

Package ‘bruno’

August 21, 2026

Title Predicting User-Defined Event Recurrence under Exchangeability

Version 0.1.0

Description Implements analytical prediction of recurrence for user-defined binary events; 'bruno' abbreviates Beta-Bernoulli Recurrence for Unobserved Next Outcomes. The procedure applies when the observed and future event indicators are judged exchangeable for the intended prediction. For an indefinitely extendible exchangeable binary sequence, de Finetti's representation theorem expresses the assigned joint probabilities as a mixture of Bernoulli laws over a mixing distribution on the unit interval (de Finetti, 1931) [doi:10.4064/fm-17-1-298-329](https://doi.org/10.4064/fm-17-1-298-329). The package adopts a beta distribution as an additional parametric specification of this mixing distribution. Users specify an initial probability μ_0 assigned to the event and a positive concentration parameter τ , giving beta parameters $a = \mu_0 * \tau$ and $b = (1 - \mu_0) * \tau$. If the declared event occurs s times among n observed cases, conditioning gives $\text{Beta}(a + s, b + n - s)$. From this conditional assessment, the package computes analytically the probability assigned to occurrence of the same event in the next exchangeable case and, for a prespecified future sample size, the exact beta-binomial predictive distribution of the number of future event occurrences. Events may be supplied directly as logical or binary indicators or defined from paired pre-post measurements through a user-specified logical expression. Prediction may be performed for a single predictive class or separately across user-defined predictive classes, using common or class-specific initial probabilities and concentration parameters. Cases for which event status cannot be determined, and cases with missing predictive-class membership in grouped analyses, are excluded without imputation; case-level classification and inclusion information are retained for audit purposes. Summary methods provide central probability intervals for the conditional beta assessment and, for future samples larger than one case, predictive intervals for the future recurrence count. The package is intended for psychological, educational, pilot-study, and research decision-making applications in which recurrence of an explicitly defined event is the predictive target and the predictive relevance of observed cases for future cases can be substantively justified. The resulting probabilities concern recurrence

of the declared event within the stated predictive class and do not independently establish latent change, intervention efficacy, causal effects, measurement validity, or a research decision.

```
License GPL-3
Encoding UTF-8
Depends R (>= 4.1.0)
Imports stats
Suggests testthat (>= 3.0.0)
Config/testthat/edition 3
LazyData true
NeedsCompilation no
Config/roxygen2/version 8.1.0
Author Irene Gianceselli [aut, cre] (ORCID:
  <https://orcid.org/0000-0002-8689-3035>)
Maintainer Irene Gianceselli <irene.gianceselli@unibz.it>
Repository CRAN
Date/Publication 2026-08-21 13:30:38 UTC
```

Contents

bruno	2
bruno_example	6
predict.bruno_fit	8
print.bruno_fit	9
print.summary_bruno_fit	10
summary.bruno_fit	11
Index	13

bruno	<i>Predict Recurrence of a Measurement-Defined Event</i>
-------	--

Description

‘bruno()’ assigns predictive probabilities to the recurrence of a prespecified binary event under a stated judgement of exchangeability. The event can be defined from paired pre-post measurements through a user-specified logical expression or supplied directly as an already classified binary event.

Usage

```
bruno(
  data = NULL,
  id = NULL,
  pre = NULL,
  post = NULL,
  event,
  mu0,
  tau,
  group = NULL,
  future_n = 1L
)
```

Arguments

data	Optional data frame containing the observations. Required when ‘pre’ and ‘post’ are used to define the event. It may also be used with a directly specified event when case identifiers or predictive classes are stored in the same data frame.
id	Optional unquoted column name identifying cases when ‘data’ is supplied, or an optional identifier vector when ‘data’ is omitted. Identifiers are retained for case-level classification and diagnostic reporting and do not enter the predictive calculation.
pre	Optional unquoted column name identifying the pre-measurement. Used together with ‘post’ when the event is defined from paired measurements.
post	Optional unquoted column name identifying the post-measurement. Used together with ‘pre’ when the event is defined from paired measurements.
event	Definition of the binary event whose recurrence is to be predicted. When ‘pre’ and ‘post’ are supplied, ‘event’ is evaluated with ‘pre’ and ‘post’ referring to the selected measurement columns. Otherwise, ‘event’ is evaluated directly and must produce a logical vector or a numeric vector containing only ‘0’ and ‘1’, apart from missing values.
mu0	Initial probability assigned to the event before the observed recurrence information is incorporated. Must be numeric, finite, and strictly between ‘0’ and ‘1’. When ‘group’ is supplied, ‘mu0’ may be a single value common to all predictive classes or a named numeric vector containing exactly one value for each analysed class.
tau	Positive finite concentration parameter of the initial beta specification. When ‘group’ is supplied, ‘tau’ may be a single value common to all predictive classes or a named numeric vector containing exactly one value for each analysed class.
group	Optional unquoted column name identifying predictive classes when ‘data’ is supplied, or an optional grouping vector when ‘data’ is omitted. Prediction is performed separately within each analysed predictive class.
future_n	Positive integer giving the prespecified number of cases in the future sample. Defaults to ‘1’.

