

Package: bayprior (via r-universe)

June 4, 2026

Title Bayesian Prior Elicitation and Diagnostics for Clinical Trials

Version 0.2.12

Description A toolkit for constructing, validating, and justifying Bayesian priors in clinical trial settings. Implements expert elicitation via quantile matching, the roulette method, and moment matching across six distribution families, linear and logarithmic expert pooling, prior-data conflict diagnostics including the Box p-value, surprise index, information divergence, and Mahalanobis distance, sensitivity analyses with tornado and influence heatmap plots, sceptical, robust, and power priors, and automated prior justification reports. Includes a fully modular 'Shiny' application for interactive use. Methods based on O'Hagan et al. (2006, ISBN:9780470029886), Box (1980) <[doi:10.2307/2982063](https://doi.org/10.2307/2982063)>, Oakley and O'Hagan (2010) <<https://tonyohagan.co.uk/shelf/>>, Schmidli et al. (2014) <[doi:10.1111/biom.12242](https://doi.org/10.1111/biom.12242)>, Ibrahim and Chen (2000) <[doi:10.1214/ss/1009212673](https://doi.org/10.1214/ss/1009212673)>, Spiegelhalter et al. (1994) <[doi:10.2307/2983527](https://doi.org/10.2307/2983527)>.

License GPL-3

URL <https://github.com/ndohpenngit/bayprior>

BugReports <https://github.com/ndohpenngit/bayprior/issues>

Depends R (>= 4.1.0)

Imports cli (>= 3.6.0), config (>= 0.3.1), dplyr (>= 1.1.0), DT (>= 0.27), ggplot2 (>= 3.4.0), glue (>= 1.6.2), golem (>= 0.4.1), plotly (>= 4.10.1), purrr (>= 1.0.0), rlang (>= 1.2.0), shiny (>= 1.7.4), shinycssloaders (>= 1.0.0), shinydashboard (>= 0.7.2), shinyjs (>= 2.1.0), shinyWidgets (>= 0.7.6), stats

Suggests htmlwidgets (>= 1.5.4), knitr (>= 1.40), patchwork (>= 1.1.0), quarto (>= 1.4), rmarkdown (>= 2.20), shinytest2 (>= 0.3.1), spelling (>= 2.2.0), testthat (>= 3.0.0), vdiff (>= 1.0.0), withr (>= 2.5.0)

VignetteBuilder knitr

Config/Needs/vignettes knitr, rmarkdown

Config/testthat/edition 3
Encoding UTF-8
Language en-GB
Roxygen list(markdown = TRUE)
RoxygenNote 7.3.3
SystemRequirements Quarto CLI (\geq 1.5, <https://quarto.org>)
Repository <https://ndohpenngit.r-universe.dev>
Date/Publication 2026-06-04 13:56:56 UTC
RemoteUrl <https://github.com/ndohpenngit/bayprior>
RemoteRef HEAD
RemoteSha 86e386af3a965bb861bc293b6d221ff416331684

Contents

aggregate_experts	3
as_prior	4
bayprior	5
calibrate_power_prior	6
conflict_mahalanobis	8
elicit_beta	9
elicit_exponential	11
elicit_gamma	12
elicit_lognormal	13
elicit_mixture	14
elicit_normal	15
elicit_roulette	16
elicit_weibull	17
plot.bayprior_power_prior	18
plot_prior_likelihood	19
plot_sensitivity	20
plot_tornado	20
print.bayprior	21
print.bayprior_conflict	21
print.bayprior_conflict_mv	22
print.bayprior_power_prior	22
prior_conflict	23
prior_report	24
robust_prior	25
run_app	27
sceptical_prior	27
sensitivity_cri	29
sensitivity_grid	30
summary.bayprior	31

Index **32**

aggregate_experts *Aggregate multiple expert priors into a consensus prior*

Description

Combines elicited priors from multiple experts into a single consensus distribution using either linear pooling (mixture) or logarithmic pooling (normalised product of densities). Includes diagnostics for inter-expert disagreement.

Usage

```
aggregate_experts(
  priors,
  weights = NULL,
  method = c("linear", "logarithmic"),
  disagreement_threshold = 0.5
)
```

Arguments

priors	A named list of bayprior objects, one per expert.
weights	Numeric vector of expert weights (summing to 1). If NULL, equal weights are applied.
method	Character. "linear" (default) or "logarithmic" pooling.
disagreement_threshold	Numeric in (0, 1). Triggers a warning when the pairwise Bhattacharyya coefficient drops below this value, flagging substantial expert disagreement. Default 0.5.

Details

Linear pooling (externally Bayesian): The consensus density is a weighted mixture $\pi(\theta) = \sum_k w_k \pi_k(\theta)$. This is the most commonly used approach in clinical trial settings (O'Hagan et al., 2006). The resulting prior always lies within the convex hull of individual expert priors.

Logarithmic pooling (internally Bayesian): The consensus density is proportional to $\prod_k \pi_k(\theta)^{w_k}$, which produces a sharper consensus when experts agree, but can be severely influenced by outlying expert opinions.

Value

A bayprior object (dist = "mixture" for linear pooling, dist = "log_pool" for logarithmic), with an additional \$aggregation component containing:

method	Pooling method used
weights	Applied weights
disagreement	Pairwise Bhattacharyya coefficients
n_experts	Number of experts

References

O'Hagan, A., et al. (2006). *Uncertain Judgements: Eliciting Experts' Probabilities*. Wiley.

