Package 'powerbrmsINLA'

May 9, 2026

Title Bayesian Power Analysis Using 'brms' and 'INLA'

Version 1.1.1

Maintainer Tony Myers <admyers@aol.com>

Description Provides tools for Bayesian power analysis and assurance calculations using the statistical frameworks of 'brms' and 'INLA'. Includes simulation-based approaches, support for multiple decision rules (direction, threshold, ROPE), sequential designs, and visualisation helpers. Methods are based on Kruschke (2014, ISBN:9780124058880) ``Doing Bayesian Data Analysis: A Tutorial with R, JAGS, and Stan'', O'Hagan & Stevens (2001) <doi:10.1177/0272989X0102100307> ``Bayesian Assessment of Sample Size for Clinical Trials of Cost-Effectiveness'', Kruschke (2018) <doi:10.1177/2515245918771304> ``Rejecting or Accepting Parameter Values in Bayesian Estimation'', Rue et al. (2009) <doi:10.1111/j.1467-9868.2008.00700.x> ``Approximate Bayesian inference for latent Gaussian models by using integrated nested Laplace approximations'', and Bürkner (2017) <doi:10.18637/jss.v080.i01> ``brms: An R Package for Bayesian Multilevel Models using Stan''.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.2

Depends R (>= 4.1.0)

Imports brms (>= 2.19.0), dplyr (>= 1.1.0), ggplot2 (>= 3.4.0), pbapply, rlang (>= 1.1.0), tibble (>= 3.2.0), scales (>= 1.2.0), viridisLite (>= 0.4.0), stats, utils, magrittr (>= 2.0.0)

Suggests INLA (>= 22.05.07), testthat (>= 3.0.0), rmarkdown, MASS, circular, sn

URL https://github.com/Tony-Myers/powerbrmsINLA

BugReports https://github.com/Tony-Myers/powerbrmsINLA/issues

Additional_repositories https://inla.r-inla-download.org/R/stable

NeedsCompilation no

Author Tony Myers [aut, cre] (ORCID: <https://orcid.org/0000-0003-4516-4829>)

Repository CRAN

Date/Publication 2025-11-16 19:30:02 UTC

Contents

`.plot_decision_assurance_curve_from_summary` *Plot decision/assurance curve across n*

Description

Plot decision/assurance curve across n

Usage

```
.plot_decision_assurance_curve_from_summary(
  x,
  y_metric = c("assurance", "power_direction", "power_threshold", "power_rope",
    "bf_hit_10"),
  target = NULL,
  effect_filter = NULL,
  first_n_label = TRUE
)
```

Arguments

x	Engine result list (with $summary) or a data.frame.
y_metric	One of "assurance","power_direction","power_threshold","power_rope","bf_hit_10".
target	Optional horizontal target line.
effect_filter	Named list for exact-match filtering (e.g., list(treatment=0.5)).
first_n_label	If TRUE, annotate first n reaching target.

Value

ggplot object.

---

add_decision_overlay *Add sample-size decision overlay to an assurance contour*

Description

Overlays the output of `decide_sample_size()` as a step-line on a contour plot (e.g. from `plot_power_contour()`).

Usage

```
add_decision_overlay(p, decisions, x_effect = NULL, colour = "red")
```

Arguments

p	A ggplot object, typically from `plot_power_contour()`.
decisions	A data.frame or tibble returned by `decide_sample_size()`, containing at least one effect column and `n_recommended`.
x_effect	Name of the effect column to use on the x-axis. If `NULL`, the first column that is not `n_recommended` or `rationale` is used.
colour	Colour for the overlay line (default "red").

Details

If the decisions contain multiple effect-grid columns, the overlay is aggregated over all effect columns except `x_effect` by taking the worst-case (maximum) recommended n at each value of `x_effect`.

Value

A ggplot object.

beta_binom_power *Analytic Assurance for Beta-Binomial Designs*

Description

Computes assurance (power) using generating and audience Beta priors for a binomial count via a Beta-Binomial predictive distribution.

