# Package 'bayespmtools'

June 5, 2026

**Title** Bayesian Sample Size and Precision Considerations for Risk Prediction Models

**Version** 0.0.2

**Maintainer** Mohsen Sadatsafavi <mohsen.sadatsafavi@ubc.ca>

**Description** Performs Bayesian sample size, precision, and value-of-information analysis for external validation of existing multi-variable prediction models using the approach proposed by Sadatsafavi and colleagues (2026) <doi:10.1002/sim.70389>.

**URL** https://github.com/resplab/bayespmtools

**BugReports** https://github.com/resplab/bayespmtools/issues

**License** GPL-3

**Encoding** UTF-8

**Depends** R (>= 3.5.0)

**Imports** fastLogisticRegressionWrap, logitnorm, mc2d, mcmapper, pROC, cobs, OOR, quantreg

**LazyData** true

**Suggests** knitr, rmarkdown, ggplot2, testthat (>= 3.0.0)

**VignetteBuilder** knitr

**Config/testthat/edition** 3

**Config/roxygen2/version** 8.0.0

**NeedsCompilation** no

**Author** Mohsen Sadatsafavi [aut, cre] (ORCID: <https://orcid.org/0000-0002-0419-7862>), Anna Luo [ctb]

**Repository** CRAN

**Date/Publication** 2026-06-05 06:40:02 UTC

# Contents



---

`bpm_valprec` *Bayesian Precision / VoI Calculator*

---

## Description

Bayesian precision and value-of-information calculator for external validation studies of risk prediction models at fixed sample sizes.

## Usage

```
bpm_valprec(
  N,
  evidence,
  targets,
  n_sim = NULL,
  method = "sample",
  threshold = NULL,
  dist_type = "logitnorm",
  impute_cor = TRUE,
  ex_args = NULL
)
```

## Arguments

| | |
|---|---|
| N | Numeric vector of sample sizes to evaluate. |
| evidence | A named list containing prior evidence components for model performance parameters (e.g., prevalence, discrimination, calibration). Alternatively, evidence may be a data frame of pre-posterior draws (element $sample) returned by a previous call to this function or to bpm_valprec(), in which case those draws are used directly. |
| targets | A named list of targets to compute.<br>**eciw.metric** Logical; compute expected CI width.<br>**qciw.metric** Numeric scalar in (0,1); CI width quantile.<br>**oa.nb** Logical; compute optimality assurance for net benefit.<br>**voi.nb** Logical; compute EVSI and EVSI/EVPI. |
| n_sim | #' Number of Monte Carlo simulations used to generate the pre-posterior distribution. If evidence is a data frame from previous calls to relevant functions, n_sim will automatically be set to the number of rows of the data frame. |
| method | Method to compute CI widths. One of "sample" or "2s". |
| threshold | Decision threshold for net benefit calculations. Required if oa.nb or voi.nb are requested. |
| dist_type | Distribution for calibrated risks. Default is "logitnorm". |
| impute_cor | Logical; whether to induce correlation between parameters. |
| ex_args | Optional list of extra arguments. May include f_progress, a custom progress function. |

## Value

A list with elements:

**results** Matrix of requested metrics by sample size.

**sample** Monte Carlo sample used for computations.

**evidence** Processed evidence object.

**targets** Targets as supplied by the user.

**ciws** Simulated CI widths for requested metrics.

## Examples

```
evidence <- list(
  prev ~ beta(116, 155),           # Outcome prevalence
  cstat ~ beta(3628, 1139),        # C-statistic
  cal_mean ~ norm(-0.009, 0.125),  # Mean calibration error
  cal_slp ~ norm(0.995, 0.024)     # Calibration slope
)

res <- bpm_valprec(
  N = c(1000, 1500),
  evidence = evidence,
```

```
  targets = list(eciw.cstat = TRUE, qciw.cal_slp=0.9, voi.nb=0.8),
  threshold=0.2,
 n_sim = 100        # faster and safer on CRAN. Please increase this value for real-world use.
)

print(res$results)
```

---

bpm_valsamp *Bayesian Sample Size Calculator for External Validation*

---

## Description

Bayesian sample size calculation for external validation studies of clinical risk prediction models. The function evaluates sample sizes required to meet precision-, assurance-, or decision-based targets using pre-posterior simulation.