Examples

```
p1 <- elicit_beta(mean = 0.25, sd = 0.08, method = "moments", expert_id = "E1",
  label = "Response rate")
p2 <- elicit_beta(mean = 0.35, sd = 0.10, method = "moments", expert_id = "E2",
  label = "Response rate")
p3 <- elicit_beta(mean = 0.30, sd = 0.09, method = "moments", expert_id = "E3",
  label = "Response rate")

consensus <- aggregate_experts(
  priors = list(E1 = p1, E2 = p2, E3 = p3),
  weights = c(0.4, 0.3, 0.3),
  method = "linear"
)
print(consensus)
plot(consensus)
```

as_prior

Constructor for bayprior from raw parameters

Description

Construct a bayprior object directly from known hyperparameters (e.g., literature-based priors), bypassing elicitation.

Usage

```
as_prior(dist, params, label = "Prior", expert_id = "Literature")
```

Arguments

dist	Character. One of "beta", "normal", "gamma".
params	Named list of hyperparameters.
label	Character. Description of the quantity.
expert_id	Character. Source identifier.

Value

A bayprior object.

Examples

```
# Historical Beta(2, 8) prior on response rate
prior <- as_prior("beta", list(alpha = 2, beta = 8),
                 label = "Historical response rate prior")
```

bayprior

*bayprior: Bayesian Prior Elicitation for Clinical Trials***Description**

A toolkit for constructing, validating, and justifying Bayesian priors in clinical trial settings. Implements SHELF-style expert elicitation (quantile matching, roulette method, moment matching) across six distribution families, linear and logarithmic expert pooling with compatibility validation, prior-data conflict diagnostics (Box p-value, surprise index, KL divergence, Bhattacharyya overlap, Mahalanobis check) for binary, continuous, Poisson/count, and survival data types, sensitivity analyses with tornado and influence plots, sceptical/robust/power priors, and automated HTML/PDF/Word regulatory reports aligned with FDA/EMA expectations. Includes a fully modular Shiny application with automatic output reset on input change.

Main workflow

1. **Elicitation** – [elicit_beta](#), [elicit_normal](#), [elicit_gamma](#), [elicit_lognormal](#), [elicit_exponential](#), [elicit_weibull](#), [elicit_roulette](#), [elicit_mixture](#)
2. **Expert pooling** – [aggregate_experts](#)
3. **Conflict diagnostics** – [prior_conflict](#), [conflict_mahalanobis](#)
4. **Sensitivity analysis** – [sensitivity_grid](#), [sensitivity_cri](#)
5. **Robust priors** – [sceptical_prior](#), [robust_prior](#), [calibrate_power_prior](#)
6. **Reporting** – [prior_report](#)
7. **Shiny app** – [run_app](#)

Distribution families

beta Response rates and proportions – support (0, 1)

normal Mean differences and log odds ratios – support (-Inf, Inf)

gamma Event rates and median survival – support (0, Inf)

lognormal Hazard ratios and PK parameters – support (0, Inf)

exponential Constant hazard rates and Poisson rate priors – support (0, Inf). Conjugate with Poisson and survival data via Gamma-Poisson/Exponential updating.

weibull Non-constant hazard survival times (OS, PFS) – support (0, Inf). Posterior approximated via Normal matching.

Data types for conflict diagnostics and sensitivity

"binary" Events / sample size (x, n). Conjugate: Beta-Binomial.

"continuous" Observed mean, SD, sample size (x, sd, n). Conjugate: Normal-Normal.

"poisson" Event count / exposure person-time (x, n). Conjugate: Gamma-Poisson.

"survival" Events / total follow-up time (x, n). Conjugate: Gamma-Exponential.

References

- O'Hagan et al. (2006). *Uncertain Judgements*. Wiley.
- Box (1980). JRSS-A, 143, 383–430.
- Schmidli et al. (2014). *Biometrics*, 70, 1023–1032.
- Ibrahim & Chen (2000). *Statistical Science*, 15, 46–60.
- Spiegelhalter et al. (1994). JRSS-A, 157, 357–416.
- FDA Draft Guidance: Bayesian Methods in Clinical Trials (2026).

Author(s)

Maintainer: Ndoh Penn <ndohpenn9@gmail.com>

See Also

Useful links:

- <https://github.com/ndohpenngit/bayprior>
- Report bugs at <https://github.com/ndohpenngit/bayprior/issues>

calibrate_power_prior *Calibrate power prior weight via Bayes Factor*

Description

Selects the power prior weight $\delta \in (0, 1)$ that down-weights historical data before incorporating it into the current analysis.

Usage

```
calibrate_power_prior(
  historical_data,
  current_data,
  base_prior,
  target_bf = 3,
  delta_grid = seq(0.05, 1, by = 0.05),
  method = c("bayes_factor", "compatibility")
)
```

Arguments

historical_data	Named list: type, x, n, optionally sd.
current_data	Named list (same structure as historical_data).
base_prior	A bayprior object (usually a vague prior).
target_bf	Numeric. Target Bayes Factor. Default 3.
delta_grid	Numeric vector of δ values. Default <code>seq(0.05, 1.0, by = 0.05)</code> .
method	Character. "bayes_factor" (default) or "compatibility".

Value

A list of class bayprior_power_prior with components:

delta_opt Optimal power prior weight selected by the chosen method.

delta_grid The grid of delta values evaluated.

bf_grid Bayes Factor at each delta value.

compatibility_grid Box p-value at each delta value.

results Data frame with all diagnostic metrics across the grid.

power_prior A bayprior object updated with the optimal delta.

target_bf The target Bayes Factor supplied by the user.

method The calibration method used.