Usage

```
beta_binom_power(
  n,
  gen_prior_a,
  gen_prior_b,
  aud_prior_a,
  aud_prior_b,
  hdi_mass = 0.95,
  rope = NULL,
  hdi_max_width = NULL
)
```

Arguments

`n`	Sample size (number of trials).
`gen_prior_a, gen_prior_b`	Generating Beta prior parameters.
`aud_prior_a, aud_prior_b`	Audience Beta prior parameters.
`hdi_mass`	HDI mass (e.g., 0.95).
`rope`	Length-2 numeric vector for ROPE bounds, or NULL for max-width rule.
`hdi_max_width`	Positive width threshold for the HDI (used if `rope=NULL`).

Value

Assurance value between 0 and 1.

beta_weights_on_grid *Beta-Prior Weights Over an Effect Grid*

Description

Computes prior weights over a grid of true effect values by evaluating a Beta(mode, n) prior. If the grid is not in (0,1), it is rescaled linearly.

Usage

```
beta_weights_on_grid(effects, mode, n)
```

Arguments

effects	Numeric vector of effect values (grid).
mode	Prior mode in (0,1).
n	Prior concentration (> 2).

Value

Normalised numeric weights over the grid (sum to 1).

---

brms_inla_power *Core Bayesian Assurance / Power Simulation (Modern, Multi-Effect Ready)*

Description

Provides Bayesian power analysis and assurance calculation using INLA (Integrated Nested Laplace Approximation) for efficient computation. Implements simulation-based power analysis for generalized linear mixed models with automatic threading optimization.

Usage

```
brms_inla_power(
  formula,
  family = gaussian(),
  family_control = NULL,
  Ntrials = NULL,
  E = NULL,
  scale = NULL,
  priors = NULL,
  data_generator = NULL,
  effect_name,
  effect_grid = 0.5,
  sample_sizes = c(50, 100, 200, 400),
  nsims = 200,
  power_threshold = 0.8,
  precision_target = NULL,
  prob_threshold = 0.95,
  effect_threshold = 0,
  credible_level = 0.95,
  rope_bounds = NULL,
  error_sd = 1,
  group_sd = 0.5,
```

```
  obs_per_group = 10,
  predictor_means = NULL,
  predictor_sds = NULL,
  seed = 123,
  inla_hyper = NULL,
  compute_bayes_factor = FALSE,
  bf_method = c("sd", "marglik"),
  bf_cutoff = 10,
  inla_num_threads = NULL,
  progress = c("auto", "text", "none"),
  family_args = list()
)
```

Arguments

formula	Model formula.
family	brms GLM family (e.g., gaussian(), binomial()).
family_control	Optional list for INLA's control.family.
Ntrials	Optional vector for binomial trials.
E	Optional vector for Poisson exposure.
scale	Optional vector scale parameter for INLA families.
priors	Optional brms::prior specification.
data_generator	Optional function(n, effect) returning a dataset.
effect_name	Character vector of fixed effect names.
effect_grid	Vector/data.frame of effect values (supports multi-effect). For single effects, use a numeric vector. For multiple effects, use a data.frame with column names matching effect_name.
sample_sizes	Vector of sample sizes.
nsims	Number of simulations per cell.
power_threshold	Decision probability threshold for summary.
precision_target	Optional credible interval width target.
prob_threshold	Posterior probability threshold for decision rules.
effect_threshold	Effect-size threshold.
credible_level	Credible interval level (default 0.95).
rope_bounds	Optional Region of Practical Equivalence bounds (length 2 vector).
error_sd	Gaussian residual standard deviation.
group_sd	Random effects standard deviation.
obs_per_group	Observations per group.
predictor_means	Optional named list of predictor means.

predictor_sds	Optional named list of predictor standard deviations.
seed	Random seed.
inla_hyper	Optional INLA-specific hyperparameters.
compute_bayes_factor	Logical, compute Bayes Factor if TRUE.
bf_method	Character. "sd" = Savage-Dickey at 0 (requires proper Normal prior on the tested coefficient); "marglik" = marginal-likelihood Bayes factor via INLA by comparing full vs reduced model (slower).
bf_cutoff	Numeric Bayes-factor threshold for declaring a "hit" (default 10).
inla_num_threads	Character string specifying INLA threading (e.g., "4:1" for 4 threads). If NULL (default), automatically detects optimal setting: "4:1" for 4+ cores, "2:1" for 2-3 cores, "1:1" otherwise.
progress	One of "auto", "text", or "none" for progress display.
family_args	List of arguments for family-specific data generators.