## Usage

```
bpm_valsamp(
  evidence,
  targets,
  n_sim = NULL,
  method = "sample",
  threshold = NULL,
  dist_type = "logitnorm",
  impute_cor = TRUE,
  ex_args = NULL
)
```

## Arguments

| | |
|---|---|
| evidence | A named list containing prior evidence components for model performance parameters (e.g., prevalence, discrimination, calibration). Alternatively, evidence may be a data frame of pre-posterior draws (element $sample) returned by a previous call to this function or to bpm_valprec(), in which case those draws are used directly. |
| targets | A named list specifying sample size targets.<br>Supported targets include:<br>• Precision-based targets using expected 95\ (prefix eciw).<br>• Assurance-based targets specifying the probability that the 95\ (prefix qciw).<br>• Net benefit targets, including optimality assurance (oa.nb) and value-of-information ratios (voi.nb = EVSI / EVPI).<br>For example, eciw.cstat = 0.1 targets an expected interval width of 0.1 for the c-statistic, while qciw.cal_slp = c(0.90, 0.22) targets a 90 percent assurance that the calibration slope interval width does not exceed 0.22. Finally, |

| | |
|---|---|
| | `oa.nb = 0.80` targets a sample size that would correspond to 80 percent assurance that the strategy with the highest NB in the sample will be the strategy with the highest NB in the population. |
| `n_sim` | Number of Monte Carlo simulations used to generate the pre-posterior distribution. If evidence is a data frame from previous calls to relevant functions, n_sim will automatically be set to the number of rows of the data frame. |
| `method` | Method used to compute the pre-posterior distribution of 95\ One of `"sample"` (simulation-based) or `"2s"` (two-stage approximation). Default is `"sample"`. |
| `threshold` | Risk threshold used for decision-analytic quantities and net benefit calculations. Required if `oa.nb` or `voi.nb` targets are specified. |
| `dist_type` | Distribution assumed for calibrated risks. Default is `"logitnorm"`. |
| `impute_cor` | Logical indicating whether correlation between performance measures should be induced when simulating from marginal evidence distributions. Default is `TRUE`. |
| `ex_args` | Optional list of additional arguments passed to internal simulation or root-finding routines (experimental feature). |

## Value

A list with the following components:

- `results`: Estimated sample sizes required to meet each target.
- `sample`: Data frame of pre-posterior simulation draws.
- `evidence`: Processed evidence object used in the analysis.
- `trace`: Trace output from the stochastic root-finding algorithm.
- `targets`: The targets argument supplied to the function.

## Examples

```
evidence <- list(
  prev ~ beta(116, 155),           # Outcome prevalence
  cstat ~ beta(3628, 1139),        # C-statistic
  cal_mean ~ norm(-0.009, 0.125),  # Mean calibration error
  cal_slp ~ norm(0.995, 0.024)     # Calibration slope
)

targets <- list(
  eciw.cstat = 0.1,
  qciw.cstat = c(0.9, 0.1),
  oa.nb      = 0.8
)

samp <- bpm_valsamp(
  evidence  = evidence,
  targets   = targets,
  n_sim     = 1000,
  threshold = 0.2
)
```

```
samp$results
```

---

`calc_ciw_2s` *Calculates Pre-Posterior Distribution of 95% CI Widths Using Two-step Method*

---

## Description

Calculates pre-posterior distribution of 95% CI widths using two-step method.

