplot

Suggests testthat (>= 3.0.0), withr, knitr, rmarkdown, brms, mcp, dplyr, gt, tidyr, scales, rjags

Config/testthat/edition 3

VignetteBuilder knitr

URL https://github.com/ABindoff/smoothbp

BugReports https://github.com/ABindoff/smoothbp/issues

NeedsCompilation yes

Author Aidan D Bindoff [aut, cre] (ORCID:
<https://orcid.org/0000-0002-0943-2702>)

Maintainer Aidan D Bindoff <aidan.bindoff@utas.edu.au>

Repository CRAN

Date/Publication 2026-06-16 07:20:02 UTC

Contents

as.data.frame.smoothbp_fit *Convert draws to a data frame*

Description

Convert draws to a data frame

Usage

```
## S3 method for class 'smoothbp_fit'
as.data.frame(x, ...)
```

Arguments

x	A smoothbp_fit object.
...	Passed to as.data.frame.draws_df.

Value

A data.frame containing the posterior draws of the model parameters.

bayes_factor.smoothbp_fit *Bayes Factor for smoothbp_fit*

Description

Bayes Factor for smoothbp_fit

Usage

```
## S3 method for class 'smoothbp_fit'
bayes_factor(x1, x2, log = FALSE, ...)
```

Arguments

x1	A smoothbp_fit object.
x2	A smoothbp_fit object.
log	Logical; if TRUE, return log Bayes Factor.
...	Passed to bridge_sampler.

Value

A numeric value representing the Bayes Factor (or log Bayes Factor if log = TRUE) comparing x1 to x2.

bridge_sampler.smoothbp_fit *Bridge Sampler for smoothbp_fit*

Description

Bridge Sampler for smoothbp_fit

Usage

```
## S3 method for class 'smoothbp_fit'
bridge_sampler(
  samples,
  method = c("auto", "rust", "bridgesampling"),
  seed = 42,
  ...
)
```

Arguments

samples	A smoothbp_fit object.
method	Character; either "auto", "rust", or "bridgesampling". Default "auto" uses Rust for continuous models.
seed	Random seed for the bridge sampler.
...	Passed to bridge_sampler.

Value

An object of class "bridge" or "bridge_list" containing the log marginal likelihood estimate.

derivative *Posterior derivative of the conditional mean*

Description

Computes the d-th derivative of the posterior conditional mean $\partial^d \mu / \partial \tau^d$ at each row of newdata, propagating full posterior uncertainty via central finite differences applied to the posterior mean function.

The τ-independent terms $b_0 + u_{0,j}$ vanish from all finite-difference stencils of order ≥ 1 and are not evaluated. Orders 1–4 are supported. For higher orders, apply finite differences to the output of derivative(order = d - 1).

Usage

```
derivative(object, ...)

## S3 method for class 'smoothbp_fit'
derivative(
  object,
  newdata,
  order = 1L,
  probs = c(0.025, 0.975),
  h = NULL,
  draws = FALSE,
  ...
)

## S3 method for class 'smoothbp_ss_fit'
derivative(
  object,
  newdata,
  order = 1L,
  probs = c(0.025, 0.975),
  h = NULL,
  draws = FALSE,
  ...
)
```

Arguments

object	A fitted smoothbp_fit or smoothbp_ss_fit.
...	Unused.
newdata	A data frame with the time variable and any covariates required by the model formulae. Include the grouping column to condition on subject-level change-point timing; omit it for population-level derivatives (subject random effects on ω set to zero).
order	Positive integer; order of the derivative. Default 1L. Orders 1–4 are supported via central finite differences.
probs	Length-2 numeric vector of credible-interval probabilities. Default c(0.025, 0.975).
h	Step size for numerical differentiation. NULL (default) uses 10^{-4} times the range of the training τ values.
draws	Logical; if TRUE return the full $S \times N$ matrix of per-draw derivative estimates rather than a summary. Default FALSE.

Value

If draws = FALSE (default): a data.frame with columns tau, estimate (posterior mean derivative), and credible-interval bounds named from probs (e.g. Q2.5, Q97.5). Rows correspond to rows of newdata.

