Package ‘skellam’

July 7, 2015

Version 0.1.3
Date 2015-07-02
Title Densities and Sampling for the Skellam Distribution
Author Jerry W. Lewis, Patrick E. Brown <patrick.brown@utoronto.ca>
Maintainer Patrick E. Brown <patrick.brown@utoronto.ca>
Description Functions for the Skellam distribution, including: density
(pmf), cdf, quantiles and random variates.
URL http://r-forge.r-project.org/projects/healthqueues/
License GPL (>= 2)
Imports stats
Suggests knitr
VignetteBuilder knitr
Repository CRAN
Repository/R-Forge/Project healthqueues
Repository/R-Forge/Revision 13
Repository/R-Forge/DateTimeStamp 2015-07-07 13:14:44
Date/Publication 2015-07-07 18:11:07
NeedsCompilation no

R topics documented:

Skellam ................................................................. 2

Index 5

1
The Skellam Distribution

Description

Density, distribution function, quantile function and random number generation for the Skellam distribution with parameters \( \lambda_1 \) and \( \lambda_2 \).

If \( Y_1 \) and \( Y_2 \) are Poisson variables with means \( \mu_1 \) and \( \mu_2 \) and correlation \( \rho \), then \( X = Y_1 - Y_2 \) is Skellam with parameters \( \lambda_1 = \mu_1 - \rho \sqrt{\mu_1 \mu_2} \) and \( \lambda_2 = \mu_2 - \rho \sqrt{\mu_1 \mu_2} \).

Usage

dskellam(x, lambda1, lambda2 = lambda1, log = FALSE)
pskellam(q, lambda1, lambda2 = lambda1, lower.tail = TRUE, log.p = FALSE)
qskellam(p, lambda1, lambda2 = lambda1, lower.tail = TRUE, log.p = FALSE)
rskellam(n, lambda1, lambda2 = lambda1)
dskellam.sp(x, lambda1, lambda2 = lambda1, log = FALSE)
pskellam.sp(q, lambda1, lambda2 = lambda1, lower.tail = TRUE, log.p = FALSE)

Arguments

- \( x, q \) vector of quantiles.
- \( p \) vector of probabilities.
- \( n \) number of observations. If \( \text{length}(n) > 1 \), the length is taken to be the number required.
- \( \lambda_1, \lambda_2 \) vectors of (non-negative) means.
- \( \log, \log.p \) logical; if TRUE, probabilities \( p \) are given as \( \log(p) \).
- lower.tail logical; if TRUE (default), probabilities are \( P[X \leq x] \), otherwise, \( P[X > x] \).

Details

dskellam returns a value equivalent to

\[
I(2\sqrt{\lambda_1\lambda_2}, \text{abs}(x))(\lambda_1/\lambda_2)^{x/2}\exp(-\lambda_1 - \lambda_2)
\]

where \( I(y, nu) \) is the modified Bessel function of the first kind. The \( \text{abs}(x) \) differs from most Skellam expressions in the literature, but is correct since \( x \) is an integer, resulting in improved portability and (in R versions prior to 2.9) improved accuracy for \( x<0 \). Exponential scaling is used in \text{besselI} to postpone numerical problems. When numerical problems do occur, a saddlepoint approximation is substituted, which typically gives at least 4-figure accuracy. An alternative representation is \( \text{dchisq}(2 \ast \lambda_1, 2 \ast (x + 1), 2 \ast \lambda_2) \ast 2 \) for \( x \geq 0 \), and \( \text{dchisq}(2 \ast \lambda_2, 2 \ast (1 - x), 2 \ast \lambda_1) \ast 2 \) for \( x \leq 0 \); but in R \text{besselI} appears to be more accurately implemented (for very small probabilities) than \text{dchisq}. 

pskellam(x, lambda1, lambda2) returns \( \text{pchisq}(2*\text{lambda2}, -2*x, 2*\text{lambda1}) \) for \( x \leq 0 \) and \( 1 - \text{pchisq}(2*\text{lambda1}, 2*(x+1), 2*\text{lambda2}) \) for \( x > 0 \). When pchisq incorrectly returns 0, a saddlepoint approximation is substituted, which typically gives at least 2-figure accuracy.