Value

List with results, summary, and settings.

`brms_inla_power_design` *Design-based wrapper for Bayesian power / assurance*

Description

Dispatches to one of the three engines depending on `design`. This function must accept `...` and pass it on unchanged.

Usage

```
brms_inla_power_design(design = c("fixed", "two_stage", "sequential"), ...)
```

Arguments

design	Character scalar: "fixed", "two_stage", or "sequential".
...	Arguments passed on to the corresponding engine.

Value

Whatever the underlying engine returns.

`brms_inla_power_parallel` *Parallel wrapper for fixed-design Bayesian power / assurance simulations*

Description

Parallelises over cells defined by sample_sizes x effect_grid for the fixed-n engine brms_inla_power().

Usage

```
brms_inla_power_parallel(
  design = c("fixed"),
  sample_sizes,
  effect_grid,
  nsims,
  n_cores = max(1L, parallel::detectCores() - 1L),
  seed = 123L,
  progress = c("auto", "text", "none"),
  ...
)
```

Arguments

`design`	Character scalar. Currently only "fixed" is supported.
`sample_sizes`	Numeric vector of sample sizes (required).
`effect_grid`	Numeric vector or data frame defining effect scenarios (required).
`nsims`	Integer number of simulations per cell.
`n_cores`	Integer number of worker processes. Default is max(1L, parallel::detectCores() - 1L).
`seed`	Integer base seed. Each cell uses seed + cell_id.
`progress`	Logical or character; controls wrapper-level progress bar.
`...`	Further arguments passed directly to brms_inla_power(), such as formula, family, priors, effect_name, compute_bayes_factor, bf_method, inla_hyper, inla_num_threads, etc.

Value

A list with components summary, results, and settings.

brms_inla_power_sequential *Sequential Bayesian Assurance Simulation Engine (Modern, Multi-Effect Ready)*

Description

Simulates assurance sequentially in batches, stopping early per cell based on Wilson confidence intervals.

Usage

```
brms_inla_power_sequential(
  formula,
  family = gaussian(),
  family_control = NULL,
  Ntrials = NULL,
  E = NULL,
  scale = NULL,
  priors = NULL,
  data_generator = NULL,
  effect_name,
  effect_grid,
  sample_sizes,
  metric = c("direction", "threshold", "rope", "bf"),
  target = 0.8,
  prob_threshold = 0.95,
  effect_threshold = 0,
  rope_bounds = NULL,
  credible_level = 0.95,
  compute_bayes_factor = FALSE,
  error_sd = 1,
  group_sd = 0.5,
  obs_per_group = 10,
  predictor_means = NULL,
  predictor_sds = NULL,
  seed = 1,
  batch_size = 20,
  min_sims = 40,
  max_sims = 600,
  ci_conf = 0.95,
  margin = 0.02,
  inla_num_threads = NULL,
  family_args = list(),
  progress = TRUE
)
```