If `draws = TRUE`: an $S \times N$ numeric matrix of per-draw derivative estimates.

Examples

```
## Not run:
nd <- data.frame(tau = seq(0, 10, by = 0.1))

# Population-level rate of change (1st derivative)
dfit <- derivative(fit, newdata = nd)

# Curvature (2nd derivative)
dfit2 <- derivative(fit, newdata = nd, order = 2)

# Subject-level rate of change
nd_subj <- data.frame(tau = seq(0, 10, by = 0.1), subject = "s01")
dfit_subj <- derivative(fit, newdata = nd_subj)

# Full posterior draws for custom summaries
drmat <- derivative(fit, newdata = nd, draws = TRUE)
apply(drmat, 2, median)

## End(Not run)
```

`fitted.smoothbp_fit` *Fitted values for smoothbp_fit objects*

Description

Fitted values for smoothbp_fit objects

Usage

```
## S3 method for class 'smoothbp_fit'
fitted(object, newdata = NULL, summary = TRUE, ...)
```

Arguments

`object`	A `smoothbp_fit` object.
`newdata`	Optional data frame for prediction.
`summary`	Logical; if `TRUE` (default), returns the mean and 95% CI of the fitted values.
`...`	Unused.

Value

If `summary = TRUE`, a `data.frame` containing the observation index, mean fitted value, and 95% CI bounds. If `summary = FALSE`, a matrix of dimension `S x N` where `S` is the number of posterior draws and `N` is the number of observations, containing the posterior draws of the fitted values.

fixed — *Fix a parameter at a specific value*

Description

Used within omega or rho lists in smoothbp() to specify that a parameter is fixed and should not be estimated.

Usage

```
fixed(value)
```

Arguments

value	The fixed value(s) (numeric scalar or vector).

Value

A smoothbp_fixed object.

hypothesis — *Test hypotheses and compute evidence ratios from posterior draws*

Description

Evaluates one or more hypothesis expressions against the posterior draws of a smoothbp_fit object, returning posterior probabilities and evidence ratios.

Usage

```
hypothesis(object, hypotheses, ci = 0.95, ...)
```

Arguments

object	A smoothbp_fit object.
hypotheses	A character vector of hypothesis strings.
ci	Width of the credible interval ($0 < ci < 1$). Default 0.95.
...	Unused.

Value

An object of class `c("smoothbp_hypothesis", "data.frame")` with one row per hypothesis and columns:

`Hypothesis` The original hypothesis string.

`Estimate` Posterior mean of the contrast.

`Est.Error` Posterior SD of the contrast.

`CI.lower`, `CI.upper` Credible interval bounds.

`P(H)` Posterior probability of the hypothesis.

`Evid.Ratio` Evidence ratio $P(H)/(1 - P(H))$.

`Star` Informal star coding based on the evidence ratio.

Hypothesis syntax

Write hypotheses as character strings using exact parameter names as they appear in `fit$param_names` (e.g. `"b2_(Intercept)"`, `"omega_(Intercept)"`). No backtick-quoting is needed; the function handles special characters internally.

Two forms are accepted:

Directional An expression containing `>`, `<`, `>=`, or `<=`. The hypothesis is evaluated as a contrast: the left-hand side minus the right-hand side (direction-adjusted), and $P(H \mid \text{data})$ is the proportion of posterior draws satisfying the condition. Examples: `"b2_(Intercept) > 0"`, `"omega_(Intercept) < 4"`, `"b2_(Intercept) - b1_(Intercept) > 0"`.

Contrast A numeric expression without a comparison operator. The function summarises the posterior distribution of the derived quantity and reports $P(\text{expression} > 0)$ as the directional probability. Example: `"b2_(Intercept) - b1_(Intercept)"`.

Point-null hypotheses (`==`) are not supported because they require the Savage-Dickey density ratio; use `bayestestR::rope()` for interval-based equivalence testing instead.

Evidence ratio interpretation

$$ER = \frac{P(H \mid \text{data})}{1 - P(H \mid \text{data})}$$

An ER of 19 corresponds to $P(H) = 0.95$; ER = 1 means the posterior is equally split. Star codes: *** ER > 99, ** ER > 19, * ER > 3.

