Package ‘shrinkr’

July 6, 2026

Type Package

Title Modular Bayesian Hierarchical Shrinkage Models

Version 0.4.5

Date 2026-06-29

Description Implements a two-stage Bayesian hierarchical modeling framework for applying shrinkage to subgroup-specific effects. The package separates model fitting (Stage 1) from hierarchical shrinkage (Stage 2), enabling modular sensitivity analyses without refitting expensive Markov chain Monte Carlo (MCMC) chains. Supports flexible prior specifications through the 'distributional' package, mixture approximations via 'mclust', and efficient 'Stan'-based inference.

License GPL (>= 3)

Encoding UTF-8

Depends R (>= 4.1)

Imports rstan (>= 2.32.7), mclust, posterior, distributional, cli, tibble, dplyr, purrr, stats, utils, methods, Rcpp (>= 0.12.0), RcppParallel (>= 5.0.1), rstantools (>= 2.5.0)

Suggests testthat (>= 3.0.0), ggplot2, knitr, rmarkdown, patchwork, tidybayes, bayesplot, tidyverse, ggdist, ggridges, brms, tidyr, survival, MASS, Matrix, beastt

LinkingTo BH (>= 1.66.0), Rcpp (>= 0.12.0), RcppEigen (>= 0.3.3.3.0), RcppParallel (>= 5.0.1), rstan (>= 2.32.7), StanHeaders (>= 2.32.10)

SystemRequirements GNU make, C++17 compiler

Biarch true

URL `https://gsk-biostatistics.github.io/shrinkr/`

BugReports `https://github.com/GSK-Biostatistics/shrinkr/issues`

VignetteBuilder knitr

NeedsCompilation yes

Config/roxygen2/version 8.0.0

Author Jacob M. Maronge [aut, cre] (ORCID:
<https://orcid.org/0000-0003-3606-0841>),
GlaxoSmithKline Research & Development Limited [cph, fnd],
Trustees of Columbia University [cph] (R/stanmodels.R, configure,
configure.win)

Maintainer Jacob M. Maronge <`jacob.m.maronge@gsk.com`>

Repository CRAN

Date/Publication 2026-07-06 13:40:02 UTC

Contents

`shrinkr-package` *shrinkr: Modular Bayesian Hierarchical Shrinkage Models*

Description

The `shrinkr` package provides a flexible framework for two-stage Bayesian hierarchical modeling. It enables post-hoc shrinkage of subgroup-specific posterior estimates from any Bayesian model, with support for diverse prior specifications and diagnostic tools.

Key Features

Two-Stage Workflow:

- Stage 1: Fit any Bayesian model without shrinkage
- Stage 2: Apply hierarchical shrinkage with flexible priors

Flexible Priors:

- Standard families (Normal, Student-t, Cauchy, Lognormal)
- Heavy-tailed (Inverse-Gamma, Half-Cauchy, Half-t)
- Bounded (Uniform)
- Mixture priors (spike-and-slab)
- Truncated distributions

Input Methods:

- Full posterior samples (via mixture approximation)
- Point estimates + variance/covariance

Main Functions

Core Workflow:

- `fit_mixture()`: Approximate Stage 1 posteriors with Gaussian mixture
- `shrink()`: Main user interface for hierarchical shrinkage

Prior Specification:

- `prior_spike_slab()`: Create spike-and-slab mixture prior
- `prior_mixture()`: Create custom mixture prior
- `sample_prior_predictive()`: Generate prior predictive samples for checking

Extraction & Visualization:

- `extract_mu_tau()`: Extract hyperparameter draws
- `extract_theta()`: Extract group-level draws
- `summarise_mu_tau()`: Summarize hyperparameters
- `summarise_theta()`: Summarize group-level estimates
- `theta_contrasts()`: Compute linear combinations of theta
- `plot()`: Visualize shrinkage effect and mixture approximation quality

Getting Started

See `vignette("getting_started", package = "shrinkr")` for a basic workflow, or `vignette("brms_integration", package = "shrinkr")` for a survival analysis example.

Use Cases

- **Meta-analysis:** Shrink study-specific effects
- **Clinical trials:** Borrow information across subgroups or historical controls
- **Genomics:** Regularize gene-specific effects
- **Simulation studies:** Compare shrinkage methods systematically

Package Options

- `shrinkr.refresh`: Controls Stan sampling progress output (default: 100)

Author(s)

Maintainer: Jacob M. Maronge `<jacob.m.maronge@gsk.com>` (ORCID)

Authors:

- Jacob M. Maronge `<jacob.m.maronge@gsk.com>` (ORCID)

Other contributors:

- GlaxoSmithKline Research & Development Limited [copyright holder, funder]
- Trustees of Columbia University (R/stanmodels.R, configure, configure.win) [copyright holder]

References

Maronge, J. M. (2026). shrinkr: Modular Bayesian Hierarchical Shrinkage Models. R package version 0.4.3.

See Also

- Stan: `https://mc-stan.org/`
- distributional package: `https://pkg.mitchelloharawild.com/distributional/`

Examples

```
## Not run:
# This example fits a Stan model, so it is not run during package checks.
library(shrinkr)
priors <- list(
  mu = distributional::dist_normal(0, 5),
  tau = distributional::dist_truncated(distributional::dist_student_t(3, 0, 1), lower = 0)
)
fit <- shrink(
  mle = c(0.0, 0.5, 1.0),
  var_matrix = c(0.25, 0.25, 0.25),
```

```
  hierarchical_priors = priors,
  iter = 1000, chains = 2, seed = 1
)
summary(fit)

## End(Not run)
```

`as.data.frame.shrinkr_fit` *Convert shrinkr_fit to data.frame*

Description

Extracts posterior draws as a regular data frame. This is a convenience wrapper around `as_draws_df()` that returns a plain data.frame.

Usage

```
## S3 method for class 'shrinkr_fit'
as.data.frame(
  x,
  row.names = NULL,
  optional = FALSE,
  variables = NULL,
  include_internals = FALSE,
  ...
)
```

Arguments

`x`	A `shrinkr_fit` object from `shrink()`.
`row.names`	NULL or character vector giving row names.
`optional`	Logical; if `TRUE`, setting row names and converting column names is optional.
`variables`	Character vector of parameter names to extract. If `NULL`, returns all user-facing parameters (excludes internals).
`include_internals`	Logical; if `TRUE`, includes internal Stan parameters. Default `FALSE`.
`...`	Additional arguments passed to `as_draws_df()`.

Value

A data.frame with columns for chain, iteration, draw, and requested parameters.