Arguments

formula	brms-style model formula.
family	GLM family (e.g., gaussian(), binomial()).
family_control	Optional list for INLA's control.family.
Ntrials	Optional vector of binomial trial counts (for binomial families).
E	Optional vector of exposures (for Poisson families).
scale	Optional numeric vector for scale parameter in INLA.
priors	brms prior specification object.
data_generator	Optional function(n, effect) to simulate data.
effect_name	Character vector of fixed effects to assess.
effect_grid	Data frame or vector of effect values.
sample_sizes	Vector of sample sizes.
metric	Character; one of "direction", "threshold", "rope", or "bf" for Bayesian decision metric.
target	Target assurance value for stopping.
prob_threshold	Posterior probability threshold for decision metrics.
effect_threshold	Effect-size threshold.
rope_bounds	Numeric length-2 vector defining ROPE.
credible_level	Credible interval level for Bayesian inference.
compute_bayes_factor	Logical; TRUE if metric is "bf".
error_sd	Residual standard deviation.
group_sd	Standard deviation of random effects.
obs_per_group	Number of observations per group.
predictor_means	Optional named list of predictor means.
predictor_sds	Optional named list of predictor standard deviations.
seed	Random seed.
batch_size	Number of simulations per sequential look.
min_sims	Minimum simulations before early stopping.
max_sims	Maximum simulations per cell.
ci_conf	Confidence level for Wilson confidence intervals.
margin	Margin around target for early stopping decision.
inla_num_threads	Character string specifying INLA threading (e.g., "4:1"). If NULL (default), automatically detects optimal setting based on CPU cores.
family_args	List of family-specific args passed to data generator.
progress	Logical; if TRUE, show progress messages.

Details

Sequential Bayesian Assurance Simulation Engine (Modern, Multi-Effect Ready)

Simulates assurance sequentially in batches, stopping early per cell based on Wilson confidence intervals.

Value

List containing summary per cell and simulation settings.

Examples

```
## Not run:
# Sequential design with automatic threading
results <- brms_inla_power_sequential(
  formula = outcome ~ treatment,
  effect_name = "treatment",
  effect_grid = c(0.2, 0.5, 0.8),
  sample_sizes = c(50, 100, 200),
  metric = "direction",
  target = 0.80
)
print(results$summary)

## End(Not run)
```

---

brms_inla_power_two_stage

Two-Stage Bayesian Assurance Simulation (Multi-Effect, User-Friendly API)

Description

Runs a two-stage Bayesian assurance simulation with formula-based multi-effect grids and adaptive refinement.

Usage

```
brms_inla_power_two_stage(
  formula,
  effect_name,
  effect_grid,
  n_range,
  stage1_k_n = 8,
  stage1_nsims = 100,
  stage2_nsims = 400,
  refine_metric = c("direction", "threshold", "rope"),
  refine_target = 0.8,
```

```
  prob_threshold = 0.95,
  effect_threshold = 0,
  obs_per_group = NULL,
  error_sd = NULL,
  group_sd = 0.5,
  band = 0.06,
  expand = 1L,
  inla_num_threads = NULL,
  ...
)
```

Arguments

formula	Model formula.
effect_name	Character vector of fixed effect names; must match formula terms.
effect_grid	Data frame with columns named by effect_name specifying effect values.
n_range	Numeric length-2 vector specifying sample size range.
stage1_k_n	Number of grid points in stage 1.
stage1_nsims	Number of simulations per cell in stage 1.
stage2_nsims	Number of simulations per cell in stage 2.
refine_metric	Metric used for refinement; one of "direction", "threshold", or "rope".
refine_target	Target assurance for refined cells.
prob_threshold	Posterior probability threshold for decision.
effect_threshold	Effect-size threshold for decision metric.
obs_per_group	Number of observations per group for grouping factors.
error_sd	Residual standard deviation.
group_sd	Standard deviation of random effects.
band	Numeric width of the target refinement band.
expand	Integer; how much to expand the refinement grid around candidates.
inla_num_threads	Character string specifying INLA threading (e.g., "4:1"). If NULL (default), automatically detects optimal setting based on CPU cores.
...	Additional arguments passed to internal functions.

Value

A list with combined simulation results, summary, and stage parameters.

