Package ‘xkcd’

February 27, 2025

Version 0.0.1

Date 2022-02-28

Title xkcd distribution

Author Hongru Zhao <zhao1118@umn.edu>

Maintainer Hongru Zhao <zhao1118@umn.edu>

Depends R (>= 3.1.0)

Description
Density, distribution function, quantile function and random generation for the xkcd distribution.

NeedsCompilation yes

R topics documented:

---

xkcd *The xkcd Distribution*

Description

Density, distribution function, quantile function and random generation for the xkcd distribution with standard deviation equal to *sd*.

Usage

```
dxkcd(x, sd = 1, log.p = FALSE, swap.end.points = FALSE)
pxkcd(q, sd = 1, log.p = FALSE, swap.end.points = FALSE)
qxkcd(p, sd = 1, log.p = FALSE, swap.end.points = FALSE)
rxkcd(n, sd = 1)
```

Arguments

`x, q`	vector of quantiles.
`p`	vector of probabilities.
`n`	number of observations. If length(n) > 1, the length is taken to be the number required.
`sd`	vector of standard deviations.
`log.p`	logical; if TRUE, probabilities p are given as log(p).
`swap.end.points`	logical; if TRUE, probabilities are P[X <= x] otherwise, P[X > x].

Details

This is the distribution introduced by xkcd comic 2118 (see references).

Suppose `f` is the probability density function (PDF) of a normal distribution, and we have a random vector `(X,Y)` uniformly distributed on the region bounded above by the graph of `f` and bounded below by the horizontal axis. Then the marginal distribution of `X` is this normal distribution, the conditional distribution of `Y` given `X` is uniform on the interval from zero to `f(X)`. The marginal distribution of `Y` is the `xkcd` distribution that we want R functions for.

Let `h(y)` be the distance from the mean of `X` to either of the points where `f(x) = y`. Then the distribution function (DF) of `Y` is

$$G(y) = 1 - F(\mu + h(y)) + F(\mu - h(y)) + 2yh(y), 0 < y \leq f(\mu),$$

and the probability density function (PDF) of `Y` simplifies to

$$g(y) = 2h(y), 0 \leq y \leq f(\mu).$$

Value

dxkcd gives the density, *pxkcd* gives the distribution function, *qxkcd* gives the quantile function, and *rxkcd* generates random deviates.

The length of the result is determined by n for rxkcd, 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.

For sd = 0 this gives the limit as sd decreases to 0, a point mass at mu. sd < 0 is an error and returns NaN.

For pxkcd, if swap.end.points = T, log.p = FALSE, and 0< q< 10^-10, to avoid catastrophic cancellation the probability by taloy expansion, it is appoximated as

$$\frac{4}{3}(2 \times sd^3(2\pi)^{1/2})^{1/2}q^{3/2})$$

and if swap.end.points = T, log.p = FALSE, and 0< q< 10^-10, we consider

$$log(\frac{4}{3}(2 \times sd^3(2\pi)^{1/2})^{1/2}) + \frac{3}{2}log(q)$$

Otherwise, pxkcd performs the distribution function above directly.

Author(s)

Hongru Zhao

References

"Normal Distribution." Xkcd, https://xkcd.com/2118/.

Examples

```
require(graphics)

dxkcd(1/sqrt(2*pi)) == 0
dxkcd(0) == Inf
pxkcd(0) == 0
pxkcd(0, log.p = TRUE) == -Inf
pxkcd(1/sqrt(2*pi)) == 1
pxkcd(1/sqrt(2*pi), log.p = TRUE) == 0

## Using "log = TRUE" for an extended range :
par(mfrow = c(2,1))
plot(function(x) dxkcd(x, log = TRUE), 0, 1/sqrt(2*pi),
     main = "log { Xkcd density }")
curve(log(dxkcd(x)), add = TRUE, col = "red", lwd = 2)
mtext("dxkcd(x, log=TRUE)", adj = 0)
mtext("log(dxkcd(x))", col = "red", adj = 1)

plot(function(x) pxkcd(x, log.p = TRUE), 0, 1/sqrt(2*pi),
     main = "log { Xkcd Cumulative }")
curve(log(pxkcd(x)), add = TRUE, col = "red", lwd = 2)
mtext("pxkcd(x, log=TRUE)", adj = 0)
mtext("log(pxkcd(x))", col = "red", adj = 1)
```

Index