

Package: bonsaiforest2 (via r-universe)

October 15, 2026

Type Package

Title Flexible Bayesian Shrinkage Models for Subgroup Analysis

Version 0.1.0

Description An extension of the 'bonsaiforest' package that provides increased flexibility in prior selection and modeling options. It supports four types of endpoints and simplifies the fitting and interpretation of Bayesian models for subgroup analysis. Leveraging 'brms', it distinguishes between prognostic and predictive factors using differential shrinkage priors and generates publication-ready forest plots.

License MIT + file LICENSE

URL <https://openpharma.github.io/bonsaiforest2/>

BugReports <https://github.com/openpharma/bonsaiforest2/issues>

Encoding UTF-8

LazyData true

Imports brms, checkmate, dplyr, ggplot2, posterior, rlang, splines2, stats, stringr, survival, tibble

Roxygen list(markdown = TRUE)

Suggests BH, bookdown, cmdstanr, knitr, RcppEigen, rmarkdown, testthat (>= 3.0.0)

Config/testthat/edition 3

VignetteBuilder knitr

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

Depends R (>= 3.5)

Config/roxygen2/version 8.1.0

Config/pak/sysreqs cmake make libicu-dev

Repository <https://openpharma.r-universe.dev>

Date/Publication 2026-10-15 13:03:01 UTC

RemoteUrl <https://github.com/openpharma/bonsaiforest2>

RemoteRef HEAD
RemoteSha 5a6dbf386d371fdda50570c0061b0a0a14dfe47c

Contents

combine_summaries	2
estimate_subgroup_effects	3
fit_brms_model	4
plot.subgroup_summary	7
prepare_formula_model	8
run_brms_analysis	12
shrink_data	16
summary_subgroup_effects	17
Index	19

combine_summaries	<i>Combine Multiple Subgroup Summaries for Comparison Plotting</i>
-------------------	--

Description

Combines estimates from multiple subgroup summary objects into a single summary object that can be plotted to compare models.

Usage

```
combine_summaries(summary_list)
```

Arguments

summary_list Named list of subgroup_summary objects.

Value

subgroup_summary. Combined object with a "Model" column in estimates.

Examples

```
## Not run:
summary1 <- summary_subgroup_effects(brms_fit = model1)
summary2 <- summary_subgroup_effects(brms_fit = model2)

combined <- combine_summaries(list(
  "One-way model" = summary1,
  "Global" = summary2
))
plot(combined)

## End(Not run)
```

estimate_subgroup_effects

Estimate Marginal Subgroup Treatment Effects

Description

The function uses a counterfactual, marginal approach based on the posterior predictive distribution. It averages over all other covariates to provide robust estimates of subgroup-specific effects.

Usage

```
estimate_subgroup_effects(
  brms_fit,
  trt_var = NULL,
  data = NULL,
  subgroup_vars = "auto",
  response_type = NULL,
  ndraws = NULL,
  conf = 0.95
)
```

Arguments

brms_fit	A <code>brmsfit</code> object. Fitted model object from <code>fit_brms_model()</code> or <code>run_brms_analysis()</code> . Must contain the necessary attributes for extracting treatment variable and response type information.
trt_var	A character string or <code>NULL</code> . Treatment variable name. If <code>NULL</code> , automatically extracted from model attributes (set by <code>fit_brms_model()</code>). Must be a binary variable coded as 0/1 in the dataset.
data	A data frame or <code>NULL</code> . Dataset used for model fitting. If <code>NULL</code> , automatically extracted from model attributes (set by <code>fit_brms_model()</code>). This dataset is used for generating counterfactual predictions.
subgroup_vars	A character vector or "auto". Subgroup variable names for which to estimate treatment effects. If "auto" (default), automatically detects treatment interaction terms (colon syntax) and random effect grouping factors (pipe syntax) from all formula components (<code>unshrunktermeffect</code> , <code>shprogeffect</code> , <code>shpredeffect</code>).
response_type	A character string or <code>NULL</code> . Outcome type, one of "binary", "count", "continuous", or "survival". If <code>NULL</code> , automatically extracted from model attributes (set by <code>fit_brms_model()</code>). This determines the appropriate scale for effect estimation.
ndraws	An integer or <code>NULL</code> . Number of posterior draws to use for estimation. If <code>NULL</code> (default), all available posterior draws are used. Reducing this can speed up computation at the cost of precision.

conf A numeric scalar. Credible interval confidence level. Default is 0.95 (corresponding to 95% credible intervals). Example: use 0.90 for 90% credible intervals.