Examples

```
## Not run:
# Two-stage design with threading
effect_grid <- expand.grid(
  treatment = c(0.2, 0.5, 0.8),
  covariate = c(0.1, 0.3)
)
results <- brms_inla_power_two_stage(
  formula = outcome ~ treatment + covariate,
  effect_name = c("treatment", "covariate"),
  effect_grid = effect_grid,
  n_range = c(50, 200),
  stage1_nsims = 3,
  stage2_nsims = 3,
   error_sd = 1
)
print(results$summary)

## End(Not run)
```

`decide_sample_size` *Decide recommended sample size from power/assurance results*

Description

Returns the smallest n per effect setting that meets user-specified targets. Works with both brms_inla_power() and brms_inla_power_sequential() outputs.

Usage

```
decide_sample_size(
  x,
  direction = NULL,
  threshold = NULL,
  rope_in = NULL,
  bf10 = NULL,
  bf_prop_min = 0,
  targets = NULL
)
```

Arguments

`x`	A list with `$summary` (engine output) or a data.frame summary itself.
`direction`	Numeric in $[0, 1]$, required power for `power_direction` (optional).
`threshold`	Numeric in $[0, 1]$, required power for `power_threshold` (optional).
`rope_in`	Numeric in $[0, 1]$, maximum allowed Pr(in ROPE) (optional). Note: since summaries usually contain `power_rope = Pr(outside ROPE) >= prob_threshold`, we compare `(1 - power_rope) <= rope_in` when `rope_in` is given.

bf10	Numeric Bayes-factor cutoff (e.g., 10). If provided, we look for a column named bf_hit_<bf10>; if not found, we fall back to any bf_hit_* column present.
bf_prop_min	Numeric in $[0, 1]$, the minimum proportion of simulations that must achieve BF >= bf10 (default 0).
targets	Optional list alternative to the direct args. Ignored if any direct arg is non-NULL.

Details

You can pass targets directly via arguments (direction, threshold, rope_in, bf10) or via `targets = list(direction=..., threshold=..., rope_in=..., bf10=...)`. Direct arguments take precedence if supplied.

Value

A data.frame with recommended n per effect combination and the rationale.

---

hdi_of_icdf *Highest Density Interval from an Inverse CDF*

Description

Computes an HDI of given mass from any distribution for which you have a quantile function (inverse CDF).

Usage

```
hdi_of_icdf(qfun, width = 0.95, tol = 1e-08, ...)
```

Arguments

qfun	Quantile function, e.g., qbeta, qnorm, ...
width	Desired HDI mass (e.g., 0.95).
tol	Optimizer tolerance.
...	Additional arguments passed to qfun.

Value

Named numeric vector with elements ll and ul.

`min_n_beta_binom` *Minimum n for Target Assurance (Beta-Binomial)*

Description

Minimum n for Target Assurance (Beta-Binomial)

Usage

```
min_n_beta_binom(
  gen_prior_mode,
  gen_prior_n,
  desired_power,
  aud_prior_mode = 0.5,
  aud_prior_n = 2,
  hdi_mass = 0.95,
  rope = NULL,
  hdi_max_width = NULL,
  n_start = 20,
  n_max = 1e+05,
  verbose = TRUE
)
```

Arguments

`gen_prior_mode`	Generating prior mode in (0,1).
`gen_prior_n`	Generating prior concentration (>= 2).
`desired_power`	Target assurance value in (0,1).
`aud_prior_mode`	Audience prior mode in (0,1).
`aud_prior_n`	Audience prior concentration (>= 2).
`hdi_mass`	HDI mass (e.g., 0.95).
`rope`	Length-2 numeric vector for ROPE bounds, or NULL for max-width rule.
`hdi_max_width`	Positive width threshold for the HDI (used if `rope=NULL`).
`n_start`	Starting sample size for search.
`n_max`	Maximum sample size to try.
`verbose`	If TRUE, prints progress.

Value

Smallest n meeting the target assurance.

`plot_assurance_with_robustness` *Plot Assurance with Robustness Ribbon (Multi-Effect Grid Friendly)*

Description

Compares assurance results from multiple scenarios by showing the range ("ribbon") of values across scenarios for each sample size and effect grid variable.