See Also

`as_draws_df.shrinkr_fit()` for posterior package format, `extract_mu_tau()` for hyperparameters only

Examples

```
set.seed(1)
draws <- data.frame(
  mu = rnorm(20, 0.2, 0.05),
  tau = abs(rnorm(20, 0.3, 0.03)),
  `theta[1]` = rnorm(20, 0.0, 0.1),
  `theta[2]` = rnorm(20, 0.3, 0.1),
  `theta[3]` = rnorm(20, 0.5, 0.1),
  check.names = FALSE
)
draws$tau_squared <- draws$tau^2
fit <- list(
  fit = posterior::as_draws_df(draws),
  data = list(
    G = 3, K = 1, centered = FALSE,
    vars = c("group1", "group2", "group3"),
    quantiles = data.frame(
      q2.5 = c(-0.20, 0.10, 0.30),
      q50 = c(0.00, 0.30, 0.50),
      q97.5 = c(0.20, 0.50, 0.70)
    )
  ),
  summary = posterior::summarise_draws(
    posterior::as_draws_df(draws),
    "mean", "sd",
    ~posterior::quantile2(., probs = c(0.025, 0.5, 0.975))
  ),
  diagnostics = list(n_divergent = 0, max_treedepth = 0, n_leapfrog = 0)
)
class(fit) <- "shrinkr_fit"
draws_df <- as.data.frame(fit)
head(draws_df)
mu_tau_df <- as.data.frame(fit, variables = c("mu", "tau"))
```

`as.data.frame.shrinkr_mixture`

Convert mixture fit to data frame

Description

Converts a fitted mixture model to a tidy long-format data frame. This is essentially an accessor for the `components` element.

Usage

```
## S3 method for class 'shrinkr_mixture'
as.data.frame(x, row.names = NULL, optional = FALSE, ...)
```

Arguments

`x`	A `shrinkr_mixture` object from `fit_mixture`.
`row.names`	Ignored (for S3 consistency).
`optional`	Ignored (for S3 consistency).
`...`	Additional arguments (currently unused).

Value

A data frame (or tibble if available) containing the component specifications with columns:

`component`	Component number (1 to K)
`variable`	Variable name
`weight`	Component weight (mixing proportion)
`mean`	Component mean for this variable
`sd`	Component marginal standard deviation for this variable

See Also

`fit_mixture` for fitting mixture models

Examples

```
set.seed(1)
samples <- list(
  group1 = matrix(rnorm(100, 0.0, 0.5), ncol = 1),
  group2 = matrix(rnorm(100, 0.5, 0.5), ncol = 1)
)
mix <- fit_mixture(samples, K_max = 2, verbose = FALSE)
df <- as.data.frame(mix)
head(df)
```

`as.data.frame.shrinkr_prior_pred`

Convert prior predictive samples to data frame

Description

Converts prior predictive samples to a tidy long-format data frame suitable for analysis and visualization with tidyverse tools.

Usage

```
## S3 method for class 'shrinkr_prior_pred'
as.data.frame(x, row.names = NULL, optional = FALSE, ...)
```

Arguments

`x`	A `shrinkr_prior_pred` object from `sample_prior_predictive`.
`row.names`	Ignored (for S3 consistency).
`optional`	Ignored (for S3 consistency).
`...`	Additional arguments (currently unused).

Value

A data frame (or tibble if tibble package is available) with columns:

`.draw`	Draw number (1 to n_draws)
`group`	Group name
`theta`	Sampled group-level effect
`mu`	Sampled global mean for this draw
`tau`	Sampled heterogeneity parameter for this draw

See Also

`sample_prior_predictive` for generating prior predictive samples

Examples

```
priors <- list(
  mu = distributional::dist_normal(0, 5),
  tau = distributional::dist_truncated(distributional::dist_normal(0, 1), lower = 0)
)
prior_pred <- sample_prior_predictive(priors, n_groups = 3, n_draws = 50)
df <- as.data.frame(prior_pred)
head(df)
```

`as_draws_df.shrinkr_fit` *Convert shrinkr_fit to draws_df*

Description

Extracts posterior draws in tidy format using the `posterior` package. By default returns user-facing parameters (mu, tau, theta, etc.) and excludes internal parameterization details. Set `include_internals = TRUE` to access all parameters including theta_c and z.

Usage

```
## S3 method for class 'shrinkr_fit'
as_draws_df(x, variables = NULL, include_internals = FALSE, ...)
```

Arguments

`x`	A `shrinkr_fit` object from `shrink()`.
`variables`	Character vector of parameter names to extract. Options include: • `"mu"` - Global mean • `"tau"` - Heterogeneity SD • `"tau_squared"` - Heterogeneity variance • `"theta"` or `"theta[i]"` - Subgroup effects If `NULL` (default), returns all user-facing parameters.
`include_internals`	Logical; if `TRUE`, includes internal Stan parameters (`theta_c`, `z`) used for parameterization. Default `FALSE`. Only applies when `variables` = `NULL`.
`...`	Additional arguments passed to `posterior::as_draws_df()`.

Value

A `posterior::draws_df` with columns for chain, iteration, draw, and requested parameters.

See Also

`shrink()` for fitting models, `extract_mu_tau()` for hyperparameters only

Examples

```
set.seed(1)
draws <- data.frame(
  mu = rnorm(20, 0.2, 0.05),
  tau = abs(rnorm(20, 0.3, 0.03)),
  `theta[1]` = rnorm(20, 0.0, 0.1),
  `theta[2]` = rnorm(20, 0.3, 0.1),
  `theta[3]` = rnorm(20, 0.5, 0.1),
  check.names = FALSE
)
draws$tau_squared <- draws$tau^2
fit <- list(
  fit = posterior::as_draws_df(draws),
  data = list(
    G = 3, K = 1, centered = FALSE,
    vars = c("group1", "group2", "group3"),
    quantiles = data.frame(
      q2.5 = c(-0.20, 0.10, 0.30),
      q50 = c(0.00, 0.30, 0.50),
      q97.5 = c(0.20, 0.50, 0.70)
    )
  ),
  summary = posterior::summarise_draws(
    posterior::as_draws_df(draws),
    "mean", "sd",
    ~posterior::quantile2(., probs = c(0.025, 0.5, 0.975))
  ),
```

```
  diagnostics = list(n_divergent = 0, max_treedepth = 0, n_leapfrog = 0)
)
class(fit) <- "shrinkr_fit"
all_draws <- posterior::as_draws_df(fit)
posterior::variables(all_draws)
theta_draws <- posterior::as_draws_df(fit, variables = c("theta[1]", "theta[2]"))
```

`extract_mu_tau` *Extract mu and tau parameters*

Description

Extracts posterior draws for the hyperparameters mu (global mean) and tau (heterogeneity standard deviation) from a fitted shrinkage model.

Usage

```
extract_mu_tau(x, ...)
```

Arguments

`x`	A `shrinkr_fit` object from `shrink()`.
`...`	Additional arguments (currently unused).