Details

This post-processing function estimates marginal treatment effects for specified subgroups from a fitted `brmsfit` object.

Value

`list` with two named elements:

estimates `tibble` where each row represents a subgroup (or "Overall" effect), with columns for **Subgroup**, **Median**, **CI_Lower**, and **CI_Upper** from posterior distribution

draws `data.frame` containing posterior draws for each subgroup

<code>fit_brms_model</code>	<i>Fit a Bayesian Hierarchical Model using brms</i>
-----------------------------	---

Description

This function serves as a wrapper to fit a Bayesian model using the `brms` package. It uses a formula and data prepared by `prepare_formula_model` and simplifies the process of assigning complex priors to the different non-linear components of the model.

Usage

```
fit_brms_model(
  prepared_model,
  intercept_prior = NULL,
  unshrunk_prior = NULL,
  shrunk_prognostic_prior = NULL,
  shrunk_predictive_prior = NULL,
  stanvars = NULL,
  ...
)
```

Arguments

prepared_model

A list object. Object returned from `prepare_formula_model()` containing the elements: **formula** (a `brmsformula` object), **data** (a modified `data.frame` with appropriate contrast coding), **response_type** (a character string), and **trt_var** (a character string). The **trt_var** element is stored as an attribute on the fitted model for use by downstream functions like `estimate_subgroup_effects()`.

<code>intercept_prior</code>	A character string, <code>brmsprior</code> object, or <code>NULL</code> . Prior specification for the model intercept in the <code>unshrunktermeffect</code> component. Not used for survival models (Cox models have no intercept).
<code>unshrunk_prior</code>	A character string, <code>brmsprior</code> object, or <code>NULL</code> . Prior specification for unshrunk terms in the <code>unshrunktermeffect</code> component (excludes intercept). These are main effects and interactions for which no regularization is desired.
<code>shrunk_prognostic_prior</code>	A character string, <code>brmsprior</code> object, or <code>NULL</code> . Prior specification for regularized prognostic effects in the <code>shprogeffect</code> component. These are main effects for which strong shrinkage/regularization is desired.
<code>shrunk_predictive_prior</code>	A character string, <code>brmsprior</code> object, or <code>NULL</code> . Prior specification for regularized predictive effects (treatment interactions) in the <code>shpredeffect</code> component. These are treatment interactions for which strong shrinkage/regularization is desired.
<code>stanvars</code>	A <code>stanvars</code> object or <code>NULL</code> . Custom Stan code created via <code>brms::stanvar()</code> for implementing hierarchical priors or other advanced Stan functionality. See the brms documentation for details on creating <code>stanvars</code> objects.
<code>...</code>	Additional arguments passed to <code>brms::brm()</code> . Common arguments include <code>chains</code> (number of MCMC chains), <code>iter</code> (number of iterations per chain), <code>cores</code> (number of CPU cores to use), <code>backend</code> (Stan backend), and <code>refresh</code> (progress update frequency).

Value

`brmsfit`. Fitted Bayesian model object with attributes: `response_type` (`character`), `model_data` (`data.frame`), and `trt_var` (`character`) for downstream analysis functions.

Prior Specification

Priors are specified separately for each model component using dedicated parameters. This provides explicit control and prevents errors. The function accepts prior definitions as character strings (e.g., `"normal(0, 2.5)"`) or as `brmsprior` objects for complex hierarchical or parameter-specific priors.

Available prior parameters:

- `intercept_prior`: Prior for the intercept in the unshrunk component
- `unshrunk_prior`: Prior for all unshrunk terms (main effects you trust)
- `shrunk_prognostic_prior`: Prior for shrunk prognostic effects (regularized)
- `shrunk_predictive_prior`: Prior for shrunk predictive effects (regularized)

Default Priors by Response Type and Model Structure:

If you don't specify priors, the function uses sensible defaults that adapt to the model structure:

For all response types (continuous, binary, count, survival):

- `intercept_prior`: NULL (brms default)
- `unshrunk_prior`: NULL (brms default)
- `shrunk_prognostic_prior`: `horseshoe(1)` (strong regularization)
- `shrunk_predictive_prior`: `horseshoe(1)` (strong regularization)

For models with random effects (One-way model) (pipe-pipe || syntax):