References

Ibrahim, J. G. & Chen, M.-H. (2000). Power prior distributions for regression models. *Statistical Science*, 15, 46-60.

Gravestock, I. & Held, L. (2017). Adaptive power priors with empirical Bayes for clinical trials. *Pharmaceutical Statistics*, 16, 349-360.

Examples

```
base <- elicit_beta(mean = 0.5, sd = 0.2, method = "moments",
  label = "Response rate")
calib <- calibrate_power_prior(
  historical_data = list(type = "binary", x = 12, n = 40),
  current_data   = list(type = "binary", x = 18, n = 50),
  base_prior     = base,
  target_bf      = 3
)
print(calib)
plot(calib)
```

conflict_mahalanobis *Multivariate prior-data conflict via Mahalanobis distance*

Description

Tests joint prior-data conflict across two correlated endpoints using the Mahalanobis distance. Under the null (no conflict) the squared distance follows a chi-squared distribution with degrees of freedom equal to the number of endpoints.

Usage

```
conflict_mahalanobis(  
  prior_means,  
  prior_cov,  
  obs_means,  
  obs_cov,  
  alpha = 0.05,  
  labels = NULL  
)
```

Arguments

prior_means	Numeric vector of length k. Prior means for each endpoint.
prior_cov	k x k numeric matrix. Prior covariance matrix.
obs_means	Numeric vector of length k. Observed data means.
obs_cov	k x k numeric matrix. Observed data covariance (Var/n for each diagonal; Cov/n for off-diagonal).
alpha	Numeric. Significance level for the chi-squared test. Default 0.05.
labels	Character vector of length k. Endpoint labels for output.

Details

Assumptions: This test assumes that the prior and observed summary statistics are approximately multivariate Normal. For proportion endpoints (e.g. response rates), transform to the **log-odds scale** before entering means and variances. For hazard ratios, use the **log scale**. Results may be unreliable if the Normal approximation is poor (e.g. for small samples with extreme proportions).

Current limitation: The function is designed for bivariate ($k = 2$) endpoints. While it will accept $k > 2$, the Shiny interface currently only exposes two endpoints. Support for $k \geq 3$ is a planned extension.

Distribution family: The Mahalanobis approach is distribution-agnostic at the summary statistic level – it does not require a specific prior family (Beta, Normal, etc.). Any continuous prior whose mean and covariance can be extracted is supported.

Value

A named list with components:

mahal_distance Mahalanobis distance D .

mahal_d2 Squared distance D^2 .

p_value Chi-squared p-value ($df = k$).

conflict_flag Logical. TRUE if $p_value < \alpha$.

labels Endpoint labels.

References

Mahalanobis, P. C. (1936). On the generalised distance in statistics. *Proceedings of the National Institute of Sciences of India*, 2, 49–55.

Examples

```
pm <- c(0.35, 0.60)
pcov <- matrix(c(0.010, 0.003, 0.003, 0.015), 2, 2)
om <- c(0.55, 0.58)
ocov <- matrix(c(2e-4, 4e-5, 4e-5, 2e-4), 2, 2)
conflict_mahalanobis(pm, pcov, om, ocov, labels = c("Response rate", "OS rate"))
```

elicit_beta

Elicit a Beta prior via quantile matching or moment matching

Description

Fits a Beta(α , β) distribution to expert-specified quantiles or moments. Implements the structured elicitation framework recommended in the SHELF methodology and FDA guidance on Bayesian clinical trials.

Usage

```
elicit_beta(
  quantiles = NULL,
  mean = NULL,
  sd = NULL,
  method = c("quantile", "moments"),
  expert_id = "Expert_1",
  label = "Unknown quantity",
  tol = 1e-06
)
```

Arguments

quantiles	Named numeric vector of quantile specifications, e.g. <code>c("0.05" = 0.1, "0.50" = 0.3, "0.95" = 0.6)</code> . At least two quantiles required.
mean	Optional numeric. Expert-specified mean for moment matching.
sd	Optional numeric. Expert-specified SD for moment matching.
method	Character. One of "quantile" (default) or "moments".
expert_id	Character. Identifier for this expert's elicitation.
label	Character. Description of the quantity being elicited.
tol	Numeric. Optimisation tolerance. Default 1e-6.

Value

An object of class bayprior with components:

```

dist "beta"
params Named list with alpha and beta
method Elicitation method used
expert_id Expert identifier
label Quantity label
fit_summary Summary statistics of fitted prior

```

Examples

```

# Expert believes response rate is ~30%, with 90% CI of [10%, 60%]
prior <- elicit_beta(
  quantiles = c("0.05" = 0.10, "0.50" = 0.30, "0.95" = 0.60),
  expert_id = "Expert_1",
  label     = "Response rate (treatment arm)"
)
print(prior)
plot(prior)

# Moment-based elicitation
prior_mom <- elicit_beta(
  mean = 0.30, sd = 0.12,
  method = "moments",
  label = "Response rate"
)

```

elicit_exponential *Elicit an Exponential prior via moments, rate, or quantile matching*

Description

Fits an Exponential(rate) prior from expert-specified moments or quantiles. Suitable for constant-hazard survival models and Poisson rate priors. The Exponential distribution is the conjugate prior likelihood for a Gamma prior on the rate parameter.

