

Package ‘BayesSurveillance’

July 11, 2026

Type Package

Title Bayesian Surveillance Methods for Healthcare Performance Monitoring

Version 0.0.2

Description Provides Bayesian surveillance methods for prospective monitoring of healthcare performance, patient safety, and clinical quality indicators. The package implements beta-binomial monitoring for binary outcomes, gamma-Poisson monitoring for count outcomes, posterior predictive alert probabilities, Bayesian early-warning signal detection, risk-adjusted surveillance, simulation tools, decision-support methods, and graphical summaries. These methods support continuous performance monitoring and timely detection of adverse trends in healthcare systems. The methodology is motivated by established risk-adjusted monitoring, sequential surveillance, and healthcare quality-improvement frameworks
<doi:10.1093/biostatistics/1.4.441>,
<doi:10.1002/sim.1546>,
<doi:10.1136/bmjqs.2008.031831>, and
<doi:10.1136/bmjqs-2016-005526>.

License GPL-3

Encoding UTF-8

RoxygenNote 7.3.3

Depends R (>= 4.2.0)

Imports stats

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

URL <https://github.com/zerish12/BayesSurveillance>

BugReports <https://github.com/zerish12/BayesSurveillance/issues>

NeedsCompilation no

Author Muhammad Zahir Khan [aut, cre]

Maintainer Muhammad Zahir Khan <zahirstat007@gmail.com>

Repository CRAN
Date/Publication 2026-07-11 09:00:14 UTC

Contents

adaptive_update	2
compute_dynamic_bewrs	3
estimate_peib	3
evaluate_policy	4
fit_bewrs	4
plot_peib_distribution	5
plot_policy_summary	5
recommend_action	6
simon_two_stage	6
simulate_surveillance_data	7
update_policy	8
Index	9

adaptive_update	<i>Adaptive BEWRS policy update</i>
-----------------	-------------------------------------

Description

Runs the full adaptive surveillance pipeline on a dataset: fit BEWRS, compute Dynamic BEWRS, estimate PEIB, and recommend Bayes-optimal actions.

Usage

```
adaptive_update(data, ...)
```

Arguments

- data Surveillance data.
- ... Passed to fit_bewrs().

Value

Policy recommendations with Dynamic BEWRS, PEIB, and expected loss.

compute_dynamic_bewrs *Compute Dynamic BEWRS scores*

Description

Compute Dynamic BEWRS scores

Usage

```
compute_dynamic_bewrs(
  fit,
  id_cols = c("provider", "pathway"),
  time_col = "time",
  window = 3,
  weights = c(current = 1, persistence = 0.8, deterioration = 0.6)
)
```

Arguments

fit	Object from fit_bewrs().
id_cols	Columns defining surveillance units.
time_col	Time column.
window	Number of previous periods for persistence.
weights	Numeric vector for current risk, persistence, deterioration.

Value

Data frame with posterior risk components and dynamic_bewrs.

estimate_peib *Estimate Provider-specific Expected Intervention Benefit (PEIB)*

Description

PEIB is the expected reduction in future breach probability under each action.

Usage

```
estimate_peib(
  risk_data,
  actions = c("no_action", "monitor", "review", "escalate"),
  effect_prior = c(no_action = 0, monitor = 0.05, review = 0.18, escalate = 0.28)
)
```

Arguments

risk_data Output from compute_dynamic_bewrs().
 actions Character vector of actions.
 effect_prior Named numeric vector giving prior expected relative risk reduction by action.

Value

Data frame in long format with action-specific PEIB.

evaluate_policy	<i>Evaluate policy performance</i>
-----------------	------------------------------------

Description

Evaluate policy performance

Usage

```
evaluate_policy(recommendations, outcome_col = "future_breach")
```

Arguments

recommendations
 Output from recommend_action().
 outcome_col Column with future breach outcome if available.

Value

Named performance summary.

fit_bewrs	<i>Fit a lightweight Bayesian early-warning surveillance model</i>
-----------	--

Description

This first implementation uses an empirical-Bayes beta-binomial shrinkage model. It is intended as a stable package foundation before adding Stan/brms backends.