Random effects in shrunk predictive terms (e.g., `~ (0 + trt || subgroup)`) automatically receive `normal(0, 1)` priors on the standard deviation scale for each coefficient and group combination, regardless of the `shrunk_predictive_prior` setting. For example:

- `prior(normal(0, 1), class = "sd", coef = "trt", group = "subgroup")`

Example:

```
fit_brms_model(
  prepared_model = prepared,
  unshrunk_prior = "normal(0, 2.5)",
  shrunk_prognostic_prior = "horseshoe(1)",
  shrunk_predictive_prior = "horseshoe(1)"
)
```

Examples

```
if (require("brms") && require("survival")) {
  # 1. Create Sample Data
  set.seed(123)
  n <- 100
  sim_data <- data.frame(
    time = round(runif(n, 1, 100)),
    status = sample(0:1, n, replace = TRUE),
    trt = sample(0:1, n, replace = TRUE),
    age = rnorm(n, 50, 10),
    region = sample(c("A", "B"), n, replace = TRUE),
    subgroup = sample(c("S1", "S2", "S3"), n, replace = TRUE)
  )
  sim_data$trt <- factor(sim_data$trt, levels = c(0, 1))
  sim_data$region <- as.factor(sim_data$region)
  sim_data$subgroup <- as.factor(sim_data$subgroup)

  # 2. Prepare the formula and data using colon syntax with recommended ~ 0 + for shrunk terms
  prepared_model <- prepare_formula_model(
    data = sim_data,
    response_formula = Surv(time, status) ~ trt,
    shrunk_predictive_formula = ~ 0 + trt:subgroup,
    unshrunk_terms_formula = ~ age,
    shrunk_prognostic_formula = ~ 0 + subgroup,
    response_type = "survival",
    stratification_formula = ~ region
  )

  # 2b. Alternatively, using one-way model (random effects) with pipe-pipe notation
```

```

# prepared_model <- prepare_formula_model(
#   data = sim_data,
#   response_formula = Surv(time, status) ~ trt,
#   unshrunk_terms_formula = ~ age,
#   shrunk_predictive_formula = ~ (0 + trt || subgroup),
#   response_type = "survival",
#   stratification_formula = ~ region
# )

# 3. Fit the model
## Not run:
fit <- fit_brms_model(
  prepared_model = prepared_model,
  unshrunk_prior = "normal(0, 2)",
  shrunk_prognostic_prior = "horseshoe(scale_global = 1)",
  shrunk_predictive_prior = "horseshoe(scale_global = 1)",
  backend = "cmdstanr",
  chains = 2, iter = 1000, warmup = 500, refresh = 0
)

print(fit)

## End(Not run)
}

```

plot.subgroup_summary

Plot Marginal Subgroup Treatment Effects

Description

Creates a forest plot from a `subgroup_summary` object or list of objects.

Usage

```

## S3 method for class 'subgroup_summary'
plot(x, x_lab = NULL, title = NULL, ...)

```

Arguments

<code>x</code>	subgroup_summary or list. Object created by <code>summary_subgroup_effects()</code> , or a named list of such objects for comparison plots.
<code>x_lab</code>	character(1) or NULL. Custom label for x-axis.
<code>title</code>	character(1) or NULL. Custom title for plot.
<code>...</code>	Additional arguments (currently unused).

Details

This function creates forest plots for subgroup treatment effects. It supports two modes:

Single Model Plot: When `x` is a single `subgroup_summary` object, creates a traditional forest plot with estimates and confidence intervals displayed as text.

Multiple Model Comparison: When `x` is a named list of `subgroup_summary` objects, creates a comparative plot where different models are distinguished by colors and shapes. This is useful for comparing one-way models vs global models, different prior specifications, or sensitivity analyses.

Value

`ggplot`. Forest plot visualization of subgroup treatment effects.

Examples

```
## Not run:
# Single model plot
summary1 <- summary_subgroup_effects(brms_fit = model1)
plot(summary1, title = "Subgroup Effects")

# Multiple model comparison
summary2 <- summary_subgroup_effects(brms_fit = model2)
comparison <- list(
  "Model 1" = summary1,
  "Model 2" = summary2
)
plot(comparison, title = "Model Comparison")

## End(Not run)
```

```
prepare_formula_model
```

Prepare a Multi-Part brms Formula and Corresponding Data

Description

This function serves as a pre-processor for building complex Bayesian models with the `brms` package. It automates the construction of a multi-part, non-linear formula by classifying covariates into three distinct categories: unshrunk terms, shrunk prognostic, and shrunk predictive.

