

Package ‘smoothbp’

June 2, 2026

Title Hierarchical Piecewise Regression with Smoothed Change-Points

Version 0.2.2

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](https://doi.org/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

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 2026-06-02 06:40:11 UTC

Contents

as.data.frame.smoothbp_fit	2
bayes_factor.smoothbp_fit	3
bridge_sampler.smoothbp_fit	4
fitted.smoothbp_fit	4
fixed	5
hypothesis	5
log_lik	7
pip	8
plot.smoothbp_fit	8
plot.smoothbp_pip	9
print.smoothbp_fit	9
prior_gamma	10
prior_invgamma	10
prior_normal	11
prior_spike_slab	11
recovery_plot	12
robustify	13
simulate_smoothbp	13
smoothbp	15
smoothbp_priors	16
smoothbp_ss	17
space_omega_priors	19
summary.smoothbp_fit	19
summary.smoothbp_ss_fit	20
tab_smoothbp	21
trace_plot	22
true_params	22
update.smoothbp_fit	23
Index	25

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.

```
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 $S \times N$ where S is the number of posterior draws and N is the number of observations, containing the posterior draws of the fitted values.

fixed	<i>Fix a parameter at a specific value</i>
-------	--

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	<i>Test hypotheses and compute evidence ratios from posterior draws</i>
------------	---

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 $S \times N$, where S is the number of posterior draws and N is the number of observations.

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	<i>Specify a gamma prior for a parameter</i>
-------------	--

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	<i>Specify an inverse-gamma prior for a variance component</i>
----------------	--

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	<i>Specify a normal (or truncated normal) prior for a regression coefficient</i>
--------------	--

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	<i>Specify a spike-and-slab prior for variable selection</i>
------------------	--

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 <code>prior_normal()</code> 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	<i>Parameter recovery plot</i>
---------------	--------------------------------

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

<code>fit</code>	A smoothbp_fit object.
<code>dat</code>	The data frame used to fit <code>fit</code> , which must carry a "true_params" attribute (returned by <code>simulate_smoothbp</code>).
<code>level</code>	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	<i>Robustify a smoothbp_fit object using a Bayesian Sandwich approach</i>
-----------	---

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	<i>Simulate data from the smooth change-point model</i>
-------------------	---

Description

Generates synthetic data from the model used by `smoothbp`, including optional between-subject random intercepts. 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,
  b2 = 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.
b1	Pre-change-point slope.
b2	Change in slope at the change-point.
omega	Change-point location (must be positive).
rho	Sharpness of the transition (must be positive; larger values give a sharper kink).
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 is:

$$y_{ij} = (b0 + u_j) + b1 \cdot d_{ij} + b2 \cdot d_{ij} \cdot \sigma(d_{ij} \cdot \rho) + \varepsilon_{ij}$$

where $d_{ij} = \tau_{ij} - \omega$ and $\sigma(\cdot)$ is the logistic sigmoid.

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, b2, omega, rho, sigma, sigma_u, the vector of subject-level deviations u, and the seed used.

Examples

```
dat <- simulate_smoothbp(
  n_subj = 20, n_obs = 8,
  b0 = 5, b1 = -0.3, b2 = 1.2,
  omega = 3, rho = 4, sigma = 0.4, sigma_u = 0.5,
  seed = 42
)
head(dat)
attr(dat, "true_params")
```

smoothbp	<i>Fit a hierarchical piecewise regression model with smoothed change-points</i>
----------	--

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 = 0.65,
  cores = getOption("smoothbp.cores", 1L),
  hierarchical = NULL,
  .verbose = TRUE
)
```

Arguments

formula	A two-sided formula identifying the response and time variable, e.g. <code>value ~ tau</code> .
b0	One-sided formula for the b_0 linear predictor.
b1	One-sided formula for b_1 . Default <code>~ 1</code> .
deltas	List of one-sided formulas for slope changes. Default <code>list(~ 1)</code> .
omega	List of one-sided formulas for change-point locations. Default <code>list(~ 1)</code> . Can also contain <code>fixed()</code> values for known change-points.
rho	List of one-sided formulas for transition sharpness. Default <code>list(~ 1)</code> . Can also contain <code>fixed()</code> values for fixed sharpness.
data	A data frame.
priors	A <code>smoothbp_priors</code> object.
chains	Number of chains. Default 4.

<code>iter</code>	Total iterations per chain. Default 2000.
<code>warmup</code>	Warmup iterations. Default 1000.
<code>seed</code>	Random seed.
<code>step_om</code>	Initial HMC/MH step size for omega.
<code>step_rho</code>	Initial HMC/MH step size for rho.
<code>target_accept</code>	Target HMC acceptance probability.
<code>cores</code>	Number of CPU cores.
<code>hierarchical</code>	Character vector; which parameters should be treated as hierarchical. Legacy argument, now mostly auto-detected from (1 group) formula syntax.
<code>.verbose</code>	Print progress.

Value

A `smoothbp_fit` object.

<code>smoothbp_priors</code>	<i>Collect priors for all model parameters</i>
------------------------------	--

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)
)
```

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.

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	<i>Fit a smooth change-point model with spike-and-slab variable selection</i>
-------------	---

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 = 0.65,
cores = getOption("smoothbp.cores", 1L),
.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 <code>fixed()</code> values.
rho	List of formulas for sharpness. Can also contain <code>fixed()</code> values.
data	A data frame.
priors	A <code>smoothbp_priors</code> object.
spike	A <code>prior_spike_slab()</code> 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.
.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

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

Details

For hierarchical models where `omega` has random effects (e.g., $\sim 1 + (1 | \text{group})$), 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 ($b_0, b_1, b_2, \omega, \rho$), 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	<i>Fixed-effects table for smoothbp_fit objects</i>
--------------	---

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

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

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	<i>Trace plots with automatic poor-mixing highlighting</i>
------------	--

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

<code>fit</code>	A <code>smoothbp_fit</code> object.
<code>pars</code>	Character vector of parameter names to include. Defaults to all non-random-effect parameters.
<code>type</code>	One of "trace" (default), "density", or "both".
<code>rhat_thresh</code>	Rhat threshold above which a parameter is flagged as poorly mixing. Default 1.05.
<code>ess_thresh</code>	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	<i>Print true parameters from a simulated dataset</i>
-------------	---

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.
 ... Ignored.

Value

A new `smoothbp_fit` object.

Index

`as.data.frame.smoothbp_fit`, 2

`bayes_factor.smoothbp_fit`, 3

`bridge_sampler`, 3, 4

`bridge_sampler.smoothbp_fit`, 4

`fitted.smoothbp_fit`, 4

`fixed`, 5, 15, 18

`hypothesis`, 5

`log_lik`, 7

`pip`, 8

`plot.smoothbp_fit`, 8

`plot.smoothbp_pip`, 9

`print.smoothbp_fit`, 9

`prior_gamma`, 10

`prior_invgamma`, 10

`prior_normal`, 11

`prior_normal()`, 11

`prior_spike_slab`, 11

`prior_spike_slab()`, 18

`recovery_plot`, 12

`robustify`, 13

`simulate_smoothbp`, 12, 13, 22

`smoothbp`, 13, 15, 23

`smoothbp()`, 5

`smoothbp_priors`, 15, 16, 18

`smoothbp_ss`, 17

`smoothbp_ss()`, 11

`space_omega_priors`, 19

`summary.smoothbp_fit`, 19

`summary.smoothbp_ss_fit`, 20

`tab_smoothbp`, 21

`trace_plot`, 8, 22

`true_params`, 22

`update.smoothbp_fit`, 23