## Usage

```
calc_ciw_2s(N, parms)
```

## Arguments

| | |
|---|---|
| `N` | A vector of sample sizes |
| `parms` | Parameters for the distribution containing: cal_int: calibration intercept cal_slp: calibration slope prev: prevalence dist_type: distribution type cstat: c-statistic dist_type: one of ("logitnorm", "beta, "probitnorm") dist_parm1: first parameter of the distribution dist_parm2: second parameter of the distribution |

## Value

List of length N, of vectors containing 95% confidence interval width for each of: cstat: c-statistic cal_oe: observed to expected ratio cal_mean: mean calibration cal_int: calibration intercept cal_slp: calibration slope

---

`calc_ciw_mc` *#'Calculates Pre-Posterior Distribution of 95% CI Widths Based on Given Method*

---

## Description

Calculates pre-posterior distribution of 95% CI widths based on given method

## Usage

```
calc_ciw_mc(N, parms_sample, method)
```

### Arguments

| | |
|---|---|
| N | A vector of sample sizes |
| parms_sample | Matrix of parameters for the distribution each row with appropriate parameters: cstat: c-statistic prev: prevalence dist_type: distribution type dist_parm1: first parameter of distribution dist_parm2: second parameter of distribution cal_int: calibration intercept cal_slp: calibration slope |
| method | Method to calculate 95% confident interval width, one of sample, 2s |

### Value

List of matrices each with dimension (number of rows in parms_sample x length N) containing 95% confidence interval width for each of: cstat: c-statistic cal_oe: observed to expected ratio cal_mean: mean calibration cal_int: calibration intercept cal_slp: calibration slope

---

## calc_ciw_sample *Calculates Pre-Posterior Distribution of 95% CI Widths Using Sampling-based Simulation*

---

### Description

Calculates pre-posterior distribution of 95% CI widths using sampling-based simulation

### Usage

```
calc_ciw_sample(N, parms)
```

### Arguments

| | |
|---|---|
| N | A vector of sample sizes |
| parms | Parameters for the distribution containing: prev: prevalence dist_type: distribution type dist_parm1: first parameter of distribution dist_parm2: second parameter of distribution cal_int: calibration intercept cal_slp: calibration slope |

### Value

List of length N, of vectors containing 95% confidence interval width for each of: cstat: c-statistic cal_oe: observed to expected ratio cal_mean: mean calibration cal_int: calibration intercept cal_slp: calibration slope

## calc_cstat *Calculates the C-statistic of Model*

### Description

Calculates the c-statistic given the model type and parameters.

### Usage

```
calc_cstat(type, parms, m = NULL)
```

### Arguments

| | |
|---|---|
| type | A character string; one of c("beta", "logitnorm", "probitnorm") indicating the model type. |
| parms | A numeric vector containing parameters relevant to the model. |
| m | Mean, default is NULL |

### Value

The C-statistic

## calc_riley_vars *Calculates Approximate Variances and Covariance for Performance Metrics*

### Description

Calculates approximate variances performance metrics and covariance of calibration intercept and slope using the Riley framework

### Usage

```
calc_riley_vars(N, parms)
```

### Arguments

| | |
|---|---|
| N | sample size of the validation dataset |
| parms | list containing model and distribution parameters: prev: expected prevalence cstat: c-statistic of the model dist_type: one of ("logitnorm", "beta, "probitnorm") dist_parm1: first parameter of the distribution dist_parm2: second parameter of the distribution cal_int: calibration intercept cal_slp: calibration slope |

### Value

list of approximate variances and covariance of the performance metrics.

## calc_se_sp *Calculates the Sensitivity and Specificity*

### Description

Calculate the sensitivity and specificity of the model at given threshold

### Usage

```
calc_se_sp(dist_type, dist_parms, cal_int, cal_slp, threshold, prev)
```

### Arguments

| | |
|---|---|
| dist_type | The distribution type, one of c("logitnorm", "beta", "probitnorm"). |
| dist_parms | Vector of the two parameters of interest given the distribution. |
| cal_int | The calibration intercept. |
| cal_slp | The calibration slope. |
| threshold | The risk threshold |
| prev | The outcome prevalence, the expectation of the model |

### Value

A vector containing sensitivity and specificity

### Examples

```
calc_se_sp("beta", c(1,1), 0.9, 0.75, 0.5, 0.5)
```

## find_n_mean *Calculates Sample Size Given Target Mean CI*

### Description

Calculates sample size N, so that the mean confidence interval is equal to given target, assumes function is decreasing and convex