Usage

```
prepare_formula_model(
  data,
  response_formula,
  unshrunk_terms_formula = NULL,
```

```

shrunk_prognostic_formula = NULL,
shrunk_predictive_formula = NULL,
response_type = c("binary", "count", "continuous", "survival"),
stratification_formula = NULL
)

```

Arguments

- data** A data frame. Dataset containing all necessary variables for model fitting. Must include the response variable, treatment variable, and all covariates specified in the formula arguments. This data will be modified (contrast coding applied) and returned for use in model fitting.
- response_formula** A formula object. The response specification defining the outcome variable and treatment. Examples: `outcome ~ trt` for continuous outcomes, `n_events ~ trt + offset(log(days))` for count outcomes, or `Surv(time, status) ~ trt` for survival models. The treatment variable (right-hand side) will be automatically extracted and converted to a numeric binary variable (0/1).
- unshrunk_terms_formula** A formula object or NULL. Formula specifying unshrunk terms for the `unshrunktermeffect` component. May include main effects and treatment interactions without regularization. Supports three syntaxes that can be combined: colon notation (e.g., `~ age + sex + trt:biomarker`), star notation (e.g., `~ age + trt*biomarker`), or random effects notation (e.g., `~ age + (0+trt || biomarker)`).
- shrunk_prognostic_formula** A formula object or NULL. Formula specifying prognostic main effects to be regularized in the `shprogeffect` component. These are covariates where shrinkage/regularization is desired. Recommended: use `~ 0 + ...` or `~ -1 + ...` syntax to explicitly remove the intercept and apply one-hot encoding for symmetric regularization across all levels without privileging a reference group.
- shrunk_predictive_formula** A formula object or NULL. Formula specifying predictive terms (treatment interactions) to be regularized in the `shpredeffect` component. Supports three syntaxes that can be combined: colon notation (e.g., `~ 0 + trt:region`), star notation (e.g., `~ 0 + trt*region`), or random effects notation (e.g., `~ (0 + trt || region)`). Recommended: use `~ 0 + ...` or `~ -1 + ...` syntax to explicitly remove the intercept and apply one-hot encoding.
- response_type** A character string. Type of outcome variable, one of "binary", "count", "continuous", or "survival". This determines the appropriate likelihood function and link function for the model.
- stratification_formula** A formula object or NULL. Formula specifying stratification variable (e.g., `~ strata_var`) for modeling baseline hazard (survival models) or distributional parameters (other models). For survival models, estimates separate baseline hazard functions (`bhaz`) for each stratum. For continuous

models, models varying residual standard deviation (**sigma**). For count models, models varying overdispersion (**shape**).

Details

This classification allows for applying different priors (non-informative or shrinkage) to different parts of the model. The function also prepares the corresponding data using R's contrast coding system for proper factor handling and interactions.

Value

list with six named elements:

formula **brmsformula** object containing the complete multi-part model specification

data **data.frame** with modified contrast coding (must be used for model fitting)

response_type **character**(1) indicating the outcome type ("**binary**", "**count**", "**continuous**", or "**survival**")

trt_var **character**(1) name of treatment variable extracted from response formula

stan_variable_names **list** of character vectors showing Stan parameter names for each model component

has_intercept **logical**(1) indicating whether the unshrunk formula includes an intercept (**TRUE**) or was specified with `~ 0 + ...` (**FALSE**)

Key Features

- **Multi-Part Formula Construction:** It generates a **brmsformula** object with up to three distinct linear components (**unshrunktermeffect**, **shprogeffect**, **shpredeffect**), which are combined in a non-linear model. This allows for assigning different priors to each component.
- **Automated Interaction Handling:** Predictive terms support three syntaxes that can be used together or separately: colon notation (e.g., `~ trt:subgroup`), star notation (e.g., `~ trt*subgroup`), or random effects notation (e.g., `~ (0 + trt || subgroup)`). You can mix these syntaxes in the same model (e.g., `~ trt:var1 + trt*var2 + (0 + trt || var3)`). All variables involved in interactions are converted to factors with appropriate contrasts (reference coding for unshrunk, one-hot for shrunk), enabling proper model fitting and prediction on new data without manual dummy variable creation.
- **Hierarchical Integrity Check:** When a predictive term like `trt:subgroup` is specified, the function checks whether the corresponding prognostic effect **subgroup** is included in the model. If missing, it issues a warning message to alert you about the violation of the marginality principle, allowing you to decide whether to add the main effect. Note: Star notation (*) automatically includes main effects, so these variables are excluded from the marginality check.
- **Intelligent Defaults:** The main treatment variable is automatically added as an unshrunk prognostic term if not specified elsewhere. It also provides warnings and resolves overlaps if a term is accidentally specified in multiple categories.

