Package ‘crossrun’

October 8, 2018

Version 0.1.0

Title Joint Distribution of Number of Crossings and Longest Run

Description Joint distribution of number of crossings and the longest run in a series of independent Bernoulli trials. The computations uses an iterative procedure where computations are based on results from shorter series. The procedure conditions on the start value and partitions by further conditioning on the position of the first crossing (or none).

Depends R (>= 3.5)

License GPL-3

Encoding UTF-8

URL https://github.com/ToreWentzel-Larsen/crossrun

LazyData true

Imports Rmpfr (>= 0.7-1)

RoxygenNote 6.1.0

Suggests knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation no

Author Tore Wentzel-Larsen [aut, cre], Jacob Anhøj [aut]

Maintainer Tore Wentzel-Larsen <tore.wentzellarsen@gmail.com>

Repository CRAN

Date/Publication 2018-10-08 14:20:10 UTC

R topics documented:

boxprobt ................................................................. 2
clshift ............................................................... 3
crossrunbin .......................................................... 3
crossrunchange .................................................... 4
A box cumulative sum is defined as the cumulative sum over a lower left rectangle. This function is primarily for use when the components are point probabilities for the number of crossings $C$ and the longest run $L$, then component $(c, l)$ in the result is the probability $P(C \geq c, L \leq l)$.

**Usage**

`boxprobt(mtrx)`

**Arguments**

- `mtrx` (mpfr array)

**Value**

mpfr array

**Examples**

```r
nill <- Rmpfr::mpfr(0L, 120)
one <- Rmpfr::mpfr(1L, 120)
two <- Rmpfr::mpfr(2L, 120)
contents <- c(one, nill, nill, one, one, two, two, two)
mtrx3 <- Rmpfr::mpfr2array(contents, dim = c(3, 3))
print(mtrx3)
print(boxprobt(mtrx3))
```
**clshift**

*Number of Crossings and Longest Run*

**Description**

Auxiliary function for simclbin, computing the number of crossings (type=0) or longest run (type=2) in a sequence of independent normal observations. Crossings and runs are related to whether the observations are above a shift.

**Usage**

```r
clshift(seri, shift = 0, type = 0)
```

**Arguments**

- `seri` numeric; seri a sequence of random draws
- `shift` numeric; shift for the observations
- `type` numeric; 0 number of crossings, 1 longest run

**Value**

number of crossings or longest run, numeric

---

**crossrunbin**

*Joint Distribution for Crossings and Runs*

**Description**

Joint probability distribution for the number of crossings C and the longest run L in a sequence of n independent Bernoulli observations with success probability p. To enhance precision, results are stored in mpfr arrays and the probabilities are multiplied by \(m^{n-1}\) for a multiplier m.

**Usage**

```r
crossrunbin(nmax = 100, prob = 0.5, mult = 2, prec = 120, printn = FALSE)
```

**Arguments**

- `nmax` max sequence length.
- `prob` success probability.
- `mult` multiplier for joint probabilities.
- `prec` mpfr precision.
- `printn` logical for progress output.
Value

list of joint probabilities.

Examples

```r
crb10.6 <- crossrunbin(nmax=10, prob=.6, printn=TRUE)
print(crb10.6$pt[[10]])
```

---

**crossrunchange**

*Joint Distribution for Crossings and Runs, Varying Success Probability.*

Description

Joint probability distribution for the number of crossings C and the longest run L in a sequence of n independent Bernoulli observations with possibly varying success probability. To enhance precision, results are stored in mpfr arrays and the probabilities are multiplied by $m^{n-1}$ for a multiplier m.

Usage

```r
crossrunchange(nmax = 100, prob = rep(0.5, 100), mult = 2, prec = 120, printn = FALSE)
```

Arguments

- `nmax`: max sequence length.
- `prob`: success probabilities.
- `mult`: multiplier for joint probabilities.
- `prec`: mpfr precision.
- `printn`: logical for progress output.

Value

list pt of joint probabilities. Cumulative probabilities qt within each row are also included. Further, mostly for code checking, lists pat and qat conditional on starting with a success, and pbt and qbt conditional of starting with a failure, are included.

Examples

```r
prob10 <- c(rep(.5,5),rep(.7,5))
crchange10 <- crossrunchange(nmax=10, prob=prob10, printn=TRUE)
print(crchange10$pt[[10]])
```
crossrunshift

Description

wrapper for crossrunbin, success probability = pnorm(shift).