### Usage

```
find_n_mean(target, N, ciws, decreasing = T, convex = T)
```

### Arguments

| | |
|---|---|
| target | The target mean confidence interval width |
| N | Sample sizes corresponding to each row of ciws,= |
| ciws | Matrix of confidence intervals widths, each row corresponding to N |
| decreasing | Logical. Constraining function to decreasing |
| convex | Logical. Constraining function to convex |

### Value

Integer. Estimated sample size needed to achieve the target

---

## find_n_quantile *Calculates Sample Size Given Target Quantile*

---

### Description

Find sample size N, so that the specified quantile is equal to given target

### Usage

```
find_n_quantile(target, N, q, ciws)
```

### Arguments

| | |
|---|---|
| target | The desired quantile target value |
| N | Sample sizes corresponding to each row of ciws |
| q | Desired quantile level, between 0 and 1. |
| ciws | A matrix of confidence intervals widths, each row corresponding to N |

### Value

Estimated sample size needed to achieve the target

`infer_cal_int_from_mean` *Infer Calibration Intercept from Mean Calibration*

## Description

Infer calibration intercept from mean calibration given a fixed calibration slope and a given distribution for calibrated risks

## Usage

```
infer_cal_int_from_mean(dist_type, dist_parms, cal_mean, cal_slp, prev = NULL)
```

## Arguments

| | |
|---|---|
| `dist_type` | The distribution type, one of c("logitnorm", "probitnorm", "beta"). |
| `dist_parms` | The two parameters that index the type. |
| `cal_mean` | The mean calibration. |
| `cal_slp` | The calibration slope. |
| `prev` | Outcome prevalence. Optional; if not provided, estimate is as the expected value of the distribution of calibrated risks. |

## Value

The estimated calibration intercept

`infer_cal_int_from_oe` *Infer Calibration Intercept from O/E ratio*

## Description

Infer calibration intercept from observed-to-expected outcome ratio given a fixed calibration slope and a given distribution for calibrated risks

## Usage

```
infer_cal_int_from_oe(dist_type, dist_parms, cal_oe, cal_slp, prev = NULL)
```

## Arguments

| | |
|---|---|
| `dist_type` | The distribution type, one of c("logitnorm", "probitnorm", "beta"). |
| `dist_parms` | The two parameters that index the type. |
| `cal_oe` | The observed-to-expected outcome ratio. |
| `cal_slp` | The calibration slope. |
| `prev` | Outcome prevalence. Optional; if not provided, estimate is as the expected value of the distribution of calibrated risks. |

### Value

The estimated calibration intercept

---

`infer_correlation` *Calculates Correlation*

---

### Description

Calculates correlation based on simulated data

### Usage

```
infer_correlation(dist_type, dist_parms, cal_int, cal_slp, n, n_sim)
```

### Arguments

| | |
|---|---|
| `dist_type` | The distribution type |
| `dist_parms` | The two parameters of interest for the given distribution type |
| `cal_int` | The calibration intercept. |
| `cal_slp` | The calibration slope. |
| `n` | number of observations for each simulation. |
| `n_sim` | number of simulations |

### Value

correlation among the simulated data

---

`inv_mean_quantile` *Calculates the Model Parameters Given Quantile*

---

### Description

Calculate the model parameters given the distribution type, mean, quantile, and percentile.

### Usage

```
inv_mean_quantile(type, m, q, p)
```

### Arguments

| | |
|---|---|
| `type` | The distribution type, one of c("norm", "beta", "logitnorm", "probitnorm"). |
| `m` | Mean of the of distribution. |
| `q` | The quantile value. |
| `p` | The percentile at which the quantile occurs. |

### Value

The model parameters of the given type.

---

`inv_moments` *Calculates the Model Parameters Given Moments*

---

### Description

Calculates the model parameters of interest given the first two moments.