Survival Model Details (Robust Spline Knot Calculation)

For survival models (`response_type = "survival"`), the function explicitly models the baseline hazard using B-splines via `brms::bhaz()`. It implements a highly robust method for calculating spline knots to ensure stability:

1. **Boundary Definition:** It first establishes boundary knots by taking the range of unique event times and adding a small buffer. This creates a "safe" interval for placing internal knots.
2. **Internal Knot Placement:** It calculates internal knots using quantiles of the event times that fall strictly within the boundary knots. This prevents knots from being placed at the exact edges of the data, which can cause numerical instability.
3. **Fallback for Sparse Data:** If there are too few unique event times to calculate quantile-based knots, it gracefully falls back to placing evenly spaced knots within the defined boundaries.

Data Transformation and Contrast Coding

This function returns a modified `data.frame` that must be used in subsequent calls to `fit_brms_model()`, not the original data. The transformations are:

- **Treatment variable:** Converted to numeric binary (0/1) to avoid multi-level factor interactions. The first level (alphabetically or numerically) becomes 0, the second becomes 1.
- **Factor covariates:** Automatic contrast coding based on shrinkage type:
 - **Shrunk terms:** One-hot encoding (all factor levels represented). This removes the reference group and treats all subgroups symmetrically without privileging a specific reference level (ensures the exchangeability assumption). Recommended: explicitly remove the intercept using `~ 0 + var` or `~ -1 + var` syntax. If not specified, the function will still apply one-hot encoding but may include a redundant intercept that could complicate interpretation.
 - **Unshrunk terms:** Dummy encoding (reference level dropped). Recommended: Use standard formula syntax `~ var` with intercept for clarity and proper reference coding.
- **Custom contrasts:** If you have pre-specified contrasts via `contrasts(data$var) <- <matrix>` before calling this function, they will be preserved. Otherwise, defaults are applied based on shrinkage category.

Stratification

The `stratification_formula_str` argument allows for estimating certain parameters separately for different groups. Its behavior depends on the `response_type`:

- **survival:** Estimates a separate baseline hazard function (`bhaz`) for each level of the stratification variable.
- **continuous:** Models the residual standard deviation (`sigma`) as varying across levels of the stratification variable.
- **count:** Models the overdispersion parameter (`shape`) as varying across levels of the stratification variable.

Examples

```

if (require("brms") && require("survival")) {
  # 1. Create Sample Data
  set.seed(123)
  n <- 100
  sim_data <- data.frame(
    time = round(runif(n, 1, 100)),
    status = sample(0:1, n, replace = TRUE),
    trt = sample(0:1, n, replace = TRUE),
    age = rnorm(n, 50, 10),
    region = sample(c("A", "B"), n, replace = TRUE),
    subgroup = sample(c("S1", "S2", "S3"), n, replace = TRUE)
  )
  sim_data$trt <- factor(sim_data$trt, levels = c(0, 1))
  sim_data$region <- as.factor(sim_data$region)
  sim_data$subgroup <- as.factor(sim_data$subgroup)

  # 2a. Example with colon interaction syntax (recommended: use ~ 0 + for shrunk terms)
  prepared_model <- prepare_formula_model(
    data = sim_data,
    response_formula = Surv(time, status) ~ trt,
    unshrunk_terms_formula = ~ age + subgroup,
    shrunk_predictive_formula = ~ 0 + trt:subgroup,
    response_type = "survival",
    stratification_formula = ~ region
  )

  # 2b. Alternatively, using pipe-pipe (||) syntax
  # prepared_model <- prepare_formula_model(
  #   data = sim_data,
  #   response_formula = Surv(time, status) ~ trt,
  #   unshrunk_terms_formula = ~ age,
  #   shrunk_predictive_formula = ~ (0 + trt || subgroup),
  #   response_type = "survival",
  #   stratification_formula = ~ region
  # )

  # 3. View the results
  print(prepared_model$formula)
  print(head(prepared_model$data))
}

```

run_brms_analysis

Run a Full Bayesian Hierarchical Model Analysis

Description

This high-level wrapper function streamlines the entire process of preparing and fitting a complex Bayesian hierarchical model with **brms**. It serves as a convenient single entry point for the complete workflow.