Usage

crossrunshift(nmax = 100, shift = 0, mult = 2, prec = 120, printn = FALSE)

Arguments

- **nmax**: max sequence length.
- **shift**: mean of normal distribution.
- **mult**: multiplier for joint probabilities.
- **prec**: mpfr precision.
- **printn**: logical for progress output.

Value

list pt of joint probabilities. Cumulative probabilities qt within each row are also included. Further, mostly for code checking, lists pat and qat conditional on starting with a success, and pbt and qbt conditional of starting with a failure, are included.

Examples

crs20 <- crossrunshift(nmax=20,printn=TRUE)
pn(crs20$pt[[20]])

crossrunsymm

Joint Probabilities for Crossings and Runs, Symmetric Case

Description

Joint probability distribution for the number of crossings C and the longest run L in a sequence of n independent Bernoulli observations with success probability p. To enhance precision, results are stored in mpfr arrays and the probabilities are multiplied by \(m^{n-1}\) for a multiplier m. This is for the symmetric case with success probability 0.5, in which the multiplied probabilities are integers for the default value 2 of the multiplier.

Usage

crossrunsymm(nmax = 100, mult = 2, prec = 120, printn = FALSE)
Arguments

- \texttt{nmax} ; max sequence length.
- \texttt{mult} ; multiplier for joint probabilities. Default 2.
- \texttt{prec} ; mpfi precision.
- \texttt{printn} ; logical for including progress output.

Value

- \texttt{pt}, list of joint probabilities, multiplied with \( mn^{-1} \). In addition cumulative probabilities \( qt \) within each row are also included.

Examples

\begin{verbatim}
crs10 <- crossrunsymm(nmax=10,printn=TRUE)
\end{verbatim}

\begin{verbatim}
cumsumm
\end{verbatim}
Column-Wise Cumulative Sums

**Description**

Column-wise cumulative sums in mpfr array.

**Usage**

```r
cumsummcol(mtrx)
```

**Arguments**

- `mtrx` mpfr two-dimensional array.

**Value**

mpfr array with column-wise cumulative sums, same dimension as the original array.

**Examples**

```r
null <- mpfr::mpfr(0L, 1R0)
one <- mpfr::mpfr(1L, 1R0)
two <- mpfr::mpfr(2L, 1R0)
contents <- c(one, null, null, one, one, one, two, two, two)
mtrx3 <- mpfr::mpfr2array(contents, dim = c(3, 3))
print(mtrx3)
print(cumsummcol(mtrx3))
```

---

**exactbin**

**Exact Joint Probabilities for Low n**

**Description**

Exact joint probabilities, for low n, of the number of crossings C and the longest run L in n independent Bernoulli observations with success probability p. Probabilities are multiplied by $2^{n-1}$.

**Usage**

```r
exactbin(n, p = 0.5, prec = 120)
```

**Arguments**

- `n` number, length of sequence, at most 6.
- `p` success probability.
- `prec` precision in mpfr calculations. Default 120.
Value

mpfr array

Examples

`exactbin(n=6)`
`exactbin(n=5, p=0.6)`

---

**joint100.6**

*Joint probabilities, n=100, success probability 0.6*

Description

The joint probabilities of the number $C$ of crossings (0, ..., 99) and the longest run $L$ (1, ..., 100) in a series of $n=100$ independent Bernoulli observations for success probability 0.6. The probabilities are stored in the "times" representations, multiplied by $2^{100-1}$. Only the joint distributions for $n=15$, 60, 100 and success probabilities 0.5 and 0.6 are included in the package to avoid excessive storage, but many more cases are generated in the script crossrun1.R.

Usage

`joint100.6`

Format

matrix, 100 rows and 100 columns

Source

generated by the function crossrunbin and transformed from an Rmpfr array to a matrix

---

**joint100symm**

*Joint probabilities, n=100, symmetric case*

Description

The joint probabilities of the number $C$ of crossings (0, ..., 99) and the longest run $L$ (1, ..., 100) in a series of $n=100$ independent Bernoulli observations for the symmetric case (success probability 0.5). The probabilities are stored in the "times" representations, multiplied by $2^{100-1}$ and are integers in the symmetric case. Only the joint distributions for $n=15$, 60, 100 and success probabilities 0.5 and 0.6 are included in the package to avoid excessive storage, but many more cases are generated in the script crossrun1.R.