Usage

```
plot_assurance_with_robustness(
  power_results_list,
  metric = c("precision", "direction", "threshold", "bf"),
  x_effect = NULL,
  facet_by = NULL,
  precision_target = NULL,
  p_star = 0.95,
  bf_threshold = 10,
  effect_filters = NULL,
  effect_weights = NULL,
  show_individual_scenarios = FALSE,
  title = NULL,
  subtitle = NULL
)
```

Arguments

`power_results_list`	Named list of results objects from `brms_inla_power` or sequential/two-stage variants.
`metric`	Which assurance metric to compute: "precision", "direction", "threshold", or "bf".
`x_effect`	Name of effect grid column for x-axis (default: first detected grid column).
`facet_by`	Optional effect grid column(s) to facet by.
`precision_target`	CI width target if metric="precision".
`p_star`	Posterior probability threshold for "direction"/"threshold".
`bf_threshold`	BF10 threshold for "bf".
`effect_filters`	Optional named list for filtering rows (e.g. list(treatment=0)).
`effect_weights`	Optional named numeric vector for averaging over grid values.
`show_individual_scenarios`	Logical; if TRUE, overlay each scenario's curve.
`title, subtitle`	Optional plot labels.

Value

A ggplot object.

`plot_bf_assurance_curve` *Bayes-factor assurance curve (user-facing wrapper)*

Description

This is the main function users should call to visualise Bayes-factor "power" / assurance. It is a thin wrapper around `plot_bf_assurance_curve_smooth()`, so all existing behaviour is preserved.

Usage

```
plot_bf_assurance_curve(power_results, bf_threshold = 10, effect_filter = NULL)
```

Arguments

`power_results`	Output from a brms_inla_power* function (or a data.frame with columns n, bf10 and any effect columns).
`bf_threshold`	Numeric Bayes factor cutoff (default 10).
`effect_filter`	Optional named list to filter effect-grid columns, e.g. `list(treatment = 0.3)`.

Value

A ggplot object.

`plot_bf_assurance_curve_smooth` *Bayes-factor assurance curve with Wilson CIs (multi-effect grid friendly)*

Description

Plots the proportion of simulations in which BF10 meets or exceeds a threshold, grouped by sample size and any effect grid variables.

Usage

```
plot_bf_assurance_curve_smooth(x, cutoff = 10, effect_filter = NULL)
```

Arguments

x	Engine result (list with $results) or a data.frame with at least columns n, bf10 and any effect columns (e.g., treatment).
cutoff	Numeric Bayes factor threshold for a "hit" (default 10).
effect_filter	Optional named list to filter effects, e.g. list(treatment = 0.6).

Value

A ggplot object.

---

plot_bf_expected_evidence

Plot Expected Evidence (mean log10 BF10, Multi-Effect Grid Friendly)

Description

Plots the average log10 BF10 against any effect grid variable, grouped/faceted.

Usage

```
plot_bf_expected_evidence(
  power_results,
  x_effect = NULL,
  facet_by = NULL,
  n = NULL,
  agg_fun = mean,
  title = NULL,
  subtitle = NULL
)
```

Arguments

power_results	Simulation results from a brms_inla_power* function with compute_bayes_factor = TRUE.
x_effect	Name of effect grid column for x-axis (default: first grid column).
facet_by	Optional grid column(s) to facet by (default: NULL).
n	Optional sample size to filter to (NULL means plot all; else one curve per grid/facet).
agg_fun	Aggregation function if >1 entries per cell (default: mean).
title, subtitle	Optional plot labels.

Value

A ggplot object.

`plot_bf_heatmap` *Plot Bayes Factor Heatmap (mean log10 BF10, Multi-Effect Grid Friendly)*

Description

Heatmap of mean log10 BF10 as a function of two effect grid columns (x/y), with optional faceting.

