Package ‘flexstanr’

July 17, 2026

Title Portable Backend Layer for 'Stan' Models

Version 0.1.0

Description Gives a 'Stan'-based R package one interface for fitting its models through either 'rstan' or (optionally) 'cmdstanr'. Collects and validates sampler options, guarding against mixing one backend's argument vocabulary into the other, dispatches the fit to the chosen backend, and exposes backend-agnostic accessors for reading posterior draws, extracting parameters, and running generated quantities. The host package supplies its own compiled models; flexstanr resolves them from the calling package at run time, so the same code works whichever backend is installed.

License MIT + file LICENSE

Encoding UTF-8

Language en-US

Depends R (>= 4.1.0)

Imports methods, rstan (>= 2.18.1), tools

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

VignetteBuilder knitr

Additional_repositories https://stan-dev.r-universe.dev

Config/testthat/edition 3

URL https://accidda.github.io/flexstanr/, https://github.com/ACCIDDA/flexstanr

BugReports https://github.com/ACCIDDA/flexstanr/issues

Config/roxygen2/version 8.0.0

NeedsCompilation no

Author Carl Pearson [aut, cre] (ORCID: <https://orcid.org/0000-0003-0701-7860>), Weston Voglesonger [aut]

Maintainer Carl Pearson <carl.ab.pearson@gmail.com>

Repository CRAN

Date/Publication 2026-07-17 14:20:08 UTC

Contents

`backend_draws_array` *Posterior draws of a fit as an iterations x chains x parameters array*

Description

Posterior draws of a fit as an iterations x chains x parameters array

Usage

```
backend_draws_array(raw_fit)
```

Arguments

`raw_fit`	a backend-native fit object (an rstan `stanfit` or a cmdstanr `CmdStanMCMC`).

Value

a 3-D array, dimensions iterations x chains x parameters.

Examples

```
## Not run:
draws <- backend_draws_array(fit)
dim(draws) # iterations x chains x parameters

## End(Not run)
```

backend_extract *Extract named parameters from a fit as a list of arrays*

Description

Matches the shape returned by `rstan::extract()`.

Usage

```
backend_extract(raw_fit, pars, ...)
```

Arguments

`raw_fit`	a backend-native fit object (an rstan `stanfit` or a cmdstanr `CmdStanMCMC`).
`pars`	character vector of parameter names to extract.
`...`	forwarded to the backend's extractor.

Value

a named list of draw arrays, one per parameter.

Examples

```
## Not run:
post <- backend_extract(fit, pars = c("beta", "sigma"))

## End(Not run)
```

backend_generate_quantities *Run generated quantities against a fit and return a parameter matrix*

Description

Run generated quantities against a fit and return a parameter matrix

Usage

```
backend_generate_quantities(
  raw_fit,
  data,
  draws_mat,
  pars,
  model_name = NULL,
  package = NULL
)
```

Arguments

raw_fit	a backend-native fit object (an rstan stanfit or a cmdstanr CmdStanMCMC).
data	the Stan data list for the generated-quantities run.
draws_mat	a draws matrix (rows = draws, columns = parameters). Used by the rstan backend; the cmdstanr backend runs generated quantities against the fit's own draws and ignores this argument.
pars	name of the generated parameter to return.
model_name	name of the model whose generated-quantities block to run. Required by the cmdstanr backend, which recompiles the model to run it; ignored by rstan, which reuses the model carried on raw_fit.
package	the host package the model belongs to; defaults to the calling package (see fit_model()). Only used by the cmdstanr backend.

Value

a matrix of the requested generated parameter (rows = draws).

Examples

```
## Not run:
gen <- backend_generate_quantities(fit, data = data_list,
                                   draws_mat = as.matrix(fit), pars = "y_rep")

## End(Not run)
```

---

backend_has_draws *Does a fit object carry usable posterior draws?*

Description

Detect the degenerate "no draws" case after a fit, so a caller can fail loudly instead of returning an empty fit. This is backend-aware: rstan returns a mode-2 stanfit with an empty @sim when the sampler fails to initialize (rather than erroring), while cmdstanr exposes its draws through $draws(). Unrecognized objects (e.g. test mocks) are treated as having draws so they pass through untouched.

Usage

```
backend_has_draws(raw_fit)
```

Arguments

raw_fit	a backend-native fit object (an rstan stanfit or a cmdstanr CmdStanMCMC).