Usage

```
run_brms_analysis(
  data,
  response_formula,
  response_type = c("binary", "count", "continuous", "survival"),
  unshrunk_terms_formula = NULL,
  shrunk_prognostic_formula = NULL,
  shrunk_predictive_formula = NULL,
  stratification_formula = NULL,
  intercept_prior = NULL,
  unshrunk_prior = NULL,
  shrunk_prognostic_prior = NULL,
  shrunk_predictive_prior = NULL,
  stanvars = NULL,
  ...
)
```

Arguments

- | | |
|----------------------------------|---|
| data | A data frame. Dataset containing all necessary variables including response, treatment, and covariates. Passed to <code>prepare_formula_model()</code> for preprocessing. |
| response_formula | A formula object. Response specification defining the outcome and treatment. Examples: <code>outcome ~ trt</code> for continuous, or <code>Surv(time, status) ~ trt</code> for survival. Passed to <code>prepare_formula_model()</code> . |
| response_type | A character string. Outcome type, one of "binary", "count", "continuous", or "survival". Determines the likelihood function and link used. Passed to <code>prepare_formula_model()</code> . |
| unshrunk_terms_formula | A formula object or NULL. Unshrunk terms specification for the <code>unshrunktermeffect</code> component. May include main effects and treatment interactions without regularization (e.g., <code>~ age + sex + trt*region</code>). Passed to <code>prepare_formula_model()</code> . |
| shrunk_prognostic_formula | A formula object or NULL. Prognostic effects for strong regularization in the <code>shprogeffect</code> component. <ul style="list-style-type: none"> • Fixed effects (colon syntax): Use <code>~ 0 + ...</code> syntax for one-hot encoding (e.g., <code>~ 0 + biomarker1 + biomarker2</code>). • Random effects (one-way model, pipe-pipe syntax): Use random effects notation (e.g., <code>~ (0 + 1 biomarker)</code>) with automatic <code>normal(0, 1)</code> priors on SD scale. Passed to <code>prepare_formula_model()</code>. |
| shrunk_predictive_formula | A formula object or NULL. Treatment interactions for strong regularization in the <code>shpredeffect</code> component. <ul style="list-style-type: none"> • Fixed effects (colon syntax): Use <code>~ 0 + ...</code> syntax for one-hot encoding (e.g., <code>~ 0 + trt:subgroup</code>). |

- **Random effects (one-way model, pipe-pipe syntax):** Use random effects notation (e.g., `~ (0 + trt || subgroup)`) with automatic `normal(0, 1)` priors on SD scale. Passed to `prepare_formula_model()`.
- stratification_formula**
A formula object or NULL. Stratification variable specification (e.g., `~ strata_var`). For survival models, estimates separate baseline hazards per stratum. For other models, models varying distributional parameters. Passed to `prepare_formula_model()`.
- intercept_prior**
A character string, `brmsprior` object, or NULL. Intercept prior for `unshrunktermeffect` component. Not used for survival models (Cox models have no intercept). Example: `"normal(0, 10)"`. Passed to `fit_brms_model()`.
- unshrunk_prior**
A character string, `brmsprior` object, or NULL. Prior for unshrunk terms (non-intercept coefficients in `unshrunktermeffect` component). Example: `"normal(0, 2.5)"`. Passed to `fit_brms_model()`.
- shrunk_prognostic_prior**
A character string, `brmsprior` object, or NULL. Prior for regularized prognostic effects in `shprogeffect` component.
- **Fixed effects:** Typically strong regularization like `"horseshoe(scale_global = 1)"`.
 - **Random effects (one-way model):** Automatically uses `normal(0, 1)` priors on SD scale. Passed to `fit_brms_model()`.
- shrunk_predictive_prior**
A character string, `brmsprior` object, or NULL. Prior for regularized predictive effects (treatment interactions) in `shpredeffect` component.
- **Fixed effects:** Typically strong regularization like `"horseshoe(scale_global = 0.5)"`.
 - **Random effects (one-way model):** Automatically uses `normal(0, 1)` priors on SD scale. Passed to `fit_brms_model()`.
- stanvars**
A `stanvars` object or NULL. Custom Stan code created via `brms::stanvar()` for implementing hierarchical priors or other advanced Stan functionality. Passed to `fit_brms_model()`.
- ...**
Additional arguments passed to `brms::brm()` via `fit_brms_model()`. Common arguments include `chains`, `iter`, `cores`, `backend`, and `refresh`.