Value

A `posterior::draws_df` with columns:

`.chain`	Chain index
`.iteration`	Iteration within chain
`.draw`	Overall draw index
`mu`	Global mean parameter
`tau`	Heterogeneity parameter
`tau_squared`	Variance (tau^2)

See Also

`shrink()` for fitting models, `summarise_mu_tau()` for summary statistics, `as_draws_df.shrinkr_fit()` for all parameters

Examples

```
set.seed(1)
draws <- data.frame(
  mu = rnorm(20, 0.2, 0.05),
  tau = abs(rnorm(20, 0.3, 0.03)),
  `theta[1]` = rnorm(20, 0.0, 0.1),
  `theta[2]` = rnorm(20, 0.3, 0.1),
  `theta[3]` = rnorm(20, 0.5, 0.1),
  check.names = FALSE
)
draws$tau_squared <- draws$tau^2
fit <- list(
  fit = posterior::as_draws_df(draws),
  data = list(
    G = 3, K = 1, centered = FALSE,
    vars = c("group1", "group2", "group3"),
    quantiles = data.frame(
      q2.5 = c(-0.20, 0.10, 0.30),
      q50 = c(0.00, 0.30, 0.50),
      q97.5 = c(0.20, 0.50, 0.70)
    )
  ),
  summary = posterior::summarise_draws(
    posterior::as_draws_df(draws),
    "mean", "sd",
    ~posterior::quantile2(., probs = c(0.025, 0.5, 0.975))
  ),
  diagnostics = list(n_divergent = 0, max_treedepth = 0, n_leapfrog = 0)
)
class(fit) <- "shrinkr_fit"
mu_tau <- extract_mu_tau(fit)
summarise_mu_tau(fit)
posterior::summarise_draws(mu_tau)
```

`extract_theta` *Extract theta (group-level effect) parameters*

Description

Extracts posterior draws for the group-level effects (theta parameters) from a fitted shrinkage model. This is the hierarchically shrunk version of the subgroup effects.

Usage

```
extract_theta(x, ...)
```

Arguments

`x`	A `shrinkr_fit` object from `shrink()`.
`...`	Additional arguments passed to `as_draws_df.shrinkr_fit()`.

Value

A `posterior::draws_df` with columns:

`.chain`	Chain index
`.iteration`	Iteration within chain
`.draw`	Overall draw index
`theta[1]`, `theta[2]`, ...	Group-level effects

See Also

`shrink()` for fitting models, `extract_mu_tau()` for hyperparameters, `summarise_theta()` for summary statistics, `theta_contrasts()` for pairwise comparisons

Examples

```
set.seed(1)
draws <- data.frame(
  mu = rnorm(20, 0.2, 0.05),
  tau = abs(rnorm(20, 0.3, 0.03)),
  `theta[1]` = rnorm(20, 0.0, 0.1),
  `theta[2]` = rnorm(20, 0.3, 0.1),
  `theta[3]` = rnorm(20, 0.5, 0.1),
  check.names = FALSE
)
draws$tau_squared <- draws$tau^2
fit <- list(
  fit = posterior::as_draws_df(draws),
  data = list(
    G = 3, K = 1, centered = FALSE,
    vars = c("group1", "group2", "group3"),
    quantiles = data.frame(
      q2.5 = c(-0.20, 0.10, 0.30),
      q50 = c(0.00, 0.30, 0.50),
      q97.5 = c(0.20, 0.50, 0.70)
    )
  ),
  summary = posterior::summarise_draws(
    posterior::as_draws_df(draws),
    "mean", "sd",
    ~posterior::quantile2(., probs = c(0.025, 0.5, 0.975))
  ),
  diagnostics = list(n_divergent = 0, max_treedepth = 0, n_leapfrog = 0)
)
class(fit) <- "shrinkr_fit"
theta_draws <- extract_theta(fit)
posterior::summarise_draws(theta_draws)
summarise_theta(fit)
```

`fit_mixture` *Fit Gaussian mixture models to posterior samples*

Description

Fits a multivariate Gaussian mixture model (GMM) jointly across all supplied variables using **mclust**. The function is intended for approximating posterior draws from Bayesian models and produces a tidy component table suitable for visualization and shrinkage modeling.

Usage

```
fit_mixture(samples, K_max = 5L, verbose = FALSE, model_names = NULL, ...)
```

Arguments

`samples`	Posterior samples in one of the following formats: • **Data frame or matrix** (recommended): Each column represents one group/variable to shrink, with rows as posterior draws. Column names are used as variable labels. Example: output from `posterior::as_draws_df()` or a matrix where columns = groups. • **Named list of vectors**: Each list element contains posterior samples for one group (as a numeric vector). Names are used as variable labels. All vectors must have the same length. • **Named list of matrices**: Each list element is a matrix of posterior samples for one group. For univariate parameters, these should be single-column matrices (n × 1). All matrices must have the same number of rows (draws). At least **two** groups/variables are required for hierarchical shrinkage. Non-numeric columns in data frames are automatically dropped. Rows with missing values are removed before fitting.
`K_max`	Integer (≥ 1). Maximum number of mixture components to consider during model selection by BIC. Internally capped at `n - 1` for stability.
`verbose`	Logical. If `TRUE`, progress and diagnostic messages are printed.
`model_names`	Optional character vector of **mclust** covariance model codes to consider (e.g., `"EII"`, `"VVV"`, etc.). If `NULL` (default), all models appropriate for the data dimension are considered by **mclust**. This is a convenience wrapper for `modelNames` in `mclust::Mclust()`.
`...`	Additional arguments forwarded to `mclust::Mclust()`, such as `prior`, `initialization`, `control = mclust::emControl()`, `warn`, or `verbose`. Use these to fine-tune EM control, priors, or initialization.

Details

- Requires at least **two** variables; shrinkage across a single variable is not meaningful.
- Rows with any missing values are removed before fitting.

- Component weights are normalized to sum to one.
- The `sd` column reports marginal standard deviations (square roots of diagonal entries) from each component covariance matrix.

Value

A list with class "shrinkr_mixture" containing:

- `components` — data frame with columns: `group`, `component`, `variable`, `weight`, `mean`, and `sd` (marginal standard deviations).
- `K` — number of mixture components selected.
- `vars` — character vector of variable names.
- `weights` — vector of component weights (mixing proportions).
- `covs` — list of component covariance matrices (p × p each).
- `model_name` — selected mclust covariance structure (e.g., "VVV").
- `bic` — Bayesian Information Criterion for the fitted model.
- `n_samples` — number of samples used in fitting.
- `n_vars` — number of variables.
- `mclust_fit` — the complete mclust model object (for advanced use).
- `diagnostics` — list with sample size details, removed rows, and quality warnings.

Covariance structures in mclust

mclust parameterizes component covariances via eigen-decomposition and offers a set of model families controlling volume (V), shape (S), and orientation (O). Common codes include (non-exhaustive):

Spherical `"EII"`: equal volume, spherical
`"VII"`: variable volume, spherical

Diagonal `"EEI"`: equal volume & shape (axis-aligned)
`"VEI"`: variable volume, equal shape
`"EVI"`: equal volume, variable shape
`"VVI"`: variable volume & shape

Ellipsoidal (full covariance) `"EEE"`: equal volume, shape, orientation
`"EEV"`: equal volume & shape, variable orientation
`"VEV"`: variable volume, equal shape, variable orientation
`"VVV"`: variable volume, shape, and orientation (most flexible)

If `model_names` = `NULL` (default), `mclust::Mclust()` selects among the models appropriate for the data dimension via BIC. In practice, this lets the data decide between parsimonious structures (e.g., `"EII"`, `"VVI"`) and fully flexible ones (e.g., `"VVV"`). You can restrict or expand the search space by supplying `model_names`.