Usage

```
fit_bewrs(
  data,
  numerator = "numerator",
  denominator = "denominator",
  target = "target"
)
```

Arguments

data	Data frame with numerator, denominator, provider, pathway, time, and target columns.
numerator	Column name for successes / patients meeting target.
denominator	Column name for eligible patients.
target	Column name or numeric target threshold.

Value

An object of class bayes_surveillance_fit.

plot_peib_distribution *Plot PEIB distribution*

Description

Plot PEIB distribution

Usage

```
plot_peib_distribution(policy)
```

Arguments

policy	Output from recommend_action() or adaptive_update().
--------	--

Value

Invisibly returns PEIB values.

plot_policy_summary *Plot policy action counts*

Description

Plot policy action counts

Usage

```
plot_policy_summary(policy)
```

Arguments

policy	Output from recommend_action() or adaptive_update().
--------	--

Value

Invisibly returns the action count table.

recommend_action	<i>Recommend Bayes-optimal surveillance action</i>
------------------	--

Description

Recommend Bayes-optimal surveillance action

Usage

```
recommend_action(
  peib_data,
  action_cost = c(no_action = 0, monitor = 0.25, review = 1, escalate = 2),
  breach_cost = 10
)
```

Arguments

peib_data	Output from estimate_peib().
action_cost	Named numeric vector of action costs.
breach_cost	Cost of future breach.

Value

One recommendation per original row.

simon_two_stage	<i>Simon Two-Stage Phase II Trial Design</i>
-----------------	--

Description

Searches for a Simon two-stage design for a single-arm phase II trial with a binary outcome. The design compares an unacceptable response probability under the null hypothesis with a desirable response probability under the alternative hypothesis, while controlling the type I error rate and ensuring the required power.

Usage

```
simon_two_stage(p0, p1, alpha = 0.05, power = 0.8, n_max = 100)
```

Arguments

p0	Unacceptable response probability under the null hypothesis.
p1	Desirable response probability under the alternative hypothesis.
alpha	Maximum type I error rate. Default is 0.05.
power	Minimum desired power. Default is 0.80.
n_max	Maximum total sample size to search. Default is 100.

Value

A data frame containing the selected two-stage design.

Examples

```
simon_two_stage(
  p0 = 0.20,
  p1 = 0.40,
  alpha = 0.20,
  power = 0.60,
  n_max = 15
)
```

```
simon_two_stage(
  p0 = 0.20,
  p1 = 0.40,
  alpha = 0.05,
  power = 0.80,
  n_max = 80
)
```

simulate_surveillance_data

Simulate provider-pathway surveillance data

Description

Simulate provider-pathway surveillance data

Usage

```
simulate_surveillance_data(
  n_provider = 20,
  n_pathway = 5,
  n_time = 18,
  seed = NULL
)
```

Arguments

n_provider	Number of providers.
n_pathway	Number of pathways.
n_time	Number of time periods.
seed	Optional random seed.

Value

A data.frame with provider, pathway, time, denominator, numerator, performance, target, and future_breach.

update_policy	<i>Update surveillance policy with old and new data</i>
---------------	---

Description

This function refits the BEWRS workflow after appending new surveillance records to an existing dataset or an existing policy output. It returns updated Bayes-optimal action recommendations, not only a fitted model.

Usage

```
update_policy(old_data, new_data, keep_original_cols = TRUE, ...)
```

Arguments

old_data	Existing raw surveillance data or previous policy output.
new_data	New surveillance data.
keep_original_cols	Logical; if TRUE, strips previous policy-only columns before binding.
...	Passed to fit_bewrs().

Value

Updated policy recommendations.

Index

`adaptive_update`, [2](#)
`compute_dynamic_bewrs`, [3](#)
`estimate_peib`, [3](#)
`evaluate_policy`, [4](#)
`fit_bewrs`, [4](#)
`plot_peib_distribution`, [5](#)
`plot_policy_summary`, [5](#)
`recommend_action`, [6](#)
`simon_two_stage`, [6](#)
`simulate_surveillance_data`, [7](#)
`update_policy`, [8](#)