Usage

```
plot_bf_heatmap(
  power_results,
  x_effect = NULL,
  y_effect = "n",
  facet_by = NULL,
  n = NULL,
  agg_fun = mean,
  title = NULL,
  subtitle = NULL
)
```

Arguments

`power_results`	Simulation results from a `brms_inla_power*` function with `compute_bayes_factor = TRUE`.
`x_effect`	Name of effect grid column for x-axis (default: first grid column).
`y_effect`	Name of effect grid column for y-axis (default: "n").
`facet_by`	Optional column(s) to facet by.
`n`	Optional sample size to filter to (NULL means plot all; else show only that n).
`agg_fun`	Aggregation function (default: mean).
`title, subtitle`	Optional plot labels.

Value

ggplot object.

`plot_decision_assurance_curve` *Plot Decision Assurance Curve (Multi-Effect Grid Friendly)*

Description

Plots the assurance (proportion of simulation runs meeting a posterior probability decision rule) versus an effect grid variable, for a given metric ("direction", "threshold", or "rope") at a fixed decision probability threshold p_star.

Usage

```
plot_decision_assurance_curve(
  power_results,
  metric = c("direction", "threshold", "rope"),
  p_star = 0.95,
  x_effect = NULL,
  facet_by = NULL,
  effect_filters = NULL,
  effect_weights = NULL,
  title = NULL,
  subtitle = NULL
)
```

Arguments

`power_results`	A list returned by `brms_inla_power*`.
`metric`	Decision metric: "direction", "threshold", or "rope".
`p_star`	Numeric decision threshold in (0,1).
`x_effect`	Name of effect grid column for x-axis (default: first grid column).
`facet_by`	Optional effect grid column(s) to facet by.
`effect_filters`	Optional named list for filtering rows, e.g. list(treatment=0).
`effect_weights`	Optional named numeric vector of weights for selected x_effect values.
`title, subtitle`	Optional plot labels.

Value

A ggplot object.

`plot_decision_threshold_contour` *Plot Decision Threshold Contour (Multi-Effect Grid Friendly)*

Description

Shows assurance as a function of decision threshold p* and one effect grid column, optionally faceted.

Usage

```
plot_decision_threshold_contour(
  power_results,
  metric = c("direction", "threshold", "rope"),
  p_star_grid = seq(0.5, 0.99, by = 0.01),
  effect_var = NULL,
  facet_by = NULL,
  effect_value = NULL,
  effect_weights = NULL,
  title = NULL,
  subtitle = NULL
)
```

Arguments

`power_results`	brms_inla_power list (or two-stage, etc.)
`metric`	Which metric: "direction", "threshold", "rope"
`p_star_grid`	Numeric vector of decision thresholds (default: 0.5 to 0.99 by 0.01)
`effect_var`	Name of effect grid column for y-axis (default: first detected grid column)
`facet_by`	Optional effect grid column(s) to facet by
`effect_value`	Optional value(s) to filter for effect_var, or named list for multi-filter
`effect_weights`	Optional weights for aggregation (named by effect_var values)
`title, subtitle`	Optional plot labels.

Value

ggplot2 object.

plot_interaction_surface *Plot Interaction Assurance Surface/Heatmap/Lines (Multi-Effect Grid Friendly)*

Description

Visualizes a metric (e.g., assurance) as a function of two effect grid variables for a fixed sample size or averaged over n. Allows line, heatmap, or contour modes.

Usage

```
plot_interaction_surface(
  data,
  metric,
  effect1,
  effect2,
  n = NULL,
  line = FALSE,
  facet_by = NULL,
  agg_fun = mean,
  title = NULL,
  subtitle = NULL
)
```

Arguments

data	Data frame (typically power_results$summary).
metric	Name of the summary column to plot, e.g. "power_direction", "power_threshold".
effect1	Name of effect grid column for x-axis.
effect2	Name of effect grid column for y-axis or color/facets.
n	Optional sample size to filter to (else averages/plots all n's).
line	Logical; if TRUE, make a lineplot (effect1 on x, one line for each effect2). If FALSE, make a heatmap or contour.
facet_by	Optional grid column(s) to facet by.
agg_fun	Aggregation function if multiple entries per cell (default = mean).
title, subtitle	Optional plot labels.

Value

A ggplot object.