See Also

`plot.shrinkr_mixture` for visualizing marginal fits, `as.data.frame.shrinkr_mixture` for extracting component data

Examples

```
set.seed(1)
samples <- list(
  group1 = matrix(rnorm(100, 0.0, 0.5), ncol = 1),
  group2 = matrix(rnorm(100, 0.5, 0.5), ncol = 1),
  group3 = matrix(rnorm(100, 1.0, 0.5), ncol = 1)
)
mix <- fit_mixture(samples, K_max = 2, verbose = FALSE)
summary(mix)
```

`plot.shrinkr_fit` *Plot shrinkage fit*

Description

Visualizes the hierarchical shrinkage model fit. Creates either:

- `"shrinkage"` — Shows pre/post-shrunk estimates with arrows
- `"diagnostics"` — Multi-panel view with hyperparameters and shrinkage factor

The shrinkage plot displays:

- **Pre-shrunk estimates** (hollow circles) — from stage 1 mixture or MLEs
- **Post-shrunk estimates** (filled circles) — posterior means of theta
- **Global mean** (dashed line) — posterior mean of mu
- **Credible intervals** (optional) — for shrunken estimates
- **Arrows** (optional) — showing direction and magnitude of shrinkage

Usage

```
## S3 method for class 'shrinkr_fit'
plot(
  x,
  type = c("shrinkage", "diagnostics"),
  group_names = NULL,
  show_arrows = FALSE,
  show_intervals = TRUE,
  interval_prob = 0.95,
  point_size = 3,
  arrow_alpha = 0.6,
  dodge_width = 0.3,
  title = NULL,
  subtitle = NULL,
  ...
)
```

Arguments

`x`	A `shrinkr_fit` object from `shrink()`.
`type`	Character; type of plot. Options: • `"shrinkage"` - Basic shrinkage visualization (default) • `"diagnostics"` - Multi-panel with hyperparameters and shrinkage factor
`group_names`	Optional character vector of length G to label groups. If `NULL`, uses names from `x$data$vars` or defaults to "group1", etc.
`show_arrows`	Logical; draw arrows from pre-shrunk to post-shrunk estimates? Default `FALSE`. Only applies when `type = "shrinkage"`.
`show_intervals`	Logical; show credible intervals for both pre-shrunk and post-shrunk estimates? Default `TRUE`. Only applies when `type = "shrinkage"`.
`interval_prob`	Numeric; probability mass for credible intervals. Default 0.95 for 95% intervals. Only applies when `type = "shrinkage"`.
`point_size`	Numeric; size of points. Default 3.
`arrow_alpha`	Numeric; transparency of arrows (0-1). Default 0.6. Only applies when `show_arrows = TRUE`.
`dodge_width`	Numeric; horizontal spacing between pre-shrunk and post-shrunk estimates in the side-by-side display. Default 0.3. Larger values increase separation between estimate types.
`title`	Character; plot title. If `NULL`, uses default title.
`subtitle`	Character; plot subtitle. If `NULL`, auto-generates from global mean and tau.
`...`	Additional arguments (currently unused).

Value

A ggplot2 object (for `type = "shrinkage"`), or a patchwork object/list (for `type = "diagnostics"`).

See Also

`shrink()` for fitting models, `extract_mu_tau()` for hyperparameter draws

Examples

```
# Plotting requires a fitted shrinkr_fit object from shrink().
# The full example is not run because it fits a Stan model.
## Not run:
fit <- shrink(mixture = mix, hierarchical_priors = priors)
plot(fit)
plot(fit, show_arrows = TRUE, interval_prob = 0.95)

## End(Not run)
```

`plot.shrinkr_mixture` *Plot fitted marginal densities or QQ plots for mixture models*

Description

Overlays fitted **marginal** mixture densities from a `shrinkr_mixture` (returned by `fit_mixture()`) on top of the **observed samples** for selected variables, OR creates QQ plots comparing empirical vs fitted quantiles. The function uses the *same coercion logic* as `fit_mixture()` (via an internal `.coerce_draws_df()` helper), ensuring that variable names line up even when users pass a list of matrices.

Important: For multivariate joint fits, this produces **marginal** overlays (one panel per variable when faceting). Each marginal density is computed by summing the weighted component densities for that variable.

Usage

```
## S3 method for class 'shrinkr_mixture'
plot(
  x,
  draws = NULL,
  variables = NULL,
  type = c("density", "qq"),
  overlay = c("hist", "kde", "both", "none"),
  bins = 50,
  kde_bw = NULL,
  show_components = TRUE,
  facet = TRUE,
  n_points = 501,
  verbose = FALSE,
  ...
)
```

Arguments

`x`	A `shrinkr_mixture` object from `fit_mixture()`.
`draws`	Optional samples to show as histogram/KDE or for QQ plot. Accepts any input shape supported by `fit_mixture()`. When `NULL`, only fitted curves are drawn (QQ plot requires draws).
`variables`	Character vector of variables to plot. Defaults to all variables in `x$components$variable`. Variable names must match the names created by the fitter (and by `.coerce_draws_df()`).
`type`	One of `c("density","qq")`. Default `"density"` shows density overlay; `"qq"` creates quantile-quantile plots comparing empirical vs fitted quantiles.
`overlay`	One of `c("hist","kde","both","none")`. Default `"hist"`. Only applies when `type = "density"`.
`bins`	Integer number of bins for the histogram (default `50`).

`kde_bw`	Bandwidth for `stats::density()`; `NULL` uses the default. Ignored unless `overlay` is `"kde"` or `"both"`.
`show_components`	Logical; if `TRUE` (default) overlays per-component curves using component weights, means, and **marginal** SDs from `x$components`. Only applies when `type = "density"`.
`facet`	Logical; if `TRUE` (default) facet by variable when plotting more than one variable.
`n_points`	Integer; number of x grid points for evaluating densities (default `501`). For QQ plots, this controls the number of quantiles to compare.
`verbose`	Logical; print brief matching diagnostics.
`...`	Additional arguments (currently unused).

Details

Density plots:

The total marginal density for each variable j is computed as

$$f_j(x) = \sum_{k=1}^{K} w_k\,\phi(x \mid \mu_{jk},\, \sigma_{jk})\,,$$

using per-component marginal SDs (`sd`) already stored in `x$components`.

The plotting range per variable is taken from the sample range if available (with 5% padding), otherwise from `mean +/- 4*sd` across that variable's components–avoiding non-finite `seq()` errors when samples are absent.

QQ plots:

When `type = "qq"`, the function creates quantile-quantile plots by:

1. Computing empirical quantiles from the observed data
2. Computing theoretical quantiles from the fitted mixture CDF via numerical inversion
3. Plotting empirical vs theoretical quantiles with a 45-degree reference line

Points falling on the reference line indicate good agreement between the fitted mixture and the data. Systematic deviations suggest model misfit.

Value

A `ggplot2` object.

See Also

`fit_mixture()` for fitting mixture models

Examples

```
set.seed(1)
samples <- list(
  group1 = matrix(rnorm(100, 0.0, 0.5), ncol = 1),
  group2 = matrix(rnorm(100, 0.5, 0.5), ncol = 1)
```

```
)
mix <- fit_mixture(samples, K_max = 2, verbose = FALSE)
plot(mix, draws = samples, type = "density", variables = c("group1", "group2"))
```

---

plot.shrinkr_prior_contrasts

Plot prior predictive pairwise differences

Description

Creates a density plot of $|\theta_i - \theta_j|$ from prior predictive samples. Useful for calibrating hierarchical priors. Styling matches `plot.shrinkr_prior_pred()` for visual consistency.