Examples

```
## Not run:
# Is the change in slope positive?
hypothesis(fit, "b2_(Intercept) > 0")

# Does the change-point fall before time 4?
hypothesis(fit, "omega_(Intercept) < 4")
```

```
# Multiple hypotheses at once
hypothesis(fit, c(
  "b2_(Intercept) > 0",
  "omega_(Intercept) < 4",
  "b2_(Intercept) - b1_(Intercept) > 0"
))

# Posterior summary of a contrast (no comparison operator)
hypothesis(fit, "b2_(Intercept) - b1_(Intercept)")

## End(Not run)
```

`log_lik` *Pointwise log-likelihood matrix*

Description

Pointwise log-likelihood matrix

Usage

```
log_lik(object, ...)

## S3 method for class 'smoothbp_fit'
log_lik(object, ...)
```

Arguments

`object`	A `smoothbp_fit` object.
`...`	Unused.

Value

A matrix of pointwise log-likelihood values of dimension `S` x `N`, where `S` is the number of posterior draws and `N` is the number of observations.

`model_methods` *Generate a statistical methods description for a smoothbp_fit object*

Description

Produces a human- and AI-readable description of the fitted model, including the structural equation, priors, MCMC sampling details, convergence diagnostics, and a reproducibility code snippet. The output is suitable for direct inclusion in the Statistical Analysis section of a scientific manuscript.

Usage

```
model_methods(object, ...)

## S3 method for class 'smoothbp_fit'
model_methods(object, width = 80, ...)
```

Arguments

object	A smoothbp_fit or smoothbp_ss_fit object.
...	Unused.
width	Integer; line-wrap width for the narrative paragraph (default 80).

Details

The function name model_methods() is used rather than methods() to avoid masking the base-R utils::methods() function, which lists S3/S4 methods for a generic and does not dispatch on object class.

When K = 0 (no breakpoints, i.e. deltas = list()), the model collapses to a Bayesian linear regression (with optional random intercepts) and the output reflects this accordingly.

Value

The full methods report as a single character string (invisibly). The text is also printed to the console via cat().

References

Bacon, D. W. & Watts, D. G. (1971). Estimating the transition between two intersecting straight lines. *Biometrika*, 58(3), 525–534. doi:10.2307/2334389

Kuo, L. & Mallick, B. (1998). Variable selection for regression models. *Sankhya: The Indian Journal of Statistics*, 60(1), 65–81.

Examples

```
## Not run:
fit <- smoothbp(y ~ tau, b0 = ~ 1 + (1 | subject), data = dat,
                chains = 4L, iter = 2000L, warmup = 1000L, seed = 42L)
model_methods(fit)
txt <- model_methods(fit)
cat(txt)

## End(Not run)
```

`model_results` *Report the results of a smoothbp_fit object*

Description

Produces a human- and AI-readable results report, including a plain-English narrative, the fitted model equation with posterior mean coefficients substituted to 3 d.p., per-breakpoint summaries, the full parameter table, and convergence diagnostics. The output is entirely self-contained: no package documentation or source code is needed to interpret it.

Usage

```
model_results(object, ...)

## S3 method for class 'smoothbp_fit'
model_results(object, digits = 3, width = 80, ...)
```

Arguments

`object`	A `smoothbp_fit` or `smoothbp_ss_fit` object.
`...`	Unused.
`digits`	Integer; decimal places for all numerical output (default `3`).
`width`	Integer; line-wrap width for narrative text (default `80`).

Details

The function name `model_results()` is used rather than `results()` for consistency with `model_methods()` and to avoid any namespace conflict with results functions in other packages.

For models with covariates on any structural parameter, the substituted equation shows the intercept-only (reference-level) value; the full set of coefficients is listed in the **PARAMETER ESTIMATES** section.

Value

The full results report as a single character string (invisibly). The text is also printed to the console via `cat()`.

Examples

```
## Not run:
fit <- smoothbp(y ~ tau, b0 = ~ 1 + (1 | subject), data = dat,
                chains = 4L, iter = 2000L, warmup = 1000L, seed = 42L)
model_results(fit)
txt <- model_results(fit)

## End(Not run)
```

pip *Posterior inclusion probabilities from a spike-and-slab fit*

Description

Posterior inclusion probabilities from a spike-and-slab fit

Usage

```
pip(object, ...)
```

Arguments

object	A smoothbp_ss_fit object.
...	Ignored.