### Usage

```
inv_moments(type, moments)
```

### Arguments

| | |
|---|---|
| `type` | The distribution type, one of c("norm", "beta", "logitnorm"). |
| `moments` | A numeric vector containing the first two moments of the model |

### Value

Returns the two parameters for each model. mean and sd for norm mu and sigma for logitnorm shape1 (alpha) and shape2 (beta) for beta

---

`isaric` *Isaric Dataset*

---

### Description

Data from the International Severe Acute Respiratory and Emerging Infection Consortium regarding Regions in the UK.

### Usage

```
isaric
```

### Format

A data frame with 8 rows and 10 columns

**Region** Region where the sample was drawn

**Sample_Size** Raw number of total subjects available in the region's dataset

**n** Number of subjects used in analysis after exclusions

**n_events** Number of positive subjects

**cstat** C-statistic

**cstat_l** Lower bound for the confidence interval of the C-statistic

**cal_mean** Calibration Mean

**cal_mean_l** Lower bound for the confidence interval of the calibration mean

**cal_slope** Calibration slope

**cal_slope_l** Lower bound of the confidence interval of the calibration slope

## Source

Simulated Data

---

`moments` *Mean and Variance Calculator*

---

## Description

Calculates the first two moments (mean and variance) of the given model type and parameters.

## Usage

```
moments(type, parms)
```

## Arguments

| | |
|---|---|
| `type` | The distribution type, one of c("norm", "beta", "logitnorm", "probitnorm"). |
| `parms` | A numeric vector containing parameters relevant to the model. |

## Value

A numeric vector representing the mean and variance.

---

`plot_cal_distance` *Plots Calibration Distance from Simulation Curves*

---

## Description

simulates calibration curves based on given method, and uses plot to visualize calibration distance (difference between predicted and observed)

## Usage

```
plot_cal_distance(N, sample, method = "loess", X = (1:99)/100)
```

### Arguments

| | |
|---|---|
| N | Number of observations to simulate in each sample |
| sample | Data frame with columns: dist_type: distribution type dist_parm1: first distribution parameter (e.g. mean, alpha, shape1) dist_parm2: second distribution parameter (e.g. sd, beta, shape2) cal_int: calibration intercept cal_slp: calibration slope |
| method | One of loess or line, on default is loess |
| X | Vector of predicted probabilities, on default is 0.01 to 0.99 |

### Value

Plot of simulated calibration curves

### Examples

```
sample <- data.frame(
dist_type = rep("beta", 3),
dist_parm1 = c(1,2,3),
dist_parm2 = c(3,4,5),
cal_int = c(0, 0.05, 0.1),
cal_slp = c(1, 0.9, 0.8))
plot_cal_distance(N=200, sample=sample)
```

---

## plot_cal_instability *Plots Calibration Instability from Simulated Calibration Curves*

---

### Description

Simulates calibration curves based on given method, and uses plot to visualize calibration instability.

### Usage

```
plot_cal_instability(N, sample, method = "loess", X = (1:99)/100)
```

### Arguments

| | |
|---|---|
| N | Number of observations to simulate in each sample |
| sample | Data frame with columns: dist_type: distribution type dist_parm1: first distribution parameter (e.g. mean, alpha, shape1) dist_parm2: second distribution parameter (e.g. sd, beta, shape2) cal_int: calibration intercept cal_slp: calibration slope |
| method | One of loess or line, on default is loess |
| X | Vector of predicted probabilities, on default is 0.01 to 0.99 |

## Value

Plot of simulated calibration curves

## Examples

```
sample <- data.frame(
dist_type = rep("beta", 3),
dist_parm1 = c(1,2,3),
dist_parm2 = c(3,4,5),
cal_int = c(0, 0.05, 0.1),
cal_slp = c(1, 0.9, 0.8))
plot_cal_instability(N=200, sample=sample)
```

---

# process_evidence *Transforms Evidence Into Standardized Format*

---

## Description

Verifies that an evidence object has the required members and standardizes it into a `bpm_evidence` object. Each element's distribution is recorded together with both its native parameters (`$parms`) and its first two moments (`$moments`, named `m` and `v`).