Usage

```
## S3 method for class 'shrinkr_prior_contrasts'
plot(x, by_pair = FALSE, ...)
```

Arguments

`x`	A `shrinkr_prior_contrasts` object from `prior_pairwise_differences`.
`by_pair`	Logical; if `TRUE`, facet by pair. If `FALSE` (default), pool all pairwise differences into a single plot.
`...`	Additional arguments (currently unused).

Value

A ggplot2 object.

See Also

`prior_pairwise_differences`

Examples

```
priors <- list(
  mu = distributional::dist_normal(0, 5),
  tau = distributional::dist_truncated(distributional::dist_normal(0, 1), lower = 0)
)
prior_pred <- sample_prior_predictive(priors, n_groups = 3, n_draws = 50)
pw <- prior_pairwise_differences(prior_pred)
plot(pw)
```

`plot.shrinkr_prior_pred` *Plot prior predictive samples*

Description

Visualizes the prior predictive distribution for hyperparameters (mu and tau) and subgroup effects (theta). Requires the ggplot2 package.

Usage

```
## S3 method for class 'shrinkr_prior_pred'
plot(x, type = c("both", "hyperparameters", "theta"), ...)
```

Arguments

`x`	A `shrinkr_prior_pred` object from `sample_prior_predictive`.
`type`	Character; type of plot. Options: • `"both"` - Both hyperparameter and theta plots (default) • `"hyperparameters"` - Density plots for mu and tau only • `"theta"` - Violin plots for theta by group only
`...`	Additional arguments (currently unused).

Value

A ggplot2 object, or a list of two ggplot2 objects if `type = "both"` and patchwork package is not available. If patchwork is available and `type = "both"`, returns a combined plot.

See Also

`sample_prior_predictive` for generating samples, `as.data.frame.shrinkr_prior_pred` for extracting data

Examples

```
priors <- list(
  mu = distributional::dist_normal(0, 5),
 tau = distributional::dist_truncated(distributional::dist_student_t(3, 0, 1), lower = 0)
)
prior_pred <- sample_prior_predictive(priors, n_groups = 3, n_draws = 50)
plot(prior_pred, type = "hyperparameters")
```

`print.shrinkr_fit` *Print method for shrinkr_fit*

Description

Displays a compact summary of the fitted model including dimensions, hyperparameter estimates, and diagnostics.

Usage

```
## S3 method for class 'shrinkr_fit'
print(x, digits = 3, ...)
```

Arguments

`x`	A `shrinkr_fit` object.
`digits`	Number of digits to display. Default is 3.
`...`	Additional arguments (currently unused).

Value

Invisibly returns the input object `x`.

See Also

`shrink()` for fitting models, `summarise_theta()` for detailed theta estimates

`print.shrinkr_mixture` *Print method for mixture fits*

Description

Displays summary information about a fitted mixture model in a readable format.

Usage

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

Arguments

`x`	A `shrinkr_mixture` object from `fit_mixture`.
`...`	Additional arguments (currently unused).

Value

Invisibly returns the input object `x`.

See Also

`fit_mixture` for fitting mixture models, `summary.shrinkr_mixture` for detailed summaries

Examples

```
set.seed(1)
samples <- list(
  group1 = matrix(rnorm(100, 0.0, 0.5), ncol = 1),
  group2 = matrix(rnorm(100, 0.5, 0.5), ncol = 1)
)
mix <- fit_mixture(samples, K_max = 2, verbose = FALSE)
print(mix)
```

`print.shrinkr_prior_contrasts`

Print method for prior pairwise contrasts

Description

Print method for prior pairwise contrasts

Usage

```
## S3 method for class 'shrinkr_prior_contrasts'
print(x, digits = 3, ...)
```

Arguments

`x`	A `shrinkr_prior_contrasts` object.
`digits`	Number of digits to display. Default 3.
`...`	Additional arguments (currently unused).

Value

Invisibly returns `x`.

`print.shrinkr_prior_pred` *Print method for prior predictive samples*

Description

Displays summary information about prior predictive samples in a readable format.

Usage

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

Arguments

`x`	A `shrinkr_prior_pred` object from `sample_prior_predictive`.
`...`	Additional arguments (currently unused).

Value

Invisibly returns the input object `x`.

See Also

`sample_prior_predictive` for generating samples, `summary.shrinkr_prior_pred` for detailed summaries

Examples

```
priors <- list(
  mu = distributional::dist_normal(0, 5),
  tau = distributional::dist_truncated(distributional::dist_normal(0, 1), lower = 0)
)
prior_pred <- sample_prior_predictive(priors, n_groups = 3, n_draws = 50)
print(prior_pred)
```

`print.summary.shrinkr_mixture` *Print summary of mixture fit*

Description

Displays formatted summary statistics for a fitted mixture model.

Usage

```
## S3 method for class 'summary.shrinkr_mixture'
print(x, digits = 3, ...)
```

Arguments

`x`	A summary object from `summary.shrinkr_mixture`.
`digits`	Number of decimal digits to display. Default is 3.
`...`	Additional arguments (currently unused).

Value

Invisibly returns the input object `x`.

See Also

`summary.shrinkr_mixture` for creating summaries

Examples

```
set.seed(1)
samples <- list(
  group1 = matrix(rnorm(100, 0.0, 0.5), ncol = 1),
  group2 = matrix(rnorm(100, 0.5, 0.5), ncol = 1)
)
mix <- fit_mixture(samples, K_max = 2, verbose = FALSE)
summ <- summary(mix)
print(summ)
```

`print.summary.shrinkr_prior_pred`

Print summary of prior predictive samples

Description

Displays formatted summary statistics for prior predictive samples.

Usage

```
## S3 method for class 'summary.shrinkr_prior_pred'
print(x, digits = 3, ...)
```

Arguments

`x`	A summary object from `summary.shrinkr_prior_pred`.
`digits`	Number of decimal digits to display. Default is 3.
`...`	Additional arguments (currently unused).

Value

Invisibly returns the input object `x`.

See Also

`summary.shrinkr_prior_pred` for creating summaries

Examples

```
priors <- list(
  mu = distributional::dist_normal(0, 5),
  tau = distributional::dist_truncated(distributional::dist_normal(0, 1), lower = 0)
)
prior_pred <- sample_prior_predictive(priors, n_groups = 3, n_draws = 50)
print(summary(prior_pred))
```

`prior_mixture` *Create a mixture prior*

Description

Creates a mixture distribution using `distributional::dist_mixture()`. All standard distributional operations (sampling, density, quantiles, formatting) work automatically.

Usage

```
prior_mixture(..., weights = NULL)
```

Arguments

`...`	Component distributions (from distributional package)
`weights`	Mixture weights (normalized automatically if not summing to 1)

Value

A `distributional` mixture distribution object

Examples

```
mix <- prior_mixture(
  distributional::dist_normal(0, 0.1),
  distributional::dist_normal(0, 1),
  weights = c(0.7, 0.3)
)
```

`prior_pairwise_differences` *Compute prior predictive pairwise differences |theta_i - theta_j|*

Description

Computes the prior-implied distribution of absolute pairwise differences between subgroup effects. This is useful for calibrating priors: if your prior implies that subgroup differences of 5 units are common but clinical relevance starts at 0.5, your prior may be too diffuse.