Usage

```
elicit_exponential(
  rate = NULL,
  mean = NULL,
  quantiles = NULL,
  method = c("moments", "rate", "quantile"),
  expert_id = "Expert",
  label = "Quantity"
)
```

Arguments

rate	Numeric > 0. Rate parameter (= 1 / mean). Used when method = "rate".
mean	Numeric > 0. Prior mean (= 1 / rate). Used when method = "moments".
quantiles	Named numeric vector with at least one interior quantile. Used when method = "quantile".
method	Character. One of "moments", "rate", or "quantile". Default "moments".
expert_id	Character. Identifier for the eliciting expert.
label	Character. Human-readable label for the quantity.

Details

The Exponential distribution has a single parameter $\lambda > 0$ (the rate). Its mean is $1/\lambda$ and its SD equals its mean.

Typical use cases:

Hazard rates OS/PFS hazard in oncology trials

Event rates Adverse event rates (events per person-time)

Poisson rate priors Conjugate prior for Poisson count data

Value

A bayprior object with dist = "exponential".

Examples

```
# Mean survival 20 months => hazard rate 1/20 = 0.05
p <- elicit_exponential(mean = 0.05, method = "moments",
                        label = "Hazard rate",
                        expert_id = "Expert_1")

print(p)
plot(p)

# Direct rate specification
p2 <- elicit_exponential(rate = 0.10, method = "rate",
                         label = "AE rate per person-year")

# Quantile matching
p3 <- elicit_exponential(
  quantiles = c("0.25" = 0.02, "0.50" = 0.05, "0.75" = 0.10),
  method = "quantile",
  label = "Hazard rate"
)
```

elicit_gamma

Elicit a Gamma prior via quantile matching or moment matching

Description

Elicit a Gamma prior via quantile matching or moment matching

Usage

```
elicit_gamma(
  quantiles = NULL,
  mean = NULL,
  sd = NULL,
  method = c("quantile", "moments"),
  expert_id = "Expert_1",
  label = "Unknown quantity",
  tol = 1e-06
)
```

Arguments

quantiles	Named numeric vector of quantiles. Values must be positive.
mean	Optional numeric. Expert mean.
sd	Optional numeric. Expert SD.
method	Character. "quantile" or "moments".
expert_id	Character. Expert identifier.
label	Character. Quantity description.
tol	Numeric. Optimisation tolerance.

Value

An object of class bayprior.

Examples

```
prior <- elicit_gamma(
  mean = 5, sd = 2,
  method = "moments",
  label = "Median OS (months)"
)
```

elicit_lognormal	<i>Elicit a Log-Normal prior via quantile matching or moment matching</i>
------------------	---

Description

Fits a Log-Normal distribution to expert-specified quantiles or moments. Appropriate for positive-valued quantities such as hazard ratios, fold changes, median survival times, or PK parameters.

Usage

```
elicit_lognormal(
  quantiles = NULL,
  mean = NULL,
  sd = NULL,
  method = c("quantile", "moments"),
  expert_id = "Expert_1",
  label = "Unknown quantity",
  tol = 1e-06
)
```

Arguments

quantiles	Named numeric vector. Values must be strictly positive. E.g. $c("0.05" = 0.5, "0.50" = 2.0, "0.95" = 8.0)$.
mean	Optional numeric. Expert mean on the original scale.
sd	Optional numeric. Expert SD on the original scale.
method	Character. "quantile" (default) or "moments".
expert_id	Character. Expert identifier.
label	Character. Quantity description.
tol	Numeric. Optimisation tolerance.

Value

An object of class bayprior with `dist = "lognormal"`.

Examples

```
prior <- elicit_lognormal(
  quantiles = c("0.05" = 0.40, "0.50" = 0.70, "0.95" = 1.20),
  label     = "Hazard ratio (treatment vs control)"
)
print(prior)
```

elicit_mixture

Elicit a mixture prior

Description

Constructs a finite mixture prior from a list of component bayprior objects (e.g., from `elicit_beta`, `elicit_normal`). Mixing weights can be specified or estimated via linear pooling.

Usage

```
elicit_mixture(components, weights = NULL, label = "Mixture prior")
```

Arguments

<code>components</code>	List of bayprior objects.
<code>weights</code>	Numeric vector of mixing weights (must sum to 1). If <code>NULL</code> , equal weights are assigned.
<code>label</code>	Character. Label for the mixture prior.

Details

Same-family mixtures (e.g. Beta + Beta, Normal + Normal) have mixture means and SDs computed in closed form. **Cross-family mixtures** (e.g. Beta + Normal) also have closed-form mixture means and SDs, but the mixture *density* must be computed numerically by evaluating and summing the component densities on a grid. A warning is issued in this case:

```
"Components have different distribution families.
Mixture densities computed numerically."
```

This warning is informative, not an error. The numerical approximation is accurate in the body of the distribution but may be less reliable at the tails, particularly for components with very different supports (e.g. a Beta defined on (0, 1) mixed with a Normal defined on (-Inf, Inf)). Use `suppressWarnings()` only when you have verified the mixture is appropriate for your use case.

Value

A bayprior object with `dist = "mixture"`.