## Usage

```
process_evidence(evidence)
```

## Arguments

| | |
|---|---|
| `evidence` | A named list of evidence elements. The required members are:<br>• `prev`: outcome prevalence (defaults to a `beta` distribution),<br>• `cstat`: c-statistic (defaults to a `beta` distribution),<br>• `cal_slp`: calibration slope (defaults to `norm`), and<br>• exactly one of `cal_mean` (mean calibration), `cal_oe` (observed-to-expected ratio), or `cal_int` (calibration intercept), each defaulting to `norm`.<br>Each element may be given either as a formula, `name ~ dist(par1, par2)`, or as a named list, `name = list(type = "dist", ...)`. The supported distributions (`type`) are `"norm"`, `"beta"`, `"logitnorm"`, and `"probitnorm"`. |

## Details

The two parameters of each element may be characterized flexibly, as either native distribution parameters or summary moments. The parameters must be **either all unnamed or all named** (a mix such as `beta(0.4, var = 0.04)` is ambiguous and raises an error).

**Unnamed (positional)** parameters are taken as the native parameters of the distribution:

- `norm(mean, sd)`,

- beta(shape1, shape2),
- logitnorm(mu, sigma),
- probitnorm(mu, sigma).

**Named** parameters are matched against the following aliases (pick one pair per element):

- moments with a variance: mean/var (or m/v),
- moments with a standard deviation: mean/sd (or m/sd),
- a mean and an upper 97.5\
- native beta parameters: alpha/beta,
- native logitnorm/probitnorm parameters: mu/sigma.

When moments are supplied, the native parameters are obtained by the method of moments (or, for cih, by matching the requested quantile).

## Value

A bpm_evidence object: the standardized, restructured evidence list.

## Examples

```
# Formula form, mixing native parameters and moments:
evidence <- list(
 prev     ~ beta(116, 155),       # native beta parameters
 cstat    ~ beta(mean = 0.76, sd = 0.006),
 cal_mean ~ norm(-0.009, 0.125),
 cal_slp  ~ norm(0.995, 0.024))
process_evidence(evidence = evidence)

# Equivalent named-list form:
evidence <- list(
 prev=list(type="beta", mean=0.38, sd=0.2),
 cstat=list(mean=0.7, sd=0.05),
 cal_int=list(mean=0.2, sd=0.2),
 cal_slp=list(mean=0.8, sd=0.3))
process_evidence(evidence=evidence)
```

---

# rbnorm *Generates Samples From Normal Distribution*

---

## Description

generates samples from a normal distribution using marginal means, variances, and covariance

## Usage

```
rbnorm(n, mu1, mu2, var1, var2, cov)
```

### Arguments

| | |
|---|---|
| n | Number of samples to be generated |
| mu1 | Mean of first variable |
| mu2 | Mean of second variable |
| var1 | Variance of first variable |
| var2 | Variance of second variable |
| cov | Covariance between the two variables |

### Value

Matrix of nx2 where column 1 contains samples for the first variable, and column 2 contains samples for the second variable conditioned on the first

---

## riley_samp *Calculates Sample Size that Achieves Target CI Widths*

---

### Description

Calculates sample size that achieves target confidence interval widths using Riley's framework

### Usage

```
riley_samp(target_ciws, parms)
```

### Arguments

| | |
|---|---|
| target_ciws | Named list containing target confidence interval width for at least one of: prev: prevalence cstat: c-statistic cal_mean: mean calibration cal_oe: observed to expected outcome ratio cal_int: calibration intercept cal_slp: calibration slope |
| parms | List containing model parameters and distribution: prev: expected prevalence cstat: c-statistic of the model dist_type: one of ("logitnorm", "beta, "probitnorm") dist_parm1: first parameter of the distribution dist_parm2: second parameter of the distribution cal_int: calibration intercept cal_slp: calibration slope |

### Value

A named list of estimated sample sizes that achieve target confidence interval widths: fciw.prev, fciw.cstat, fciw.cal_mean, fciw.cal_oe, fciw.cal_int, fciw.cal_slp

# Index