Value

A named numeric vector of posterior inclusion probabilities (PIPs) for each b2 coefficient that had a spike-and-slab prior.

plot.smoothbp_fit *Trace and density plots for a smoothbp_fit*

Description

A thin wrapper around trace_plot for the standard plot() interface.

Usage

```
## S3 method for class 'smoothbp_fit'
plot(x, type = "trace", pars = NULL, ...)
```

Arguments

x	A smoothbp_fit object.
type	One of "trace" (default), "density", or "both".
pars	Character vector of parameter names. Defaults to all non-random-effect parameters.
...	Passed to trace_plot.

Value

A ggplot object, or a named list of two when type = "both".

plot.smoothbp_pip *Plot posterior inclusion probabilities*

Description

Plot posterior inclusion probabilities

Usage

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

Arguments

x	A smoothbp_pip object.
...	Unused.

Value

A ggplot object.

print.smoothbp_fit *Print a smoothbp_fit*

Description

Print a smoothbp_fit

Usage

```
## S3 method for class 'smoothbp_fit'
print(x, digits = 3, effects = c("fixed", "ran_pars"), ...)
```

Arguments

x	A smoothbp_fit object.
digits	Number of decimal places to print.
effects	Which effects to show: "fixed", "ran_pars", "ran_vals", or "all".
...	Unused.

Value

The input object x (invisibly), called for its printing side effects.

prior_gamma *Specify a gamma prior for a parameter*

Description

Specify a gamma prior for a parameter

Usage

```
prior_gamma(shape = 1, scale = 1)
```

Arguments

shape	Shape parameter (> 0).
scale	Scale parameter (> 0).

Value

A `smoothbp_prior` object.

prior_invgamma *Specify an inverse-gamma prior for a variance component*

Description

Specify an inverse-gamma prior for a variance component

Usage

```
prior_invgamma(shape = 1, scale = 1)
```

Arguments

shape	Shape parameter (> 0).
scale	Scale parameter (> 0).

Value

A `smoothbp_prior` object.

prior_normal — *Specify a normal (or truncated normal) prior for a regression coefficient*

Description

Specify a normal (or truncated normal) prior for a regression coefficient

Usage

```
prior_normal(mean = 0, sd = 1, lb = -Inf, ub = Inf)
```

Arguments

mean	Prior mean. Default 0.
sd	Prior standard deviation. Default 1.
lb	Lower bound (use -Inf for unconstrained). Default -Inf.
ub	Upper bound (use Inf for unconstrained). Default Inf.

Value

A smoothbp_prior object.

prior_spike_slab — *Specify a spike-and-slab prior for variable selection*

Description

Used with smoothbp_ss() to place a point-mass spike at zero on selected coefficients.

Usage

```
prior_spike_slab(
  pi = 0.5,
  slab = prior_normal(0, 2),
  learn_pi = FALSE,
  a = 1,
  b = 1
)
```

Arguments

pi	Prior inclusion probability. Default 0.5.
slab	A prior_normal() object for the slab component.
learn_pi	Logical; if TRUE, place a Beta(a, b) hyperprior on pi.
a	Shape parameter for the Beta hyperprior. Default 1.
b	Shape parameter for the Beta hyperprior. Default 1.

Value

A smoothbp_spike_slab object.

---

recovery_plot *Parameter recovery plot*

Description

Compares posterior estimates against the known data-generating values stored in attr(dat, "true_params") (as produced by simulate_smoothbp). For each matched parameter the plot shows the posterior mean, a credible interval, and the true value, coloured by whether the interval contains the truth.

Usage

```
recovery_plot(fit, dat, level = 0.95)
```

Arguments

fit	A smoothbp_fit object.
dat	The data frame used to fit fit, which must carry a "true_params" attribute (returned by simulate_smoothbp).
level	Credible interval width. Default 0.95.