Details

Let E_i denote the binary event defined for case i from the observed measurement outcome or outcomes. ‘bruno()’ does not assign a substantive interpretation such as improvement, deterioration, increase, or decrease to the event. The event definition is supplied by the user.

Suppose that the event occurs in s of n observed cases. The observed proportion s/n describes the recurrence frequency in those cases. It does not by itself determine the probability assigned to a future occurrence.

Let $E_1, E_2, \dots$ be judged exchangeable for the intended prediction. For an indefinitely extendible exchangeable binary sequence, de Finetti’s representation theorem expresses the assigned joint probabilities as a mixture of Bernoulli laws over a mixing distribution on the unit interval. Exchangeability is a judgement concerning the predictive relevance of the observed cases for future cases; it is not estimated or tested by ‘bruno()’.

‘bruno()’ adopts a beta distribution as an additional parametric specification of the mixing distribution. The beta specification is not implied by de Finetti’s representation theorem.

Before the observed recurrence information is incorporated, the user supplies an initial event probability μ_0 and a concentration parameter τ . The corresponding beta parameters are

$$a = \mu_0 \tau$$

and

$$b = (1 - \mu_0) \tau.$$

Thus, $\mu_0 = a/(a + b)$ and $\tau = a + b$. The parameter τ expresses the concentration of the initial beta assessment. It is not a parameter supplied by de Finetti’s representation theorem and does not automatically represent a literal number of previous observations.

Let $A_{n,s}$ denote the observed information that exactly s of the first n events occurred. Conditioning gives

$$x \mid A_{n,s} \sim \text{Beta}(a + s, b + n - s).$$

The predictive probability assigned to occurrence of the event for the next case is

$$P(E_{n+1} \mid A_{n,s}) = \frac{a + s}{a + b + n}.$$

Equivalently,

$$P(E_{n+1} \mid A_{n,s}) = \frac{\tau}{\tau + n} \mu_0 + \frac{n}{\tau + n} \frac{s}{n}.$$

For a prespecified future sample of size m , let T denote the number of future occurrences of the same event. The predictive distribution is beta-binomial:

$$P(T = t \mid A_{n,s}) = \binom{m}{t} \frac{B(a + s + t, b + n - s + m - t)}{B(a + s, b + n - s)}, \quad t = 0, \dots, m,$$

where B denotes the beta function.

The conditional beta distribution permits direct calculation of central probability intervals for x . These intervals describe the dispersion of the conditional beta assessment and are distinct from the predictive probability assigned to the next binary event.

For a future sample with more than one case, the beta-binomial distribution permits direct calculation of predictive intervals for the future recurrence count T . Conditional and predictive intervals are obtained analytically from their respective distributions. Their reporting level is specified when the fitted object is summarised or queried and is not fixed during fitting.

When 'group' is supplied, the recurrence information is conditioned and prediction is performed separately within each analysed predictive class. A common 'mu0' and 'tau' may be supplied for all classes, or class-specific values may be supplied as named vectors. Group membership does not establish exchangeability; the exchangeability judgement remains part of the predictive specification.

Cases for which the declared event cannot be determined do not contribute to n or s . When predictive classes are supplied, cases with missing class membership likewise do not contribute to a class-specific calculation. No missing values are imputed. Case-level information is retained so that analysed and excluded observations remain identifiable.

The resulting probabilities concern recurrence of the declared measurement-defined event. They do not by themselves represent latent change, intervention efficacy, measurement validity, or a research decision.

Value

An object of class "bruno_fit" containing:

results A list containing one set of predictive results for the ungrouped analysis or one set for each analysed predictive class. Each set contains n , s , the observed recurrence proportion, 'mu0', 'tau', the initial and conditional beta parameters, the predictive probability for the next case, and the beta-binomial predictive distribution for the prespecified future sample.

events Case-level audit data containing the supplied identifier or original row index, event classification, inclusion status, and the measurement and grouping information relevant to the classification.

future_n Prespecified size of the future sample.

n_total Number of cases presented for classification.

n_analyzed Number of cases contributing to the recurrence calculation.

n_excluded Number of cases excluded from the recurrence calculation.

grouped Logical value indicating whether predictive classes were supplied.

call Matched function call.