`plot_power_contour` *Draw a filled contour plot of assurance for a chosen metric, as a function of two effect grid columns and sample size.*

Description

Draw a filled contour plot of assurance for a chosen metric, as a function of two effect grid columns and sample size.

Usage

```
plot_power_contour(
  power_results,
  power_metric = c("direction", "threshold", "rope"),
  x_effect = NULL,
  y_effect = "n",
  facet_by = NULL,
  power_threshold = 0.8,
  show_threshold_line = TRUE,
  title = NULL,
  subtitle = NULL
)
```

Arguments

`power_results`	Output from a `brms_inla_power` function.
`power_metric`	Which metric to plot: "direction", "threshold", or "rope".
`x_effect`	Name of effect grid column for x-axis (default = first effect).
`y_effect`	Name of effect grid column for y-axis (default = "n").
`facet_by`	Optional effect grid column(s) to facet by.
`power_threshold`	Optional contour line for assurance (default 0.8).
`show_threshold_line`	Logical; add a red contour at `power_threshold`.
`title, subtitle`	Optional plot labels.

Value

A ggplot object.

plot_power_heatmap *Plot Bayesian Power / Assurance Heatmap (Multi-Effect Grid Friendly)*

Description

Heatmap of assurance for a chosen metric across two selected effect grid variables and sample sizes.

Usage

```
plot_power_heatmap(
  power_results,
  power_metric = c("direction", "threshold", "rope"),
  x_effect = NULL,
  y_effect = "n",
  facet_by = NULL,
  title = NULL,
  subtitle = NULL
)
```

Arguments

power_results	Output from a brms_inla_power function.
power_metric	Which metric to plot: "direction", "threshold", or "rope".
x_effect	Name of effect grid column for x-axis (default = first effect).
y_effect	Name of effect grid column for y-axis (default = "n").
facet_by	Optional effect grid column(s) to facet by.
title, subtitle	Optional plot labels.

Value

A ggplot object.

plot_precision_assurance_curve *Plot Precision Assurance Curve (Multi-Effect Grid Friendly)*

Description

Plots the assurance (proportion of runs meeting CI width <= target) vs. a chosen effect grid variable across sample size(s). Supports faceting, effect filtering, and weights.

Usage

```
plot_precision_assurance_curve(
  power_results,
  precision_target,
  x_effect = NULL,
  facet_by = NULL,
  effect_filters = NULL,
  effect_weights = NULL,
  title = NULL,
  subtitle = NULL
)
```

Arguments

power_results	List returned by brms_inla_power*.
precision_target	Numeric; credible interval width threshold for success.
x_effect	Name of effect grid column for x-axis (default: first grid column).
facet_by	Optional effect grid column(s) for faceting.
effect_filters	Optional named list for filtering rows, e.g. list(treatment=0).
effect_weights	Optional named numeric vector for weights over selected x_effect values.
title, subtitle	Optional plot labels.

Value

A ggplot object.

---

plot_precision_fan_chart

Precision assurance as a function of sample size

Description

Plots the proportion of simulations in which the posterior credible interval width is less than or equal to a target, as a function of sample size n. Optionally colours separate curves by an effect-grid variable.

Usage

```
plot_precision_fan_chart(
  power_results,
  ci_width_target,
  effect_filter = NULL,
  colour_by = NULL,
  title = NULL,
  subtitle = NULL
)
```

Arguments

power_results	Output from a brms_inla_power* function, or a data.frame with at least columns n and ci_width, plus any effect-grid columns (e.g. treatment, age_effect).
ci_width_target	Numeric, target width for the credible interval. Assurance is defined as Pr(ci_width <= ci_width_target).
effect_filter	Optional named list for filtering effect-grid columns, e.g. list(treatment = 0.3).
colour_by	Optional name of an effect-grid column to colour separate curves by. If NULL, only n is used.
title, subtitle	Optional plot labels.

Details

This implementation works directly from the per-simulation results (column ci_width) and does not rely on the robustness engine.

Value

A ggplot object.

Index