Usage

`joint100symm`
**joint1.6**

**Format**
matrix, 100 rows and 100 columns

**Source**
generated by the function crossrunsymm and transformed from an Rmpfr array to a matrix

---

**Description**
The joint probabilities of the number C of crossings (0, ... 14) and the longest run L (1, ..., 15) in a series of n=15 independent Bernoulli observations for success probability 0.6. The probabilities are stored in the "times" representations, multiplied by $2^{15-1} = 16348$. Only the joint distributions for n=15, 60, 100 and success probabilities 0.5 and 0.6 are included in the package to avoid excessive storage, but many more cases are generated in the script crossrun1.R.

**Usage**

`joint1.6`

---

**joint1Usymm**

**Format**
matrix, 15 rows and 15 columns

**Source**
generated by the function crossrunbin and transformed from an Rmpfr array to a matrix

---

**Description**
Joint probabilities of the number C of crossings (0, ... 14) and the longest run L (1, ..., 15) in a series of n=15 independent Bernoulli observations for the symmetric case (success probability 0.5). The probabilities are stored in the "times" representations, multiplied by $2^{15-1} = 16348$ and are integers in the symmetric case. Only the joint distributions for n=15, 60, 100 and success probabilities 0.5 and 0.6 are included in the package to avoid excessive storage, but many more cases are generated in the script crossrun1.R.

**Usage**

`joint1Usymm`
Format
matrix, 15 rows and 15 columns

Source
generated by the function crossrunsymm and transformed from an Rmpfr array to a matrix

---

joint60.6  
Joint probabilities, 60, success probability 0.6

Description
The joint probabilities of the number of crossings (0, ..., 59) and the longest run (1, ..., 60) in a series of n=60 independent Bernoulli observations for success probability 0.6. The probabilities are stored in the "times" representations, multiplied by $2^{60} - 1$. Only the joint distributions for n=15, 60, 100 and success probabilities 0.5 and 0.6 are included in the package to avoid excessive storage, but many more cases are generated in the script crossrun1.R.

Usage
joint60.6

Format
matrix, 60 rows and 60 columns

Source
generated by the function crossrunbin and transformed from an Rmpfr array to a matrix

---

joint60symm  
Joint probabilities, n=60, symmetric case

Description
The joint probabilities of the number of crossings (0, ..., 59) and the longest run (1, ..., 60) in a series of n=60 independent Bernoulli observations for the symmetric case (success probability 0.5). The probabilities are stored in the "times" representations, multiplied by $2^{60} - 1$ and are integers in the symmetric case. Only the joint distributions for n=15, 60, 100 and success probabilities 0.5 and 0.6 are included in the package to avoid excessive storage, but many more cases are generated in the script crossrun1.R.

Usage
joint60symm
**simclbin**

**Format**
matrix, 60 rows and 60 columns

**Source**
generated by the function crossrunsymm and transformed from an Rmpfr array to a matrix

---

**simclbin**

*Simulation of Independent Bernoulli Observations*

---

**Description**

Simulation of a sequence of independent Bernoulli Observations. To reduce the amount of random draws, each simulation is based on a sequence of standard normal variables, and whether each observation is above a shift defined by the binomial probabilities assumed.

**Usage**

```r
simclbin(nser = 100L, nsim = 1e+05UL, probs = c(0.5L, 0.6L, 0.7L, 0.8L, 0.9L))
```

**Arguments**

- `nser` length of sequence simulated
- `nsim` number of simulations
- `probs` binomial probabilities

**Value**
a data frame with the number of crossings and longest run for each probability. For instance the variables `nc0.5` and `lr0.5` are the number of crossings and the longest run for success probability 0.5. One row for each simulation.

**Examples**

```r
c10simbin <- simclbin(nser=30L, nsim=100L)
mean(c10simbin$nc0.5) # mean number of crossings, p=0.5
mean(c10simbin$lr0.9) # mean longest run, p=0.9
```
Index

*Topic **datasets**
  joint100.6, 8
  joint100symm, 8
  joint15.6, 9
  joint15symm, 9
  joint60.6, 10
  joint60symm, 10

boxprobt, 2

c1shift, 3
crossrunbin, 3
crossrunchange, 4
crossrunshift, 5
crossrunsymm, 5
cumsumm, 6
cumsummcol, 7

exactbin, 7

joint100.6, 8
joint100symm, 8
joint15.6, 9
joint15symm, 9
joint60.6, 10
joint60symm, 10

simclbin, 11