References

de Finetti, B. (1937). La prévision: Ses lois logiques, ses sources subjectives. *Annales de l'Institut Henri Poincaré*, *7*(1), 1–68.

Examples

```

dat <- data.frame(
  id = paste0("P", 1:6),
  score_pre = c(2, 3, 4, 2, 5, 3),
  score_post = c(1, 3, 5, 2, 3, 4)
)

fit <- bruno(
  data = dat,
  id = id,
  pre = score_pre,
  post = score_post,
  event = post <= pre,
  mu0 = 0.50,
  tau = 10
)

fit

event_status <- c(1, 1, 0, 1, 1, 0)
participant <- paste0("P", 1:6)

fit_direct <- bruno(
  id = participant,
  event = event_status,
  mu0 = 0.50,
  tau = 10
)

fit_direct

```

bruno_example

Constructed Example Data for bruno

Description

A small constructed data set for illustrating event classification, missing-data handling, predictive classes, and case-level traceability with [bruno()].

Format

A data frame with 12 rows and 4 variables:

id Character case identifier. Values are unique and non-missing.

pre Integer pre-measurement value.

post Integer post-measurement value. One value is missing so that an event depending on both ‘pre’ and ‘post’ can be non-determinable for that case.

group Character predictive-class label. Two non-missing classes, "A" and "B", are represented. One value is missing so that exclusion due to unavailable predictive-class membership can be illustrated separately from exclusion due to an indeterminate event.

Details

The data are artificial and do not represent observations from an empirical psychological, educational, clinical, or intervention study. The measurement values and predictive-class labels have no substantive interpretation beyond their use in package examples and tests.

No event classification is stored in the data set. The event must be declared explicitly by the user when `[bruno()]` is called.

For example, the declaration

```
'event = post <= pre'
```

defines one possible binary measurement event for illustration. Under that declaration, 8 events occur among 11 cases for which the event is determinable in the ungrouped analysis.

When 'group' is also supplied, predictive class "A" contains 4 event occurrences among 5 analysable cases and predictive class "B" contains 3 event occurrences among 5 analysable cases. The case with missing 'group' is retained in the case-level audit information but does not contribute to a class-specific recurrence calculation.

These counts are consequences of the illustrative event declaration above and are not intrinsic properties or substantive labels of the measurement values.

Source

Constructed solely for examples and software tests in 'bruno'; no empirical observations are included.

Examples

```
data(bruno_example)

bruno_example

fit <- bruno(
  data = bruno_example,
  id = id,
  pre = pre,
  post = post,
  event = post <= pre,
  mu0 = 0.50,
  tau = 10
)

fit

fit_grouped <- bruno(
  data = bruno_example,
  id = id,
  pre = pre,
```

```

    post = post,
    event = post <= pre,
    mu0 = 0.50,
    tau = 10,
    group = group
)

fit_grouped

```

predict.bruno_fit	<i>Predict Future Event Recurrence</i>
-------------------	--

Description

Computes the beta-binomial predictive distribution of the number of occurrences of the declared event in a future sample, using the conditional beta assessment stored in a "bruno_fit" object.

Usage

```

## S3 method for class 'bruno_fit'
predict(object, future_n = object$future_n, ...)

```

Arguments

object	An object of class "bruno_fit".
future_n	Optional positive integer giving the number of future cases for which the recurrence-count distribution is required. If omitted, the value originally supplied to [bruno()] is used.
...	Additional arguments. Currently unused.

Details

For a fitted predictive class with conditional beta parameters α^* and β^* , and a future sample of size m , the number of future event occurrences T has predictive distribution

$$P(T = t) = \binom{m}{t} \frac{B(\alpha^* + t, \beta^* + m - t)}{B(\alpha^*, \beta^*)}, \quad t = 0, \dots, m.$$

No simulation is used.

When 'future_n = 1', the probability associated with 't = 1' is equal to the next-case predictive probability stored in the fitted object:

$$P(T = 1) = \frac{\alpha^*}{\alpha^* + \beta^*}.$$

Value

A data frame containing the exact predictive distribution.

For an ungrouped fit, the returned columns are:

t Number of future event occurrences, from '0' to 'future_n'.

probability Predictive probability assigned to the corresponding recurrence count.

For a grouped fit, an additional 'group' column identifies the predictive class to which each distribution belongs.