Details

This function is the main user-facing entry point. It first calls `prepare_formula_model()` to build the `brmsformula` and process the data, then passes the results to `fit_brms_model()` to run the analysis.

Value

brmsfit. Fitted Bayesian model object with attached attributes: `response_type`, `model_data` (the processed data used for fitting), and `trt_var` (treatment variable name) for use by downstream analysis functions.

Examples

```

if (require("brms") && require("survival")) {
  # 1. Create Sample Data
  set.seed(123)
  n <- 100
  sim_data <- data.frame(
    time = round(runif(n, 1, 100)),
    status = sample(0:1, n, replace = TRUE),
    trt = sample(0:1, n, replace = TRUE),
    age = rnorm(n, 50, 10),
    region = sample(c("A", "B"), n, replace = TRUE),
    subgroup = sample(c("S1", "S2", "S3"), n, replace = TRUE)
  )
  sim_data$trt <- factor(sim_data$trt, levels = c(0, 1))
  sim_data$region <- as.factor(sim_data$region)
  sim_data$subgroup <- as.factor(sim_data$subgroup)

  # 2. Run the full analysis using fixed effects (colon syntax)
  ## Not run:
  full_fit_fixed <- run_brms_analysis(
    data = sim_data,
    response_formula = Surv(time, status) ~ trt,
    response_type = "survival",
    unshrunk_terms_formula = ~ age,
    shrunk_prognostic_formula = ~ 0 + region,
    shrunk_predictive_formula = ~ 0 + trt:subgroup,
    stratification_formula = ~ region,
    unshrunk_prior = "normal(0, 2)",
    shrunk_prognostic_prior = "horseshoe(scale_global = 1)",
    shrunk_predictive_prior = "horseshoe(scale_global = 1)",
    backend = "cmdstanr",
    chains = 2, iter = 1000, warmup = 500, refresh = 0
  )

  print(full_fit_fixed)

  # 3. Alternative: One-way model with random effects (pipe-pipe syntax)
  # Random effects automatically get normal(0, 1) priors on SD scale
  full_fit_oneway <- run_brms_analysis(
    data = sim_data,
    response_formula = Surv(time, status) ~ trt,
    response_type = "survival",
    unshrunk_terms_formula = ~ age,
    shrunk_prognostic_formula = ~ (0 + 1 || region),
    shrunk_predictive_formula = ~ (0 + trt || subgroup),
    stratification_formula = ~ region,
    backend = "cmdstanr",
    chains = 2, iter = 1000, warmup = 500, refresh = 0
  )

  print(full_fit_oneway)

```

```
## End(Not run)
}
```

shrink_data

Simulated data for shrinkage analysis

Description

A simulated dataset with 500 observations designed for demonstrating subgroup shrinkage methods across multiple outcome types.

Usage

```
shrink_data
```

Format

A data frame with 500 rows and 11 columns:

id Subject identifier (1-500)
trt Binary treatment indicator (0 or 1)
x1 Categorical covariate with 2 levels (a, b)
x2 Categorical covariate with 3 levels (a, b, c)
x3 Categorical covariate with 4 levels (a, b, c, d)
y Continuous outcome
response Binary response (0 or 1)
tt_event Time-to-event outcome (months)
event_yn Event indicator (0 = censored, 1 = event)
fup_duration Follow-up duration in months
count Poisson count outcome

Generation process

The dataset was generated with `set.seed(16)` using the following steps:

Covariates

Treatment (**trt**) is assigned using permuted blocks of size 4, yielding a balanced binary indicator (0/1) across the 500 subjects. Three latent standard-normal variables (**z1**, **z2**, **z3**) are drawn independently, then **z2** is replaced by $0.4 * z1 + \sqrt{1 - 0.4^2} * z2$ to induce a correlation of 0.4 between **z1** and **z2**. The three subgroup covariates are obtained by discretising these latent variables:

- **x1**: cut of **z1** at 0.5 $\rightarrow$ binary levels **a** (69\ and **b** (31\
- **x2**: cut of **z2** at -0.5 and $0.5 \rightarrow$ three levels **a**, **b**, **c**. Because **z2** is correlated with **z1**, **x2** is also moderately correlated with **x1**.

- **x3**: cut of **z3** at -0.66 , 0 , and $0.66 \rightarrow$ four approximately equal-sized levels **a**, **b**, **c**, **d**.

