

Package: cbbinom (via r-universe)

October 16, 2024

Title Continuous Analog of a Beta-Binomial Distribution

Version 0.2.0

Description Implementation of the $d/p/q/r$ family of functions for a continuous analog to the standard discrete beta-binomial with continuous size parameter and continuous support with x in $[0, \text{size} + 1]$.

License MIT + file LICENSE

Suggests extraDistr, ggplot2, testthat ($\geq 3.0.0$)

Config/testthat/edition 3

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

LinkingTo BH, hypergeo2, Rcpp

Imports hypergeo2 ($\geq 0.2.0$), Rcpp

URL <https://github.com/zhuxr11/cbbinom>

BugReports <https://github.com/zhuxr11/cbbinom/issues>

Repository <https://zhuxr11.r-universe.dev>

RemoteUrl <https://github.com/zhuxr11/cbbinom>

RemoteRef HEAD

RemoteSha 231a97f9925fd15475f679e07d14400ff5444e60

Contents

cbbinom	2
Index	5

Description

Density, distribution function, quantile function and random generation for a continuous analog to the beta-binomial distribution with parameters size, alpha and beta. The usage and help pages are modeled on the d-p-q-r families of functions for the commonly-used distributions in the stats package.

Usage

```
dcbbinom(x, size, alpha = 1, beta = 1, ncp = 0, log = FALSE, prec = NULL)
```

```
pcbbinom(  
  q,  
  size,  
  alpha = 1,  
  beta = 1,  
  ncp = 0,  
  lower.tail = TRUE,  
  log.p = FALSE,  
  prec = NULL  
)
```

```
qcbbinom(  
  p,  
  size,  
  alpha = 1,  
  beta = 1,  
  ncp = 0,  
  lower.tail = TRUE,  
  log.p = FALSE,  
  prec = NULL,  
  tol = 1e-06,  
  max_iter = 10000L  
)
```

```
rcbbinom(  
  n,  
  size,  
  alpha = 1,  
  beta = 1,  
  ncp = 0,  
  prec = NULL,  
  tol = 1e-06,  
  max_iter = 10000L
```

)

Arguments

<code>x, q</code>	vector of quantiles.
<code>size</code>	number of trials (zero or more).
<code>alpha, beta</code>	non-negative parameters of the Beta distribution.
<code>ncp</code>	non-centrality parameter.
<code>log, log.p</code>	logical; if TRUE, probabilities <code>p</code> are given as <code>log(p)</code> .
<code>prec</code>	arguments passed on to genhypergeo , vectorized and recycled along with distribution parameters.
<code>lower.tail</code>	logical; if TRUE (default), probabilities are $P[X \leq x]$, otherwise, $P[X > x]$.
<code>p</code>	vector of probabilities.
<code>tol, max_iter</code>	arguments passed on to uniroot , vectorized and recycled along with distribution parameters.
<code>n</code>	number of observations. If <code>length(n) > 1</code> , the length is taken to be the number required.

Details

Derived from the continuous binomial distribution (Iliencko 2013), the continuous beta-binomial distribution is defined as:

$$P(x|n, \alpha, \beta) = \int_0^1 \frac{B_{1-p}(n+1-x, x)}{B(n+1-x, x)} \frac{p^{\alpha-1}(1-p)^{\beta-1}}{B(\alpha, \beta)} dp,$$

where x is the quantile, n is the size, $B_p(a, b) = \int_0^p u^{a-1}(1-u)^{b-1} du$ is the incomplete beta function.

When simplified, the distribution becomes:

$$P(x|n, \alpha, \beta) = \frac{\Gamma(n+1)B(n+1-x+\beta, \alpha)}{\Gamma(x)\Gamma(n+2-x)B(\alpha, \beta)} {}_3F_2(a; b; z),$$

where ${}_3F_2(a; b; z)$ is [generalized hypergeometric function](#), $a = \{1-x, n+1-x, n+1-x+\beta\}$, $b = \{n+2-x, n+1-x+\alpha+\beta\}$, $z = 1$.

Heuristically speaking, this distribution spreads the standard probability mass at integer x to the interval $[x, x+1]$ in a continuous manner. As a result, the distribution looks like a smoothed version of the standard, discrete beta-binomial but shifted slightly to the right. The support of the continuous beta-binomial is $[\emptyset, \text{size} + 1]$, and the mean is approximately $\text{size} * \alpha / (\alpha + \beta) + 1/2$.

Supplying `ncp != 0` moves the support of beta-binomial to $[\text{ncp}, \text{size} + 1 + \text{ncp}]$. For example, to build a continuous beta-binomial with approximately non-shifted mean, use `ncp = -0.5`.

These functions are also available in [Rcpp](#) as `cbbinom::cpp_[d/p/q/r]cbbinom()`, and their non-vectorized versions in [Rcpp](#) as `cbbinom::[d/p/q/r]cbbinom_()`. To use them, please use `[[Rcpp::depends(cbbinom)]]` and `#include <cbbinom.h>`.

Value

dcbbinom gives the density, pcbbinom the distribution function, qcbbinom the quantile function, and rcbbinom generates random deviates.

Invalid arguments will result in return value NaN, with a warning.

The length of the result is determined by `n` for `rcbbinom`, and is the maximum of the lengths of the numerical arguments for the other functions.

The numerical arguments other than `n` are recycled to the length of the result. Only the first elements of the logical arguments are used.

Note

Change log:

- 0.1.0 Xiurui Zhu - Initiate the function.
- 0.2.0 Xiurui Zhu - Re-implement distribution function with `BH` package, add `NULL` default tolerance, and add precision parameters.

References

Iliencko, Andreii (2013). Continuous counterparts of Poisson and binomial distributions and their properties. *Annales Univ. Sci. Budapest., Sect. Comp.* 39: 137-147. http://ac.inf.elte.hu/Vol_039_2013/137_39.pdf

Examples

```
# Density function
dcbbinom(x = 5, size = 10, alpha = 2, beta = 4)
# Distribution function
(test_val <- pcbbinom(q = 5, size = 10, alpha = 2, beta = 4))
# Quantile function
qcbbinom(p = test_val, size = 10, alpha = 2, beta = 4)
# Random generation
set.seed(1111L)
rcbbinom(n = 10L, size = 10, alpha = 2, beta = 4)
```

Index

BH, [4](#)

cbbinom, [2](#)

dcbbinom(cbbinom), [2](#)

generalized hypergeometric function, [3](#)

genhypergeo, [3](#)

pcbbinom(cbbinom), [2](#)

qcbbinom(cbbinom), [2](#)

rcbbinom(cbbinom), [2](#)

Rcpp, [3](#)

uniroot, [3](#)