Examples

```
p1 <- elicit_beta(mean = 0.2, sd = 0.08, method = "moments", expert_id = "E1")
p2 <- elicit_beta(mean = 0.4, sd = 0.10, method = "moments", expert_id = "E2")
mix <- elicit_mixture(list(p1, p2), weights = c(0.5, 0.5),
  label = "Pooled prior")
```

elicit_normal

Elicit a Normal prior via quantile matching or moment matching

Description

Elicit a Normal prior via quantile matching or moment matching

Usage

```
elicit_normal(
  quantiles = NULL,
  mean = NULL,
  sd = NULL,
  method = c("quantile", "moments"),
  expert_id = "Expert_1",
  label = "Unknown quantity",
  tol = 1e-06
)
```

Arguments

quantiles	Named numeric vector. E.g. <code>c("0.025" = -0.5, "0.50" = 0.2, "0.975" = 0.9)</code> .
mean	Optional numeric. Expert mean for moment matching.
sd	Optional numeric. Expert SD for moment matching.
method	Character. "quantile" or "moments".
expert_id	Character. Expert identifier.
label	Character. Quantity description.
tol	Numeric. Optimisation tolerance.

Value

An object of class bayprior.

Examples

```
prior <- elicit_normal(
  quantiles = c("0.025" = -0.5, "0.50" = 0.2, "0.975" = 0.9),
  label = "Log odds ratio"
)
```

elicit_roulette	<i>Roulette-method elicitation (chip-allocation)</i>
-----------------	--

Description

Implements the SHELF roulette method: the expert allocates a fixed number of "chips" across a set of pre-defined bins representing the range of the quantity. The resulting histogram is fitted to a parametric distribution.

Usage

```
elicit_roulette(  
  chips,  
  breaks,  
  family = c("beta", "normal", "gamma", "lognormal"),  
  expert_id = "Expert_1",  
  label = "Unknown quantity"  
)
```

Arguments

chips	Integer vector. Number of chips in each bin (left-to-right).
breaks	Numeric vector of length $\text{length}(\text{chips}) + 1$ defining the bin edges.
family	Character. Distribution to fit. One of "beta", "normal", "gamma", "lognormal".
expert_id	Character. Expert identifier.
label	Character. Quantity description.

Details

In the Shiny app (Prior elicitation tab) the roulette grid is rendered interactively. This function provides the **fitting back-end** that can also be called programmatically when chips are known.

Chips are converted to relative frequencies, and bin midpoints are used as representative values. The chosen family is then fitted by minimising the weighted sum of squared CDF differences (a histogram-matching approach).

Value

A bayprior object fitted to the chip histogram.

References

Oakley, J. E. & O'Hagan, A. (2010). SHELF: the Sheffield Elicitation Framework. University of Sheffield.

Examples

```
# Expert places 0, 2, 5, 8, 5, 2, 1 chips across bins [0,.1,.2,...,.7]
prior <- elicit_roulette(
  chips = c(0L, 2L, 5L, 8L, 5L, 2L, 1L),
  breaks = seq(0, 0.7, by = 0.1),
  family = "beta",
  label = "Response rate"
)
print(prior)
```

elicit_weibull	<i>Elicit a Weibull prior via moments, direct parameters, or quantile matching</i>
----------------	--

Description

Fits a Weibull(shape, scale) prior from expert-specified moments or quantiles. The Weibull distribution generalises the Exponential and is widely used for survival analysis with non-constant hazard.

Usage

```
elicit_weibull(
  shape = NULL,
  scale = NULL,
  mean = NULL,
  sd = NULL,
  quantiles = NULL,
  method = c("moments", "params", "quantile"),
  expert_id = "Expert",
  label = "Quantity"
)
```

Arguments

- shape Numeric > 0. Shape parameter k . Used when method = "params".
- scale Numeric > 0. Scale parameter λ . Used when method = "params".
- mean Numeric > 0. Prior mean. Used with sd when method = "moments".
- sd Numeric > 0. Prior SD. Used with mean when method = "moments".
- quantiles Named numeric vector with at least two interior quantiles. Used when method = "quantile".
- method Character. One of "moments", "params", or "quantile". Default "moments".
- expert_id Character. Identifier for the eliciting expert.
- label Character. Human-readable label for the quantity.

Details

Parameterised as in R's `stats::dweibull`: shape k and scale λ , with mean $\lambda\Gamma(1 + 1/k)$ and variance $\lambda^2[\Gamma(1 + 2/k) - \Gamma(1 + 1/k)^2]$.

Shape parameter interpretation:

- $k < 1$: decreasing hazard (e.g. early mortality selecting out)
- $k = 1$: constant hazard (reduces to Exponential)
- $k > 1$: increasing hazard (e.g. ageing, post-surgical)

Value

A bayprior object with `dist = "weibull"`.

Examples

```
# Moment matching: mean 20 months, SD 10 months
p <- elicit_weibull(mean = 20, sd = 10, method = "moments",
                   label = "Survival time (months)",
                   expert_id = "Expert_1")

print(p)
plot(p)

# Direct parameters (shape = 2 = increasing hazard)
p2 <- elicit_weibull(shape = 2, scale = 20, method = "params",
                    label = "PFS (months)")

# Quantile matching (at least 2 required)
p3 <- elicit_weibull(
  quantiles = c("0.10" = 5, "0.50" = 18, "0.90" = 40),
  method = "quantile",
  label = "OS (months)"
)
```

```
plot.bayprior_power_prior
```

Plot calibration curve for power prior weight selection

Description

Plot calibration curve for power prior weight selection

Usage

```
## S3 method for class 'bayprior_power_prior'
plot(x, ...)
```

Arguments

x A bayprior_power_prior object.
 ... Ignored.