Continuous outcome (**y**) and binary response (**response**)

$$y = 5 + 0.5 \cdot \mathbf{1}[x_1 = a] + 0.5 \cdot \mathbf{1}[x_3 = b] + 0.1 \cdot \mathbf{1}[\text{trt} = 1] + 0.4 \cdot \mathbf{1}[x_1 = a, \text{trt} = 1] + \varepsilon, \quad \varepsilon \sim N(0, 1)$$

This gives a population-level treatment effect of 0.1 , which is amplified to 0.5 within the **x1 = a** subgroup (treatment-by-**x1** interaction of 0.4). **x3 = b** acts as a prognostic covariate only. The binary outcome **response** is the indicator $y > 6$.

Time-to-event outcome (**tt_event**, **event_yn**)

Event times are drawn from an exponential distribution with rate

$$\lambda = \exp(-3.5 + \log(0.5) \cdot \mathbf{1}[x_1 = a] + \log(1.5) \cdot \mathbf{1}[x_3 = b] + \log(0.9) \cdot \mathbf{1}[\text{trt} = 1] + \log(0.5) \cdot \mathbf{1}[x_1 = a, \text{trt} = 1])$$

so the hazard ratios for treatment are 0.9 in the **x1 = b** subgroup and $0.9 \times 0.5 = 0.45$ in the **x1 = a** subgroup. Censoring combines administrative censoring (uniform accrual over 24 months, 60-month study duration) and loss to follow-up (exponential with an annual rate of 3%)

Poisson count outcome (**count**, **fup_duration**)

Individual follow-up is capped at 24 months or loss to follow-up, whichever comes first. Counts are drawn from a Poisson distribution with mean equal to the individual rate times follow-up duration, where the rate is

$$\mu = \exp(-3 + \log(0.8) \cdot \mathbf{1}[x_1 = a] + \log(0.6) \cdot \mathbf{1}[\text{trt} = 1])$$

There is no treatment-by-subgroup interaction in this outcome; the rate ratio for treatment is 0.6 across all subgroups.

The complete generation code is in `data-raw/shrink_data.R`.

summary_subgroup_effects

Create a Summary of Marginal Subgroup Treatment Effects

Description

This function orchestrates calls to `estimate_subgroup_effects` to generate a summary table for specific subgroups.

Usage

```
summary_subgroup_effects(
  brms_fit,
  trt_var = NULL,
  response_type = NULL,
  subgroup_vars = "auto",
  conf = 0.95
)
```

Arguments

brms_fit	A <code>brmsfit</code> object. Fitted model object from <code>fit_brms_model()</code> or <code>run_brms_analysis()</code> . Must contain the necessary attributes for extracting treatment variable and response type information.
trt_var	A character string or <code>NULL</code> . Treatment variable name (coded 0/1). If <code>NULL</code> , automatically extracted from model attributes (set by <code>fit_brms_model()</code>). This should match the treatment variable used in model fitting.
response_type	A character string or <code>NULL</code> . Outcome type, one of "binary", "count", "continuous", or "survival". If <code>NULL</code> , automatically extracted from model attributes (set by <code>fit_brms_model()</code>). This determines the appropriate scale for summarizing effects.
subgroup_vars	A character vector or "auto". Subgrouping variable names for which to generate summaries. Defaults to "auto", which automatically detects treatment interactions from all formula components (<code>unshrunktermeffect</code> , <code>shprogeffect</code> , <code>shpredeffect</code>). Cannot be <code>NULL</code> .
conf	A numeric scalar. Credible interval confidence level. Default is 0.95 (corresponding to 95% credible intervals). Example: use 0.90 for 90% credible intervals.

Value

`subgroup_summary`. S3 object containing:

- `estimates` tibble where each row represents a subgroup, with columns for `Subgroup`, `Median`, `CI_Lower`, and `CI_Upper` from posterior distribution
- `response_type` `character(1)` indicating outcome type
- `ci_level` `numeric(1)` credible interval level (default 0.95)
- `trt_var` `character(1)` treatment variable name

Index

* **datasets**

 shrink_data, [16](#)

combine_summaries, [2](#)

estimate_subgroup_effects, [3](#)

fit_brms_model, [4](#)

plot.subgroup_summary, [7](#)

prepare_formula_model, [8](#)

run_brms_analysis, [12](#)

shrink_data, [16](#)

summary_subgroup_effects, [17](#)