This implements the recommendation from the SDSIH Vignettes Library: inspect the prior distribution of $|\theta_i - \theta_j|$ when choosing priors for Bayesian hierarchical models.

Usage

```
prior_pairwise_differences(prior_pred)
```

Arguments

`prior_pred`	A `shrinkr_prior_pred` object from `sample_prior_predictive`.

Value

A list with class `"shrinkr_prior_contrasts"` containing:

`differences`	Data frame with columns `pair`, `abs_diff`, and `.draw`
`summary`	Data frame with per-pair summary statistics
`overall_summary`	Named numeric vector of quantiles across all pairs
`n_pairs`	Number of unique pairs
`n_draws`	Number of prior predictive draws
`group_names`	Group labels used

See Also

`sample_prior_predictive` for generating prior predictive samples, `plot.shrinkr_prior_contrasts` for visualizing the result

Examples

```
priors <- list(
  mu = distributional::dist_normal(0, 5),
  tau = distributional::dist_truncated(distributional::dist_normal(0, 1), lower = 0)
)
prior_pred <- sample_prior_predictive(priors, n_groups = 3, n_draws = 50)
pw <- prior_pairwise_differences(prior_pred)
print(pw)
pw$overall_summary
```

`prior_spike_slab` *Spike-and-slab prior for testing homogeneity*

Description

Creates a mixture of two Normal distributions. Since tau is a scale parameter, this must be wrapped in `distributional::dist_truncated()` with `lower = 0` before passing to `shrink()`:

Usage

```
prior_spike_slab(
  spike_location = 0,
  spike_scale = 0.01,
  slab_scale = 1,
  spike_prob = 0.5
)
```

Arguments

`spike_location`	Location of the spike (default 0)
`spike_scale`	Scale of the spike component (default 0.01)
`slab_scale`	Scale of the slab component (default 1)
`spike_prob`	Probability of the spike component (default 0.5)

Details

```
tau_prior <- dist_truncated(prior_spike_slab(), lower = 0)
```

Value

A spike-and-slab mixture distribution

Examples

```
tau_prior <- distributional::dist_truncated(
  prior_spike_slab(spike_prob = 0.5, spike_scale = 0.01, slab_scale = 1),
  lower = 0
)
```

`sample_prior_predictive`

Sample from prior predictive distribution

Description

Generates samples from the prior predictive distribution for the hierarchical shrinkage model. Useful for prior elicitation and sensitivity analysis.

The generative process is:

1. Sample mu from p(mu)
2. Sample tau from p(tau)
3. Sample theta_i ~ N(mu, tau) for each group i

Usage

```
sample_prior_predictive(
  hierarchical_priors,
  n_groups,
  n_draws = 1000,
  group_names = NULL
)
```

Arguments

`hierarchical_priors`	Named list with `mu` and `tau` distributional objects from the `distributional` package.
`n_groups`	Integer; number of subgroups (G).
`n_draws`	Integer; number of prior predictive samples to draw. Default 1000.
`group_names`	Optional character vector of length `n_groups` to label groups.

Value

A list with class "shrinkr_prior_pred" containing:

`mu`	Vector of mu draws
`tau`	Vector of tau draws
`theta`	Matrix of theta draws (n_draws x n_groups)
`implied_range`	Vector of ranges (max - min) of theta across groups for each draw
`implied_sd`	Vector of standard deviations of theta across groups for each draw
`group_names`	Group labels
`n_draws`	Number of draws
`n_groups`	Number of groups
`priors`	The hierarchical_priors specification used

See Also

`shrink` for fitting the hierarchical model, `plot.shrinkr_prior_pred` for visualizing prior predictive samples

Examples

```
priors <- list(
  mu = distributional::dist_normal(0, 5),
  tau = distributional::dist_truncated(distributional::dist_normal(0, 1), lower = 0)
)
prior_pred <- sample_prior_predictive(priors, n_groups = 3, n_draws = 50)
median(prior_pred$implied_range)
head(as.data.frame(prior_pred))
```

`shrink` *Bayesian Hierarchical Shrinkage Model*

Description

Applies hierarchical shrinkage to group-specific estimates using a two-stage Bayesian approach. Takes either a Gaussian mixture approximation of Stage 1 posteriors or point estimates with variance, and applies a Normal hierarchical model with flexible hyperpriors.

Usage

```
shrink(
  mixture = NULL,
  mle = NULL,
  var_matrix = NULL,
  hierarchical_priors = list(mu = distributional::dist_normal(0, 5), tau =
   distributional::dist_truncated(distributional::dist_normal(0, 2.5), lower = 0)),
  centered = FALSE,
  verbose = TRUE,
  ...
)
```

Arguments

`mixture`	A `shrinkr_mixture` object from `fit_mixture()`. Contains the Gaussian mixture approximation of Stage 1 posteriors. Either `mixture` or both `mle` and `var_matrix` must be provided.
`mle`	Numeric vector of group point estimates. Used when `mixture` is NULL.
`var_matrix`	Numeric vector of variances (length `G`) or covariance matrix (`G × G`). Required when `mle` is provided.
`hierarchical_priors`	Named list with `mu` and `tau` priors as `distributional` objects. Defaults to weakly informative priors:

- `mu`: Global mean, `dist_normal(0, 5)`
- `tau`: Between-group SD, `dist_truncated(dist_normal(0, 2.5), lower = 0)`

Supported distributions for `mu`: Normal, Student-t, mixture priors, and truncated versions of these (e.g., `dist_truncated(dist_normal(0, 5), lower = 0)`).

Supported distributions for `tau`: Normal (truncated), Student-t (truncated), Cauchy (truncated), Lognormal, Gamma, Inverse-Gamma, Exponential, Uniform, and mixture priors (including spike-and-slab via `prior_spike_slab()`).

`centered`	Logical; use centered (`TRUE`) or non-centered (`FALSE`, default) parameterization. Non-centered is more efficient when heterogeneity is small.
`verbose`	Logical; print progress messages (default `TRUE`).
`...`	Additional arguments passed to `rstan::sampling()`:

- `chains`: Number of chains (default 4)
- `iter`: Iterations per chain (default 2000)
- `warmup`: Warmup iterations (default iter/2)
- `cores`: Cores for parallel sampling
- `seed`: Random seed
- `control`: List of sampler controls (e.g., `list(adapt_delta = 0.95)`)

Details

Model Specification:

Hierarchical model (Stage 2):

$$\theta_g \mid \mu, \tau \sim \text{Normal}(\mu, \tau^2), \quad g = 1, \ldots, G$$

$$\mu \sim \pi(\mu)$$

$$\tau \sim \pi(\tau)$$

Stage 1 likelihood (Gaussian mixture approximation):

$$\theta_g \mid D_g \sim q_g(\theta_g) \approx \sum_{k=1}^{K} w_k \, \text{MVN}(\mu_k, \Sigma_k)$$

Full posterior:

$$\pi(\theta, \mu, \tau \mid D) \propto \left[\prod_{g=1}^{G} q_g(\theta_g)\right] \left[\prod_{g=1}^{G} \text{Normal}(\theta_g \mid \mu, \tau^2)\right] \pi(\mu)\pi(\tau)$$

where $q_g(\theta_g)$ approximates the Stage 1 posterior for group g.