Value

A ggplot object, or a list of two ggplots if 'patchwork' is not installed.

plot_prior_likelihood *Plot prior, likelihood, and posterior density overlays*

Description

Plot prior, likelihood, and posterior density overlays

Usage

```
plot_prior_likelihood(
  prior,
  data_summary,
  show_posterior = TRUE,
  show_conflict = TRUE,
  n_grid = 500,
  title = NULL
)
```

Arguments

prior A bayprior object from any elicit_*() function.
 data_summary Named list with n, x, optionally sd and type.
 show_posterior Logical. Default TRUE.
 show_conflict Logical. Default TRUE.
 n_grid Integer. Default 500.
 title Character. Plot title.

Value

A ggplot object.

Examples

```
prior <- elicit_beta(mean = 0.30, sd = 0.10, method = "moments",
  label = "Response rate")
plot_prior_likelihood(prior, list(n = 40, x = 20, type = "binary"))
```

plot_sensitivity *Plot sensitivity analysis results*

Description

Plot sensitivity analysis results

Usage

```
plot_sensitivity(sensitivity, target = NULL, highlight_reference = TRUE)
```

Arguments

sensitivity A bayprior_sensitivity object.
target Character. Which target quantity to plot.
highlight_reference
 Logical. Default TRUE.

Value

A ggplot object.

plot_tornado *Tornado plot of prior influence on posterior quantities*

Description

Tornado plot of prior influence on posterior quantities

Usage

```
plot_tornado(sensitivity, title = "Prior influence on posterior estimates")
```

Arguments

sensitivity A bayprior_sensitivity object.
title Character. Plot title.

Value

A ggplot object.

print.bayprior	<i>Print method for bayprior objects</i>
----------------	--

Description

Print method for bayprior objects

Usage

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

Arguments

x	A bayprior object.
...	Ignored.

Value

Invisibly returns the input bayprior object. Called for its side effect of printing a formatted summary of the prior distribution including family, parameters, mean, SD, and 95\ credible interval.

print.bayprior_conflict	<i>Print method for bayprior_conflict objects</i>
-------------------------	---

Description

Print method for bayprior_conflict objects

Usage

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

Arguments

x	A bayprior_conflict object.
...	Ignored.

Value

Invisibly returns the input bayprior_conflict object. Called for its side effect of printing conflict diagnostic statistics including Box p-value, surprise index, information divergence, Bhattacharyya overlap, and colour-coded conflict severity.

```
print.bayprior_conflict_mv
```

Print method for multivariate conflict objects

Description

Print method for multivariate conflict objects

Usage

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

Arguments

x	A bayprior_conflict_mv object.
...	Ignored.

Value

Invisibly returns the input bayprior_conflict_mv object. Called for its side effect of printing a formatted summary of the multivariate Mahalanobis conflict check, including the Mahalanobis distance, chi-squared p-value, conflict flag, per-parameter marginal z-scores, and an interpretation string.

```
print.bayprior_power_prior
```

Print method for bayprior_power_prior objects

Description

Print method for bayprior_power_prior objects

Usage

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

Arguments

x	A bayprior_power_prior object.
...	Ignored.

Value

Invisibly returns the input bayprior_power_prior object. Called for its side effect of printing a formatted summary of the power prior calibration, including the calibration method, target Bayes Factor, optimal delta weight, and the mean and SD of the resulting power prior.

prior_conflict	<i>Compute prior-data conflict diagnostics</i>
----------------	--

Description

Evaluates conflict between a specified prior and observed data using multiple complementary diagnostics: Box's (1980) predictive p-value, the surprise index (standardised distance), Kullback-Leibler divergence, and the Bhattacharyya overlap coefficient between the prior and the (normalised) likelihood.

Usage

```
prior_conflict(prior, data_summary, alpha = 0.05)
```

Arguments

prior	A bayprior object.
data_summary	Named list describing the observed data: type "binary", "continuous", "poisson", or "survival". x Number of events (binary / poisson / survival) or observed mean (continuous). n Sample size (binary / continuous), total exposure (poisson: person-time), or total follow-up time (survival). sd Observed standard deviation (continuous only).
alpha	Numeric. Significance level for the Box p-value flag. Default 0.05.

Value

An object of class bayprior_conflict containing:

box_pvalue	Box's prior predictive p-value.
surprise_index	Standardised distance between prior mean and observed data.
kl_prior_likelihood	KL divergence from prior to likelihood.
overlap	Bhattacharyya overlap coefficient in [0, 1].
conflict_severity	One of "none", "mild", "severe".
conflict_flag	Logical; TRUE if box_pvalue < alpha.
recommendation	Plain-language guidance string.
data_summary	The data summary passed in.
prior	The input prior.

References

Box, G. E. P. (1980). Sampling and Bayes' inference in scientific modelling and robustness. *Journal of the Royal Statistical Society A*, 143, 383-430.

Examples

```
prior <- elicit_beta(mean = 0.30, sd = 0.10, method = "moments",
                    label = "Response rate")
cd <- prior_conflict(prior, list(type = "binary", x = 18, n = 40))
print(cd)
```

prior_report

Generate a Prior Justification Report

Description

Renders an HTML, PDF, or Word (.docx) report using `quarto::quarto_render()`.