Details

The function looks for population-level intercept parameters only. If the fitted model has covariates in a given component (e.g. omega = ~ 1 + group) the function still extracts the (Intercept) term for comparison against the scalar true value from the simulation.

Value

A ggplot object.

Examples

```
## Not run:
dat <- simulate_smoothbp(n_subj = 20, n_obs = 8, seed = 42)
fit <- smoothbp(y ~ tau, b0 = ~ 1 + (1 | subject), data = dat,
                priors = smoothbp_priors(omega = prior_normal(3, 2, lb = 0)),
                chains = 4L, iter = 2000L, warmup = 1000L, seed = 42L)
recovery_plot(fit, dat)

## End(Not run)
```

robustify — *Robustify a smoothbp_fit object using a Bayesian Sandwich approach*

Description

Robustify a smoothbp_fit object using a Bayesian Sandwich approach

Usage

```
robustify(object, cluster, ...)
```

Arguments

`object`	A `smoothbp_fit` object.
`cluster`	String naming the column in `object$data` defining the clusters (e.g., Subject ID).
`...`	Unused.

Value

A new `smoothbp_fit` object where the MCMC draws have been affinely transformed to match the robust clustered covariance matrix.

simulate_smoothbp — *Simulate data from the smooth change-point model*

Description

Generates synthetic data from the model used by `smoothbp`, including optional between-subject random intercepts. Supports any number of breakpoints $K \geq 1$. True parameter values are stored as the `"true_params"` attribute so they can be compared against posterior estimates.

Usage

```
simulate_smoothbp(
  n_subj = 20L,
  n_obs = 8L,
  b0 = 5,
  b1 = -0.3,
  delta = 1.2,
  omega = 3,
  rho = 4,
  sigma = 0.4,
  sigma_u = 0.5,
  tau_range = c(0, 6),
  seed = NULL
)
```

Arguments

n_subj	Number of subjects (groups). Set to 1 and sigma_u = 0 for a single-group simulation.
n_obs	Observations per subject. May be a scalar (same for all subjects) or a length-n_subj integer vector for unbalanced designs.
b0	Overall intercept (conditional mean at $\tau = \omega_1$).
b1	Pre-change-point slope (evaluated relative to ω_1).
delta	Change in slope at each change-point. A numeric vector of length K; a scalar is treated as $K = 1$.
omega	Change-point location(s). A numeric vector of length K, in ascending order. A scalar is treated as $K = 1$.
rho	Sharpness of each transition. A numeric vector of length K (all values must be positive); a scalar is recycled to length K.
sigma	Residual standard deviation.
sigma_u	Between-subject SD for random intercepts. Set to 0 to suppress random effects. Default 0.5.
tau_range	Numeric vector of length 2 giving the range of the time variable. Observations are evenly spaced within this range for each subject. Default c(0, 6).
seed	Integer seed for reproducibility. Sampled randomly if NULL (default).

Details

The data-generating model for K breakpoints is:

$$y_{ij} = (b_0 + u_j) + b_1(\tau_{ij} - \omega_1) + \sum_{k=1}^{K} \delta_k \, d_{ijk} \, \text{logistic}(d_{ijk} \, \rho_k) + \varepsilon_{ij}$$

where $d_{ijk} = \tau_{ij} - \omega_k$ and logistic$(\cdot)$ is the logistic function logistic$(x) = (1 + e^{-x})^{-1}$. The pre-break slope b_1 is centred at the first change-point ω_1, so b_0 represents the conditional mean at $\tau = \omega_1$ (consistent with the fitted model).

For a single breakpoint ($K = 1$), scalar values of omega, rho, and delta are accepted for backward compatibility.

Value

A data.frame with columns:

subject Subject identifier (factor).

tau Time variable.

mu Noise-free conditional mean μ_{ij}.

y Observed response.

The attribute "true_params" is a named list containing the data-generating values of b0, b1, delta, omega, rho, sigma, sigma_u, the vector of subject-level deviations u, and the seed used.