Value

logical; TRUE if the fit carries usable draws.

Examples

```
# Unrecognized objects are treated as carrying draws (pass-through).
backend_has_draws(list())
```

---

fit_model *Fit a Stan model through the chosen backend*

Description

Dispatches a fit to the backend recorded on stan_opts (from stan_options()). The compiled model is resolved by model_name from the calling package: for "rstan", package::stanmodels[[model_name]]; for "cmdstanr", inst/stan/<model_name>.stan under package. The calling package is detected automatically and can be overridden with package.

Usage

```
fit_model(
  model_name,
  dat_stan,
  init,
  stan_opts,
  drop_pars = NULL,
  package = NULL
)
```

Arguments

model_name	name of the Stan model; used to look up the compiled model in the calling package's stanmodels (rstan) and to locate the .stan source file under its inst/stan/ (cmdstanr).
dat_stan	the Stan data list.
init	the init list, sized to the chain count.
stan_opts	the validated stan_options() list (carrying a backend element).
drop_pars	character vector of parameter names to exclude from the saved draws, or NULL to keep everything. Honored by rstan; cmdstanr cannot drop parameters and warns if any are requested.
package	name of the host package whose model is being fit. Defaults to the package that called fit_model(), which is correct for the usual case of a host package fitting one of its own models.

Value

the backend's fit object (a `stanfit` or `CmdStanMCMC`).

Examples

```
## Not run:
# From inside a host package that ships a compiled `coverage` model:
opts <- stan_options(chains = 2, iter = 500, seed = 1)
fit <- fit_model("coverage", dat_stan = data_list, init = init_list,
                 stan_opts = opts)

## End(Not run)
```

`stan_options` *Stan Sampler Options*

Description

Collects and validates sampler arguments for the chosen backend, forwarding them **verbatim** so calls feel native to that backend. Use the backend's own argument names; mixing one backend's vocabulary into the other errors with a hint. The model object is supplied separately (via `fit_model()`), while `data` and `init` are constructed internally, so none of these may be set here. `chains` defaults to 4 so downstream code can always size per-chain structures from it.

Usage

```
stan_options(..., chains = 4L, backend = "rstan")
```

Arguments

`...`	sampler arguments forwarded verbatim to the chosen backend's sampler. Use the backend's own names: for `"rstan"`, the `rstan::sampling()` arguments (`iter`, `cores`, `seed`); for `"cmdstanr"`, the `$sample()` arguments (`iter_warmup`, `iter_sampling`, `parallel_chains`, ...).
`chains`	A positive integer specifying the number of Markov chains. The default is 4.
`backend`	which Stan interface to target, one of `"rstan"` (default) or `"cmdstanr"`. Determines which argument vocabulary is accepted and which sampler `fit_model()` calls. Selecting `"cmdstanr"` errors if the cmdstanr package is not installed.

Value

a named list of validated sampler arguments, carrying a `backend` element recording the backend it was built for

Examples

```
stan_options()
stan_options(chains = 2, iter = 500)
if (requireNamespace("cmdstanr", quietly = TRUE)) {
  stan_options(backend = "cmdstanr", parallel_chains = 4, iter_warmup = 500)
}
```

`use_flexstanr` *Wire flexstanr into a host package*

Description

A one-time setup helper, in the spirit of usethis's `use_*` functions, that declares flexstanr as a dependency of the host package you run it from. It adds flexstanr (and rstan, the default backend) to the host's `Imports` and, until flexstanr is on CRAN, an interim `Remotes` entry so `remotes` / `pak` can install it from GitHub.

No further wiring is needed: the host calls `fit_model()` and the `backend_*` accessors directly, and flexstanr resolves the host's own compiled models automatically from the calling package.

Usage

```
use_flexstanr(path = ".", remote = "ACCIDDA/flexstanr", on_cran = FALSE)
```

Arguments

`path`	path to the host package's root (the directory containing its `DESCRIPTION`). Defaults to the working directory.
`remote`	the GitHub `owner/repo` used for the interim `Remotes` entry while flexstanr is pre-CRAN.
`on_cran`	set `TRUE` once flexstanr is on CRAN to skip the interim `Remotes` entry; a plain `Imports` dependency is then enough.

Value

the host package `path`, invisibly.

Examples

```
## Not run:
# from the root of your Stan package:
flexstanr::use_flexstanr()

## End(Not run)
```

Index