Usage

```
prior_report(
  prior,
  conflict = NULL,
  sensitivity = NULL,
  robust_prior = NULL,
  sceptical_prior = NULL,
  power_prior = NULL,
  output_format = c("html", "pdf", "docx"),
  output_file = NULL,
  trial_name = "Clinical Trial",
  sponsor = "Sponsor",
  date = as.character(Sys.Date()),
  author = "Biostatistics",
  notes = "",
  prior_plot = NULL,
  overlay_plot = NULL,
  tornado_plot = NULL,
  heatmap_plot = NULL,
  robust_plot = NULL,
  sceptical_plot = NULL,
  power_plot = NULL,
  open_after = interactive()
)
```

Arguments

prior	A bayprior object.
conflict	Optional bayprior_conflict from prior_conflict().
sensitivity	Optional bayprior_sensitivity from sensitivity_grid().
robust_prior	Optional output of robust_prior().
sceptical_prior	Optional output of sceptical_prior().
power_prior	Optional output of calibrate_power_prior().
output_format	"html" (default), "pdf", or "docx".
output_file	Output path without extension.
trial_name	Trial / protocol identifier.
sponsor	Sponsor name.
date	Report date string. Default Sys.Date().
author	Responsible statistician.
notes	Optional narrative text.
prior_plot	Optional pre-captured ggplot from plot(prior).
overlay_plot	Optional pre-captured ggplot from plot_prior_likelihood().
tornado_plot	Optional pre-captured ggplot from plot_tornado().
heatmap_plot	Optional pre-captured ggplot from plot_sensitivity().
robust_plot	Optional pre-captured ggplot of the robust prior.
sceptical_plot	Optional pre-captured ggplot of the sceptical prior.
power_plot	Optional pre-captured ggplot of the power prior.
open_after	Open after rendering. Default TRUE interactively.

Value

Path to the rendered report, invisibly.

robust_prior	<i>Construct a robust (heavy-tailed mixture) prior</i>
--------------	--

Description

Builds a robust prior by mixing an informative component with a vague (diffuse) component, following the RBeST/MAP robust mixture approach (Schmidli et al., 2014). This protects against prior misspecification by ensuring the posterior is not dominated by a conflicting informative prior.

Usage

```
robust_prior(
  informative,
  vague_weight = 0.2,
  vague_sd = NULL,
  label = "Robust mixture prior"
)
```

Arguments

informative	A bayprior object representing the informative component (e.g. an elicited or historical prior).
vague_weight	Numeric in (0, 1). Weight assigned to the vague (diffuse) component. Default 0.20 (80% informative, 20% vague).
vague_sd	Numeric. SD of the vague Normal component (on the natural scale). If NULL, defaults to 10x the informative prior's SD.
label	Character. Label for the robust prior.

Details

The vague component is always a **Normal distribution** centred at the informative prior's mean with $SD = vague_sd$ (default: 10x the informative SD). When the informative prior is itself Normal, both components share the same family and the mixture density is computed analytically. For any other informative prior family (Beta, Gamma, Log-Normal, Exponential, Weibull), the components have *different distribution families*, and the mixture density is **computed numerically**. A warning is issued in this case.

Value

A bayprior object with `dist = "mixture"` and `prior_type = "robust"`.

References

Schmidli, H. et al. (2014). Robust meta-analytic-predictive priors in clinical trials with historical control information. *Biometrics*, 70, 1023-1032.

Examples

```
informative <- elicit_normal(mean = 0.30, sd = 0.10,
                             method = "moments", label = "Response rate")
robust <- robust_prior(informative, vague_weight = 0.20)
plot(robust)
```

`run_app`*Run the bayprior Shiny Application*

Description

Launches the golem-structured shinydashboard application.

Usage

```
run_app(  
  onStart = NULL,  
  options = list(),  
  enableBookmarking = NULL,  
  uiPattern = "/",  
  ...  
)
```

Arguments

<code>onStart</code>	A function called before the app runs.
<code>options</code>	List of options passed to <code>shiny::shinyApp()</code> .
<code>enableBookmarking</code>	Passed to <code>shiny::shinyApp()</code> .
<code>uiPattern</code>	Passed to <code>shiny::shinyApp()</code> .
<code>...</code>	Additional arguments passed to <code>golem::with_golem_options()</code> .

Value

A `shiny.appobj`, invisibly.

`sceptical_prior`*Construct a sceptical (penalised-enthusiasm) prior*

Description

Generates a sceptical prior that places most mass at or near the null value of the treatment effect, representing a conservative stance for regulatory submissions. Implements the Spiegelhalter-Freedman sceptical prior approach and the FDA-recommended "enthusiastic vs sceptical" prior sensitivity pair.

Usage

```
sceptical_prior(
  null_value = 0,
  family = c("normal", "beta", "lognormal"),
  strength = c("moderate", "weak", "strong"),
  label = "Treatment effect",
  expert_id = "Sceptic"
)
```

Arguments

null_value	Numeric. The null treatment effect (e.g. 0 for a mean difference, 1 for a hazard ratio, 0.5 for a response-rate difference). For family = "beta", must be strictly in (0, 1).
family	Character. Distribution family. One of "normal", "beta", "lognormal".
strength	Character. How concentrated the prior is around the null: "weak", "moderate" (default), or "strong".
label	Character. Description of the quantity.
expert_id	Character. Identifier for provenance.

Details

For a Normal family, the sceptical prior is centred at null_value with SD calibrated to the strength argument:

- weak: SD = 1.0 (vague half-normal)
- moderate: SD = 0.5 (2-SD departure from null has ~5% prior probability)
- strong: SD = 0.25 (very concentrated at null)

For family = "beta", null_value must be in (0, 1).

Value

A bayprior object tagged with prior_type = "sceptical".

References

Spiegelhalter, D. J., Freedman, L. S. & Parmar, M. K. B. (1994). Bayesian approaches to randomized trials. *JRSS-A*, 157, 357-416.