What's Fixed vs. Flexible:

Fixed:

- Hierarchical distribution: $\theta_g \mid \mu, \tau \sim \text{Normal}(\mu, \tau^2)$

Flexible:

- Hyperpriors $\pi(\mu)$ and $\pi(\tau)$: Normal, Student-t, Cauchy, Lognormal, Gamma, Inverse-Gamma, Exponential, Uniform, mixtures (including spike-and-slab), and truncated versions
- Stage 1 posteriors: Can be non-Normal (handled by mixture approximation)

Critical Requirements:

1. **Stage 1 must use flat/uninformative priors** on θ_g
   - Ensures two-stage = one-stage hierarchical model
   - Stan: Don't specify prior (defaults to flat)
   - JAGS/NIMBLE: Use very wide priors
2. **Verify mixture quality:** `plot(mixture, draws = samples)`
   - Check density overlays and QQ plots
   - Poor approximation → biased shrinkage
3. **Check prior implications:** `sample_prior_predictive(hierarchical_priors)`
   - Understand what priors imply before fitting
   - Avoid prior-data conflicts
4. **Minimum 2 groups required** for heterogeneity estimation

Common Prior Choices for τ:

- Half-Normal: `dist_truncated(dist_normal(0, s), lower = 0)` - Weakly informative
- Half-t: `dist_truncated(dist_student_t(df, 0, s), lower = 0)` - Heavier tails
- Half-Cauchy: `dist_truncated(dist_cauchy(0, s), lower = 0)` - Very diffuse
- Uniform: `dist_uniform(0, U)` - Bounded heterogeneity
- Inverse-Gamma: `dist_inverse_gamma(a, b)` - Traditional choice

See `vignette("getting_started")` for complete workflow, `vignette("brms_integration")` for real examples, and package README for mathematical justification.

Value

A `shrinkr_fit` object (list) containing:

`fit`	Stan model object
`data`	Data list used for fitting
`summary`	Parameter summaries (mean, sd, quantiles, Rhat, ESS)
`diagnostics`	Sampler diagnostics (divergences, treedepth)
`priors`	Prior specifications used

See Also

Workflow functions: `fit_mixture()`, `sample_prior_predictive()`

Extract results: `extract_mu_tau()`, `extract_theta()`, `summarize_mu_tau()`, `summarize_theta()`, `theta_contrasts()`

Visualization: `plot.shrinkr_fit()`, `plot.shrinkr_mixture()`

Vignettes:

- `vignette("getting_started")` - Complete workflow with Stan example
- `vignette("brms_integration")` - Survival analysis example

Examples

```
## Not run:
# This example fits a Stan model, so it is not run during package checks.
priors <- list(
  mu = distributional::dist_normal(0, 10),
  tau = distributional::dist_truncated(distributional::dist_student_t(3, 0, 2.5), lower = 0)
)
fit <- shrink(
  mle = c(0.5, 1.2, -0.3),
  var_matrix = c(0.1, 0.15, 0.12),
  hierarchical_priors = priors,
  iter = 1000, chains = 2, seed = 1
)
summarise_theta(fit)

## End(Not run)
```

`shrinkr-imports` *Imports from stats*

Description

Imports from stats

`summarise_mu_tau` *Summarize mu and tau hyperparameters*

Description

Computes posterior summaries for the hierarchical hyperparameters (`mu`, `tau`, and `tau_squared`). Returns a data frame with one row per parameter containing posterior means, standard deviations, quantiles, and convergence diagnostics.

This is a focused alternative to `summary(fit)`, which returns summaries for all parameters including theta.

Usage

```
summarise_mu_tau(fit, probs = c(0.025, 0.5, 0.975), measures = NULL)

summarize_mu_tau(fit, probs = c(0.025, 0.5, 0.975), measures = NULL)
```

Arguments

`fit`	A `shrinkr_fit` object from `shrink()`.
`probs`	Numeric vector of quantiles to compute. Default is `c(0.025, 0.5, 0.975)` for 95% credible intervals.
`measures`	Optional character vector or list of summary measures to compute. If `NULL`, uses mean, sd, and convergence diagnostics.

Value

A data frame (tibble if available) with one row per parameter and columns:

`parameter` Parameter name (`mu`, `tau`, or `tau_squared`)

`mean` Posterior mean

`sd` Posterior standard deviation

`q2.5`, `q50`, `q97.5` Quantiles (or custom quantiles from `probs`)

`rhat` R-hat convergence diagnostic

`ess_bulk` Effective sample size (bulk)

`ess_tail` Effective sample size (tail)

See Also

`shrink()` for fitting models, `extract_mu_tau()` for raw hyperparameter draws, `summarise_theta()` for group-level summaries

Examples

```
set.seed(1)
draws <- data.frame(
  mu = rnorm(20, 0.2, 0.05),
  tau = abs(rnorm(20, 0.3, 0.03)),
  `theta[1]` = rnorm(20, 0.0, 0.1),
  `theta[2]` = rnorm(20, 0.3, 0.1),
  `theta[3]` = rnorm(20, 0.5, 0.1),
  check.names = FALSE
)
draws$tau_squared <- draws$tau^2
fit <- list(
  fit = posterior::as_draws_df(draws),
  data = list(
    G = 3, K = 1, centered = FALSE,
    vars = c(”group1”, ”group2”, ”group3”),
    quantiles = data.frame(
      q2.5 = c(-0.20, 0.10, 0.30),
      q50 = c(0.00, 0.30, 0.50),
      q97.5 = c(0.20, 0.50, 0.70)
    )
  ),
  summary = posterior::summarise_draws(
```

```
    posterior::as_draws_df(draws),
    "mean", "sd",
    ~posterior::quantile2(., probs = c(0.025, 0.5, 0.975))
  ),
  diagnostics = list(n_divergent = 0, max_treedepth = 0, n_leapfrog = 0)
)
class(fit) <- "shrinkr_fit"
summarise_mu_tau(fit)
summarise_mu_tau(fit, probs = c(0.05, 0.5, 0.95))
```

`summarise_theta` *Summarize theta parameters by group*

Description

Computes posterior summaries for subgroup effects (theta parameters). Returns a data frame with one row per group containing posterior means, standard deviations, quantiles, and convergence diagnostics.

This is a focused alternative to `summary(fit)`, which returns summaries for all parameters including mu and tau.

Usage

```
summarise_theta(
  fit,
  probs = c(0.025, 0.5, 0.975),
  group_names = NULL,
  measures = NULL
)

summarize_theta(
  fit,
  probs = c(0.025, 0.5, 0.975),
  group_names = NULL,
  measures = NULL
)
```

Arguments

`fit`	A `shrinkr_fit` object from `shrink()`.
`probs`	Numeric vector of quantiles to compute. Default is `c(0.025, 0.5, 0.975)` for 95% credible intervals.
`group_names`	Optional character vector of length G to label groups. If `NULL`, uses names from `fit$data$vars` or defaults to "group1", etc.
`measures`	Optional character vector or list of summary measures to compute. If `NULL`, uses mean, sd, and convergence diagnostics.