Examples

```
# Single breakpoint (K = 1)
dat1 <- simulate_smoothbp(
  n_subj = 20, n_obs = 8,
  b0 = 5, b1 = -0.3, delta = 1.2,
  omega = 3, rho = 4, sigma = 0.4, sigma_u = 0.5,
  seed = 42
)
head(dat1)
attr(dat1, "true_params")

# Two breakpoints (K = 2)
dat2 <- simulate_smoothbp(
  n_subj = 20, n_obs = 12,
  b0 = 5, b1 = -0.3,
  delta = c(1.2, -0.8),
  omega = c(2, 4),
  rho   = c(4, 4),
  sigma = 0.4, sigma_u = 0.5,
  seed = 42
)
head(dat2)
```

`smoothbp` *Fit a hierarchical piecewise regression model with smoothed change-points*

Description

Fit a hierarchical piecewise regression model with smoothed change-points

Usage

```
smoothbp(
  formula,
  b0 = ~1,
  b1 = ~1,
  deltas = list(~1),
  omega = list(~1),
  rho = list(~1),
  data,
  priors = smoothbp_priors(),
  chains = 4L,
  iter = 2000L,
  warmup = 1000L,
  seed = NULL,
  step_om = 0.3,
```

```
  step_rho = 0.3,
  target_accept = 0.9,
  cores = getOption("smoothbp.cores", 1L),
  hierarchical = NULL,
  reparameterise = c("none", "omega"),
  .verbose = TRUE
)
```

Arguments

formula	A two-sided formula identifying the response and time variable, e.g. value ~ tau.
b0	One-sided formula for the $b0$ linear predictor.
b1	One-sided formula for $b1$. Default ~ 1.
deltas	List of one-sided formulas for slope changes. Default list(~ 1).
omega	List of one-sided formulas for change-point locations. Default list(~ 1). Can also contain fixed() values for known change-points.
rho	List of one-sided formulas for transition sharpness. Default list(~ 1). Can also contain fixed() values for fixed sharpness.
data	A data frame.
priors	A smoothbp_priors object.
chains	Number of chains. Default 4.
iter	Total iterations per chain. Default 2000.
warmup	Warmup iterations. Default 1000.
seed	Random seed.
step_om	Initial HMC/MH step size for omega.
step_rho	Initial HMC/MH step size for rho.
target_accept	Target HMC acceptance probability (default 0.9). Raise toward 0.99 if you see divergent transitions in the sharpness (rho) parameter.
cores	Number of CPU cores.
hierarchical	**Deprecated.** Character vector; which parameters should be treated as hierarchical. This argument is no longer needed: random effects on change-point timing are auto-detected from (1|group) formula syntax (e.g. omega = list(~ 1 + (1|group))). Passing a non-NULL value generates a deprecation warning.
reparameterise	Character specifying the parameterisation for random change-points: "none" (centred) or "omega" (fully non-centred). Default is "none". Only used if random effects are present.
.verbose	Print progress.

Value

A smoothbp_fit object.

`smoothbp_priors` *Collect priors for all model parameters*

Description

Each argument accepts either:

- A single `prior_normal()` applied to all coefficients of that parameter, or
- A named list mapping coefficient names (matching column names of the design matrix) to individual `prior_normal()` objects.

Usage

```
smoothbp_priors(
  b0 = prior_normal(0, 10),
  b1 = prior_normal(0, 2),
  deltas = prior_normal(0, 2),
  omega = prior_normal(3, 2, lb = 0),
  rho = prior_normal(3, 2, lb = 0),
  sigma = prior_invgamma(1, 1),
  sigma_u = prior_invgamma(1, 1),
  sigma_re_om = prior_invgamma(1, 1),
  sigma_re_b1 = prior_invgamma(1, 1),
  sigma_re_deltas = prior_invgamma(1, 1)
)
```

Arguments

`b0`	Prior(s) for `b0` regression coefficients.
`b1`	Prior(s) for `b1` regression coefficients.
`deltas`	Prior(s) for slope change coefficients (one list per segment).
`omega`	Prior(s) for `omega` coefficients (one list per segment).
`rho`	Prior(s) for `rho` coefficients (one list per segment).
`sigma`	`prior_invgamma()` for residual SD.
`sigma_u`	`prior_invgamma()` for random-effect SD.
`sigma_re_om`	`prior_invgamma()` for random-effect SD on omega.
`sigma_re_b1`	`prior_invgamma()` for random-effect SD on b1.
`sigma_re_deltas`	`prior_invgamma()` for random-effect SD on deltas.