Examples

```
dat <- data.frame(
  id = paste0("P", 1:6),
  pre = c(2, 3, 4, 2, 5, 3),
  post = c(1, 3, 5, 2, 3, 4)
)

fit <- bruno(
  data = dat,
  id = id,
  pre = pre,
  post = post,
  event = post <= pre,
  mu0 = 0.50,
  tau = 10,
  future_n = 5
)

predict(fit)

predict(fit, future_n = 10)
```

print.bruno_fit	<i>Print a bruno Fit</i>
-----------------	--------------------------

Description

Prints the principal observed and predictive quantities stored in an object returned by [bruno()].

Usage

```
## S3 method for class 'bruno_fit'
print(x, digits = max(3L, getOption("digits") - 3L), ...)
```

Arguments

<code>x</code>	An object of class <code>"bruno_fit"</code> .
<code>digits</code>	Number of significant digits used when printing numeric probabilities and parameters.
<code>...</code>	Additional arguments. Currently unused.

Details

No additional probability calculation is performed by this method.

Value

`'x'`, invisibly.

Examples

```
dat <- data.frame(
  id = paste0("P", 1:6),
  pre = c(2, 3, 4, 2, 5, 3),
  post = c(1, 3, 5, 2, 3, 4)
)

fit <- bruno(
  data = dat,
  id = id,
  pre = pre,
  post = post,
  event = post <= pre,
  mu0 = 0.50,
  tau = 10
)

print(fit)
```

```
print.summary_bruno_fit
```

Print a bruno Summary

Description

Prints the principal quantities and probability intervals contained in an object returned by `'summary.bruno_fit()'`.

Usage

```
## S3 method for class 'summary_bruno_fit'
print(x, digits = max(3L, getOption("digits") - 3L), ...)
```

Arguments

<code>x</code>	An object of class <code>"summary_bruno_fit"</code> .
<code>digits</code>	Number of significant digits used for printed numeric quantities.
<code>...</code>	Additional arguments. Currently unused.

Value

`'x'`, invisibly.

<code>summary.bruno_fit</code>	<i>Summarise a bruno Fit</i>
--------------------------------	------------------------------

Description

Summarises the observed recurrence information, the initial and conditional beta parameters, the next-case predictive probability, and mathematically defined probability intervals derived from the fitted distributions.

Usage

```
## S3 method for class 'bruno_fit'
summary(object, level = 0.95, ...)
```

Arguments

<code>object</code>	An object of class <code>"bruno_fit"</code> .
<code>level</code>	Reporting probability for the conditional and predictive intervals. Must be strictly between <code>'0'</code> and <code>'1'</code> . Defaults to <code>'0.95'</code> .
<code>...</code>	Additional arguments. Currently unused.

Details

For conditional beta parameters α^* and β^* ,

$$x \mid A_{n,s} \sim \text{Beta}(\alpha^*, \beta^*).$$

Let the requested reporting level be $1 - \gamma$. The central equal-tail conditional probability interval is

$$[F_{\text{Beta}}^{-1}(\gamma/2), F_{\text{Beta}}^{-1}(1 - \gamma/2)].$$

This interval describes the dispersion of the conditional beta assessment for x . It is not an interval around the predictive probability for the next event.

For a future sample of size greater than one, let T denote the future recurrence count and let Q_T denote the quantile function of its beta-binomial predictive distribution. The equal-tail predictive interval is

$$[Q_T(\gamma/2), Q_T(1 - \gamma/2)].$$

Because T is discrete, the probability contained by the predictive interval can exceed the requested reporting level.

No simulation is used to compute either interval.

Value

An object of class "summary_bruno_fit" containing:

call Matched call from the fitted object.

level Reporting probability used for the intervals.

future_n Future sample size stored in the fitted object.

results A list containing one summary for the ungrouped analysis or one summary per predictive class. Each element contains the quantities stored in the fitted result together with 'conditional_interval' and, when 'future_n > 1', 'predictive_interval'.

Examples

```
dat <- data.frame(
  id = paste0("P", 1:6),
  pre = c(2, 3, 4, 2, 5, 3),
  post = c(1, 3, 5, 2, 3, 4)
)
```

```
fit <- bruno(
  data = dat,
  id = id,
  pre = pre,
  post = post,
  event = post <= pre,
  mu0 = 0.50,
  tau = 10,
  future_n = 10
)
```

```
summary(fit)
```

```
summary(fit, level = 0.90)
```

Index

* **datasets**

bruno_example, [6](#)

bruno, [2](#)

bruno_example, [6](#)

predict.bruno_fit, [8](#)

print.bruno_fit, [9](#)

print.summary_bruno_fit, [10](#)

summary.bruno_fit, [11](#)