Examples

```
sc <- sceptical_prior(null_value = 0, family = "normal",
  strength = "moderate", label = "Mean difference")
print(sc)
plot(sc)

# Beta sceptical prior centred at a null response rate
```

```
sc_b <- sceptical_prior(null_value = 0.20, family = "beta",
                        strength = "moderate", label = "Response rate")
plot(sc_b)
```

sensitivity_cri

Sensitivity of posterior CrI to prior hyperparameters

Description

A focused sensitivity analysis that tracks the width and location of the posterior credible interval (CrI) across a one- or two-dimensional hyperparameter grid. This directly answers the regulatory question: "Does the CrI change materially under plausible alternative priors?"

Evaluates how the posterior credible interval width and bounds change as prior hyperparameters vary over a specified grid. This is the preferred function for demonstrating that key regulatory conclusions (e.g., whether the CrI excludes a null value) are robust to prior choice.

Usage

```
sensitivity_cri(
  prior,
  data_summary,
  param_grid,
  cri_level = 0.95,
  threshold = NULL
)
```

```
sensitivity_cri(
  prior,
  data_summary,
  param_grid,
  cri_level = 0.95,
  threshold = NULL
)
```

Arguments

prior	A bayprior object (the reference prior).
data_summary	Named list as for prior_conflict .
param_grid	Named list of numeric vectors, one per hyperparameter.
cri_level	Numeric in (0, 1). Credible interval level. Default 0.95.
threshold	Optional numeric. Computes $\Pr(\theta > \text{threshold})$ at each grid point if supplied.

Value

A bayprior_sensitivity object (same class as sensitivity_grid) with additional columns cri_lower, cri_upper, and cri_width in the result grid.

An object of class bayprior_sensitivity whose grid contains columns cri_lower, cri_upper, and cri_width, plus optionally posterior_mean, posterior_sd, and prob_efficacy.

Examples

```
prior <- elicit_beta(mean = 0.30, sd = 0.10, method = "moments")
cri_sa <- sensitivity_cri(
  prior,
  data_summary = list(type = "binary", x = 14, n = 40),
  param_grid   = list(alpha = seq(1, 8, 0.5), beta = seq(2, 20, 1)),
  cri_level    = 0.95,
  threshold    = 0.30
)
plot_sensitivity(cri_sa, target = "cri_width")

prior <- elicit_beta(mean = 0.30, sd = 0.10, method = "moments",
  label = "Response rate")
cri_sa <- sensitivity_cri(
  prior,
  data_summary = list(type = "binary", x = 14, n = 40),
  param_grid   = list(alpha = seq(1, 8, 0.5), beta = seq(2, 20, 1)),
  cri_level    = 0.95
)
plot_sensitivity(cri_sa, target = "cri_width")
```

sensitivity_grid

Sensitivity grid over prior hyperparameters

Description

Evaluates how posterior inferences change as prior hyperparameters vary over a specified grid. This is the core function for demonstrating robustness of trial conclusions to prior choice.

Usage

```
sensitivity_grid(
  prior,
  data_summary,
  param_grid,
  target = c("posterior_mean", "posterior_sd", "prob_efficacy"),
  threshold = 0.3
)
```

Arguments

prior	A bayprior object (the reference prior).
data_summary	Named list as for <code>prior_conflict</code> .
param_grid	Named list of numeric vectors, one per hyperparameter to vary. Names must match hyperparameter names in <code>prior\$params</code> . Example: <code>list(alpha = seq(1, 8, 0.5), beta = seq(2, 20, 1))</code> .
target	Character vector. Which posterior quantities to compute. Any of "posterior_mean", "posterior_sd", "prob_efficacy".
threshold	Numeric. Efficacy threshold used in $\Pr(\theta > \text{threshold})$. Default 0.30.

Value

An object of class `bayprior_sensitivity`.

Examples

```
prior <- elicit_beta(mean = 0.30, sd = 0.10, method = "moments",
                    label = "Response rate")
sa <- sensitivity_grid(
  prior,
  data_summary = list(type = "binary", x = 14, n = 40),
  param_grid   = list(alpha = seq(1, 8, 0.5), beta = seq(2, 20, 1))
)
plot_tornado(sa)
plot_sensitivity(sa, target = "posterior_mean")
```

summary.bayprior

Summary method for bayprior objects

Description

Summary method for bayprior objects

Usage

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

Arguments

object	A bayprior object.
...	Ignored.

Value

A list with summary statistics (invisibly).

Index

aggregate_experts, 3, 5
as_prior, 4

bayprior, 5
bayprior-package (bayprior), 5

calibrate_power_prior, 5, 6
conflict_mahalanobis, 5, 8

elicit_beta, 5, 9
elicit_exponential, 5, 11
elicit_gamma, 5, 12
elicit_lognormal, 5, 13
elicit_mixture, 5, 14
elicit_normal, 5, 15
elicit_roulette, 5, 16
elicit_weibull, 5, 17

plot.bayprior_power_prior, 18
plot_prior_likelihood, 19
plot_sensitivity, 20
plot_tornado, 20
print.bayprior, 21
print.bayprior_conflict, 21
print.bayprior_conflict_mv, 22
print.bayprior_power_prior, 22
prior_conflict, 5, 23, 29, 31
prior_report, 5, 24

robust_prior, 5, 25
run_app, 5, 27

sceptical_prior, 5, 27
sensitivity_cri, 5, 29
sensitivity_grid, 5, 30
summary.bayprior, 31