Details

For multi-breakpoint models, `deltas`, `omega`, and `rho` can also be **lists of prior specifications** (one per breakpoint slot). If a single specification is provided, it is applied to all slots.

Value

A `smoothbp_priors` list.

`smoothbp_ss` *Fit a smooth change-point model with spike-and-slab variable selection*

Description

Fit a smooth change-point model with spike-and-slab variable selection

Usage

```
smoothbp_ss(
  formula,
  b0 = ~1,
  b1 = ~1,
  deltas = list(~1),
  omega = list(~1),
  rho = list(~1),
  data,
  priors = smoothbp_priors(),
  spike = prior_spike_slab(),
  b1_spike = FALSE,
  hierarchical = NULL,
  chains = 4L,
  iter = 2000L,
  warmup = 1000L,
  seed = NULL,
  step_om = 0.3,
  step_rho = 0.3,
  target_accept = 0.9,
  cores = getOption("smoothbp.cores", 1L),
  reparameterise = c("none", "omega"),
  .verbose = TRUE
)
```

Arguments

`formula`	A two-sided formula.
`b0`	One-sided formula for b0.
`b1`	One-sided formula for b1.
`deltas`	List of formulas for slope changes.
`omega`	List of formulas for change-points. Can also contain `fixed()` values.
`rho`	List of formulas for sharpness. Can also contain `fixed()` values.

data	A data frame.
priors	A smoothbp_priors object.
spike	A prior_spike_slab() object.
b1_spike	Logical; should b1 coefficients be eligible for spike-and-slab?
hierarchical	Character vector specifying which parameters should be hierarchical. Currently only "omega" is supported.
chains	Number of chains.
iter	Total iterations.
warmup	Warmup iterations.
seed	Random seed.
step_om, step_rho, target_accept	HMC/MH tuning parameters.
cores	Number of CPU cores.
reparameterise	Character specifying the parameterisation for random change-points: "none" (centred) or "omega" (fully non-centred). Default is "none". Only used if random effects are present.
.verbose	Print progress.

Value

A smoothbp_fit object.

---

space_omega_priors *Generate evenly spaced priors for candidate breakpoints*

Description

This helper function generates a list of prior_normal objects for omega (breakpoint locations) that are evenly spaced across the range of your time variable tau. This is highly recommended when using smoothbp_ss() to ensure the candidate breakpoints cover the entire domain without clumping.

Usage

```
space_omega_priors(K, tau_min, tau_max)
```

Arguments

K	Number of candidate breakpoints.
tau_min	Minimum value of the time/covariate variable.
tau_max	Maximum value of the time/covariate variable.

Details

For hierarchical models where `omega` has random effects (e.g., `~ 1 + (1 | group)`), this function automatically names the prior (`Intercept`) so it applies correctly to the global market mean, while the random effects are handled automatically by the `sigma_re_om` shrinkage variance.

Value

A list of length `K` containing prior specifications for `omega`.

`summary.smoothbp_fit` *Summarise a smoothbp_fit*

Description

Summarise a smoothbp_fit

Usage

```
## S3 method for class 'smoothbp_fit'
summary(object, effects = c("fixed", "ran_pars"), digits = 3, ...)
```

Arguments

`object`	A `smoothbp_fit` object.
`effects`	Which effects to summarise: `"fixed"`, `"ran_pars"`, `"ran_vals"`, or `"all"`.
`digits`	Number of decimal places for rounding.
`...`	Unused.

Value

A `data.frame` containing summary statistics (mean, SD, 2.5% and 97.5% quantiles, Rhat, and bulk/tail ESS) for the requested effects.

`summary.smoothbp_ss_fit` *Summarise a smoothbp_ss_fit*

Description

Returns a data frame of posterior summaries for selected parameters.