Value

A data frame (tibble if available) with one row per group and columns:

`group`	Group identifier
`mean`	Posterior mean
`sd`	Posterior standard deviation
`q2.5`, `q50`, `q97.5`	Quantiles (or custom quantiles from `probs`)
`rhat`	R-hat convergence diagnostic
`ess_bulk`	Effective sample size (bulk)
`ess_tail`	Effective sample size (tail)

See Also

`shrink()` for fitting models, `summarise_mu_tau()` for hyperparameter summaries, `theta_contrasts()` for computing contrasts

Examples

```
set.seed(1)
draws <- data.frame(
  mu = rnorm(20, 0.2, 0.05),
  tau = abs(rnorm(20, 0.3, 0.03)),
  `theta[1]` = rnorm(20, 0.0, 0.1),
  `theta[2]` = rnorm(20, 0.3, 0.1),
  `theta[3]` = rnorm(20, 0.5, 0.1),
  check.names = FALSE
)
draws$tau_squared <- draws$tau^2
fit <- list(
  fit = posterior::as_draws_df(draws),
  data = list(
    G = 3, K = 1, centered = FALSE,
    vars = c("group1", "group2", "group3"),
    quantiles = data.frame(
      q2.5 = c(-0.20, 0.10, 0.30),
      q50 = c(0.00, 0.30, 0.50),
      q97.5 = c(0.20, 0.50, 0.70)
    )
  ),
  summary = posterior::summarise_draws(
    posterior::as_draws_df(draws),
    "mean", "sd",
    ~posterior::quantile2(., probs = c(0.025, 0.5, 0.975))
  ),
  diagnostics = list(n_divergent = 0, max_treedepth = 0, n_leapfrog = 0)
)
class(fit) <- "shrinkr_fit"
summarise_theta(fit)
summarise_theta(fit, probs = c(0.05, 0.5, 0.95))
summarise_theta(fit, group_names = c("Control", "A", "B"))
```

`summary.shrinkr_fit` *Summary method for shrinkr_fit*

Description

Comprehensive posterior summary including hyperparameters and all subgroup effects with convergence diagnostics.

Usage

```
## S3 method for class 'shrinkr_fit'
summary(
  object,
  probs = c(0.025, 0.5, 0.975),
  group_names = NULL,
  digits = 3,
  ...
)
```

Arguments

`object`	A `shrinkr_fit` object.
`probs`	Numeric vector of quantiles to compute. Default is `c(0.025, 0.5, 0.975)`.
`group_names`	Optional character vector to label groups.
`digits`	Number of digits to display. Default is 3.
`...`	Additional arguments (currently unused).

Value

Invisibly returns a list with `mu_tau` and `theta` summary tables. Prints formatted output.

See Also

`shrink()` for fitting models, `summarise_theta()` for theta-only summaries

`summary.shrinkr_mixture` *Summary statistics for mixture fits*

Description

Computes comprehensive summary statistics for fitted mixture components, including component-wise and variable-wise summaries.

Usage

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

Arguments

`object`	A `shrinkr_mixture` object from `fit_mixture`.
`...`	Additional arguments (currently unused).

Value

A list with class "summary.shrinkr_mixture" containing:

`components`	Data frame with component weights and sizes
`by_variable`	Data frame with per-variable mixture summaries
`model_info`	List with model selection details
`diagnostics`	Fit diagnostics

See Also

`fit_mixture` for fitting mixture models, `print.shrinkr_mixture` for quick overview

Examples

```
set.seed(1)
samples <- list(
  group1 = matrix(rnorm(100, 0.0, 0.5), ncol = 1),
  group2 = matrix(rnorm(100, 0.5, 0.5), ncol = 1)
)
mix <- fit_mixture(samples, K_max = 2, verbose = FALSE)
summ <- summary(mix)
summ$components
```

`summary.shrinkr_prior_pred`

Summary statistics for prior predictive samples

Description

Computes comprehensive summary statistics for hyperparameters (mu and tau) and theta parameters from prior predictive samples.

Usage

```
## S3 method for class 'shrinkr_prior_pred'
summary(object, probs = c(0.025, 0.5, 0.975), ...)
```

Arguments

`object`	A `shrinkr_prior_pred` object from `sample_prior_predictive`.
`probs`	Numeric vector of quantiles to compute. Default is c(0.025, 0.5, 0.975) for 95% credible intervals.
`...`	Additional arguments (currently unused).

Value

A list with class "summary.shrinkr_prior_pred" containing:

`hyperparameters`	Data frame with summary statistics for mu and tau
`theta`	Data frame with summary statistics for each group's theta

Each data frame includes columns for mean, sd, and the requested quantiles.

See Also

`sample_prior_predictive` for generating samples, `print.shrinkr_prior_pred` for quick overview

Examples

```
priors <- list(
  mu = distributional::dist_normal(0, 5),
  tau = distributional::dist_truncated(distributional::dist_normal(0, 1), lower = 0)
)
prior_pred <- sample_prior_predictive(priors, n_groups = 3, n_draws = 50)
summ <- summary(prior_pred)
summ$theta
```

`theta_contrasts` *Linear combinations of theta*

Description

Computes posterior draws for linear combinations of subgroup effects. Useful for pairwise contrasts (e.g., treatment vs control), weighted averages, or any custom linear estimand involving theta parameters.

Usage

```
theta_contrasts(fit, contrast_matrix, labels = NULL)
```

Arguments

`fit`	A `shrinkr_fit` object from `shrink()`.
`contrast_matrix`	A numeric matrix L with `ncol(L) = G` (number of groups) and `nrow(L) = M` (number of contrasts). Each row defines one linear combination: $contrast_i = L_{i1}\theta_1 + L_{i2}\theta_2 + \ldots + L_{iG}\theta_G$
`labels`	Optional character vector of length M to name the contrasts. If `NULL`, uses "contrast1", "contrast2", etc.

Value

A `posterior::draws_df` with columns `.chain`, `.iteration`, `.draw`, and one column per contrast.

See Also

`shrink()` for fitting models, `summarise_theta()` for basic theta summaries

Examples

```
set.seed(1)
draws <- data.frame(
  mu = rnorm(20, 0.2, 0.05),
  tau = abs(rnorm(20, 0.3, 0.03)),
  `theta[1]` = rnorm(20, 0.0, 0.1),
  `theta[2]` = rnorm(20, 0.3, 0.1),
  `theta[3]` = rnorm(20, 0.5, 0.1),
  check.names = FALSE
)
draws$tau_squared <- draws$tau^2
fit <- list(
  fit = posterior::as_draws_df(draws),
  data = list(
    G = 3, K = 1, centered = FALSE,
    vars = c("group1", "group2", "group3"),
    quantiles = data.frame(
      q2.5 = c(-0.20, 0.10, 0.30),
      q50 = c(0.00, 0.30, 0.50),
      q97.5 = c(0.20, 0.50, 0.70)
    )
  ),
  summary = posterior::summarise_draws(
    posterior::as_draws_df(draws),
    "mean", "sd",
    ~posterior::quantile2(., probs = c(0.025, 0.5, 0.975))
  ),
  diagnostics = list(n_divergent = 0, max_treedepth = 0, n_leapfrog = 0)
)
class(fit) <- "shrinkr_fit"
L <- matrix(c(-1, 1, 0), nrow = 1)
```

```
contrast <- theta_contrasts(fit, L, labels = "group2_vs_group1")
posterior::summarise_draws(contrast)
```

Index