The quantile is defined as the smallest value \( x \) such that \( F(x) \geq p \), where \( F \) is the distribution function. For lower.tail=FALSE, the quantile is defined as the largest value \( x \) such that \( F(x, \text{lower.tail}=\text{FALSE}) \leq p \).

rskellam is calculated as \( \text{rpois}(n, \text{lambda1}) - \text{rpois}(n, \text{lambda2}) \).

dskellam.sp and pskellam.sp return saddlepoint approximations to the pmf and cdf. They are called by dskellam and pskellam when results from primary methods are in doubt.

Value

dskellam gives the (log) density, pskellam gives the (log) distribution function, qskellam gives the quantile function, and rskellam generates random deviates. Invalid lambdas will result in return value NaN, with a warning.

Note

The VGAM package (http://lib.stat.cmu.edu/R/CRAN/web/packages/VGAM/index.html) also contains dskellam and rskellam functions, which are syntactically similar; independently developed versions are included here for completeness. Moreover, this dskellam function offers a broader working range, correct handling of cases where at least one rate parameter is zero, enhanced argument checking, and (in R versions prior to 2.9) improved accuracy for \( x < 0 \). If both packages are loaded, get("dskellam", pos="package:skellam") or get("dskellam", pos="package:VGAM") can unambiguously specify which implementation to use.

Author(s)

Jerry W. Lewis

Source

The relation of \( \text{dgamma} \) to the modified Bessel function of the first kind was given by Skellam (1946). The relation of \( \text{pgamma} \) to the noncentral \( \chi^2 \) distribution was given by Johnson (1959). Tables are given by Strackee and van der Gon (1962), which can be used to verify this implementation (cf. direct calculation in the examples below).

qskellam uses the Cornish–Fisher expansion to include skewness and kurtosis corrections to a normal approximation, followed by a search. If getOption("verbose")==TRUE, then qskellam will not use qpois when one of the lambdas is zero, in order to verify that this search algorithm has been implemented properly.

References


Examples

```r
require('skellam')

# one lambda = 0 ~ Poisson
dskellam(0:0,10,5,0)  # dpois(0:0,10,5)
dskellam((0:10),0,5)  # dpois(0:10,5)
pskellam(0:10,5,0,lower.tail=TRUE)  # ppois(0:10,5,lower.tail=TRUE)
pskellam(0:10,5,0,lower.tail=FALSE)  # ppois(0:10,5,lower.tail=FALSE)
pskellam(-(0:10),0,5,lower.tail=FALSE)  # ppois(0:10,5,lower.tail=FALSE)
pskellam(-(0:10),0,5,lower.tail=TRUE)  # ppois(0:10,5,lower.tail=TRUE)

# both lambdas != 0 ~ convolution of Poissons
dskellam(1,0,5,0.75)  # sum(dpois(1+0:10,0.5)*dpois(0:10,0.75))
pskellam(1,0,5,0.75)  # sum(dskellam(-10:1,0,5,0.75))
dskellam(c(-1,1),c(10,12),c(10,12))  # c(0.0697968,0.0697968)
dskellam(c(-1,1),c(10,12),c(10,12),log=TRUE)
  # log(dskellam(c(-1,1),c(10,12),c(10,12)))
dskellam(256,257,1)
  # 0.024829348733183769

# exact result for comparison with saddlepoint
dskellam(-3724,2000,3000)
  # 3.1058145363400105e-308

# exact result for comparison with saddlepoint (still accurate in extreme tail)
pskellam(c(-1,0),c(12,10),c(10,12))  # c(0.2965079,0.7834921)
pskellam(c(-1,0),c(12,10),c(10,12),lower.tail=FALSE)
  # 1-pskellam(c(-1,0),c(12,10),c(10,12))
pskellam(-2:2,8.5,10,25,log.p=TRUE)  # log(pskellam(-2:2,8.5,10,25))
qskellam(c(0.05,0.95),3,4)  # c(-5,3); pskellam(cbind(-6:5,2:3),3,4)
qskellam(c(0.05,0.95),3,0)  # c(1,6); qpois(c(0.05,0.95),3)
rskellam(35,8.5,10,25)
```
Index

*Topic distribution
  Skellam, 2

dskellam (Skellam), 2
pskellam (Skellam), 2
qskellam (Skellam), 2
rskellam (Skellam), 2

Skellam, 2
skellam (Skellam), 2