Usage

```
## S3 method for class 'smoothbp_ss_fit'
summary(object, effects = "all", digits = 3, ...)
```

Arguments

object	A smoothbp_ss_fit object.
effects	Character vector controlling which parameters are included. Accepted values: "fixed" Population-level regression coefficients ($b0$, $b1$, $b2$, ω, ρ), residual SD, and spike-and-slab indicator variables (γ). "ran_pars" Random-effect variance parameter σ_u. "ran_vals" Individual group-level deviations u_j. "all" All of the above (default).
digits	Number of decimal places. Default 3.
...	Unused.

Value

A data frame with one row per selected parameter and columns variable, mean, sd, Q2.5, Q97.5, rhat, ess_bulk, ess_tail. For gamma (spike-and-slab) parameters, the mean column contains the posterior inclusion probability (PIP).

---

tab_smoothbp *Fixed-effects table for smoothbp_fit objects*

Description

Collects posterior summaries from one or more smoothbp_fit objects and displays them in a single table with parameters on rows and models in columns. All parameters present in any model are shown; models that do not include a given parameter display a dash.

Usage

```
tab_smoothbp(
  ...,
  labels = NULL,
  digits = 2,
  fmt = c("mean [CI]", "mean (SD)"),
  show_rhat = FALSE
)
```

Arguments

...	One or more smoothbp_fit or smoothbp_ss_fit objects.
labels	Character vector of column headers, one per model. If NULL (default) the deparsed call names are used.
digits	Integer; number of decimal places (default 2).
fmt	Cell format: "mean [CI]" (default) shows mean and 95% credible interval; "mean (SD)" shows mean and posterior SD.
show_rhat	Logical; append $\hat{R}$ to each cell (default FALSE).

Value

A gt_tbl object (or a knitr_kable if **gt** is not installed).

Examples

```
## Not run:
tab_smoothbp(m.ep.pw1, m.wo.pw1, m.la.pw1,
             labels = c("Episodic", "Working", "Language"))

## End(Not run)
```

`trace_plot` *Trace plots with automatic poor-mixing highlighting*

Description

Produces per-parameter trace plots from a smoothbp_fit object. Parameters with $\hat{R} > 1.05$ are flagged with a light-red background and their panel labels include the $\hat{R}$ value and a warning symbol. Parameters with low bulk-ESS (< 100) are further annotated.

Usage

```
trace_plot(
  fit,
  pars = NULL,
  type = "trace",
  rhat_thresh = 1.05,
  ess_thresh = 100
)
```

Arguments

fit	A smoothbp_fit object.
pars	Character vector of parameter names to include. Defaults to all non-random-effect parameters.
type	One of "trace" (default), "density", or "both".
rhat_thresh	Rhat threshold above which a parameter is flagged as poorly mixing. Default 1.05.
ess_thresh	Bulk-ESS threshold below which a parameter is flagged. Default 100.

Value

A ggplot object (or a named list of two when type = "both").

true_params *Print true parameters from a simulated dataset*

Description

Convenience function to display the data-generating parameters stored in the "true_params" attribute of a dataset returned by simulate_smoothbp.

Usage

```
true_params(dat)
```

Arguments

dat	A data.frame returned by simulate_smoothbp.

Value

The true_params list, invisibly.

update.smoothbp_fit *Update a fitted smoothbp model*

Description

Re-fits the model, replacing any arguments supplied here with the corresponding values stored in the original fit for anything left unspecified.

Usage

```
## S3 method for class 'smoothbp_fit'
update(
  object,
  formula,
  b0,
  b1,
  deltas,
  omega,
  rho,
  data,
  priors,
  chains,
  iter,
  warmup,
  seed,
  step_om,
```

```
  step_rho,
  target_accept,
  cores,
  .verbose = TRUE,
  ...
)
```

Arguments

`object`	A `smoothbp_fit` object.
`formula`, `b0`, `b1`, `deltas`, `omega`, `rho`, `data`, `priors`, `chains`, `iter`, `warmup`, `seed`, `step_om`, `step_rho`, `target_accept`, `cores`, `.verbose`	Replacements for the corresponding arguments of `smoothbp`. Any argument not supplied is taken from `object`, including the original `seed` and sampler tuning parameters (`step_om`, `step_rho`, `target_accept`). To use a fresh seed or different tuning, supply them explicitly.
`...`	Ignored.

Value

A new `smoothbp_fit` object.

Index