Package: paretoconv (via r-universe)

May 11, 2026

Title Calculates the N-Fold Convolution of Two Pareto Distributions

Version 0.0.1.001

Description Calculates the n-fold convolution of two Pareto distributions using the techniques devised by Colin Ramsay.

Depends R (>= 3.0.0)

License GPL-3 + file LICENSE

Encoding UTF-8

LazyData true

Imports cubature, methods, poweRlaw, stats, VGAM

Suggests devtools, ggplot2, knitr, magrittr, rmarkdown, showtext, testthat, tidyr, tippy

Remotes csgillespie/poweRlaw

VignetteBuilder knitr

RoxygenNote 7.3.3

URL https://github.com/mpadge/paretoconv

BugReports https://github.com/mpadge/paretoconv/issues

Config/pak/sysreqs make

Repository https://mpadge.r-universe.dev

Date/Publication 2026-03-23 11:36:11 UTC

RemoteUrl https://github.com/mpadge/paretoconv

RemoteRef HEAD

RemoteSha d8bacdb6c337406eca246b2dcd6063f6dfad2615

Contents

`compare_conv_distributions` *Compares paretoconv with poweRlaw distributions*

Description

This function extends directly from the `poweRlaw` function of the same name, and compares a discrete power law distribution from the `poweRlaw` package with an equivalent convoluted distribution generated by `paretoconv`.

Usage

```
compare_conv_distributions(d1, d2)
```

Arguments

`d1`	Discrete power law distribution of class `displ` from the package `poweRlaw`.
`d2`	Equivalent convoluted values for the probability density function obtained from `paretoconv (..., cdf = FALSE)`

Examples

```
## Not run:
data ('native_american', package = 'poweRlaw')
m1 <- displ$new(native_american$Cas)
m1$setXmin (3)
m1$setPars (estimate_pars(m1))
x <- sort (unique (native_american$Cas))
y <- paretoconv (x = x, a = m1$getPars (), x0 = 3, n = 2,
cdf = FALSE, quiet = FALSE)
compare_conv_distributions (m1, y)

## End(Not run)
```

get_asymp_limit *Get limit of asymptotic convergence*

Description

Get limit of asymptotic convergence

Usage

```
get_asymp_limit(a, n, x0, tol = 0.05, quiet = FALSE)
```

Arguments

a	The primary shape parameter of the Pareto distribution (single value only)
n	Number of convolutions (single value only)
x0	Lower cut-off point of classical heavy-tailed distribution (generally obtained emprically with the poweRlaw package).
tol	Convergence tolerance (values around 0.05 are typically sufficient; values < 0.01 will generally not work).
quiet	If FALSE, issue progress messages

Value

Single integer value of limit beyond which PDF may be approximated by its asymptotic power law

pareto_optimise *Locally optimal fitting of convoluted Pareto distribution to observed data*

Description

Locally optimal fitting of convoluted Pareto distribution to observed data

Usage

```
pareto_optimise(m, x0 = 1, n = 1, check_non_conv = TRUE, quiet = TRUE)
```

Arguments

m	A poweRlaw::displ object containing data to be modelled
x0	Initial guess for lower limit of Pareto distribution
n	Initial guess for number of convolutions
check_non_conv	If TRUE first checks whether a non-convoluted model may be optimal
quiet	If FALSE, progress information is dumped to screen

Value

Position of the local optimum as quantified by `x0` and `n`, along with associated Kolmogorow-Smirnov statistic quantifying maximal distance from convoluted Pareto Cumulative Distribution Function (CDF) and empirical CDF of model `m`.

Note

This function only finds local optima - it is up to the user to ensure that the given start values are near the global optimum. Values of `n = 0` are NOT searched.

`paretoconv` *paretoconv.*

Description

Calculates the n-fold convolution of two Pareto distributions using the techniques devised by Colin Ramsay.

Convolutes multiple Pareto distributions following

- For integer shape parameters: 'The Distribution of Sums of Certain I.I.D. Pareto Variates' by Colin Ramsay (Communications in Statistics - Theory and Methods 35:395-405, 2006);
- For non-integer shape parameters: 'The Distribution of Sums of I.I.D. Pareto Random Variables with Arbitrary Shape Parameter' by Colin Ramsay (Communications in Statistics - Theory and Methods 37:2177-2184, 2008).

Usage

```
paretoconv(x, a, n, x0 = 1, cdf = FALSE, asymp = TRUE, quiet = TRUE)
```

Arguments

`x`	value of independent variable (may be a vector)
`a`	The primary shape parameter of the Pareto distribution (single value only)
`n`	Number of convolutions (single value only)
`x0`	Lower cut-off point of classical heavy-tailed distribution (generally obtained emprically with the poweRlaw package).
`cdf`	If TRUE, returns the cumulative distribution function, otherwise returns the probability density function.
`asymp`	If TRUE and `x` is a vector of 10 or more elements, asymptotic convergence is checked for large `x`, with values beyond convergence replaced with directly power-law values
`quiet`	If FALSE, issue progress messages

Value

Value for the CDF or PDF from the convolution of two Pareto distributions of shape a at the value x.

Note

The Pareto distribution may be defined as f(x)=(a/b)(b/x)^(a-1), where a and b are the primary and secondary shape parameters, respectively. It presumed here without loss of generality that b=1 and thus f(x)=a x^(1-a). Convolution of multiple distritions (that is, n>0) are NOT normalised, so CDFs to not sum to unity, and PDFs do not integrate to unity.

Author(s)

Mark Padgham

See Also

Useful links:

- https://github.com/mpadge/paretoconv
- Report bugs at https://github.com/mpadge/paretoconv/issues

Examples

```
paretoconv (x=1:10, a=2, n=0)
```

`pdf_integral` *Calculates integral of probability density functions for convoluted Pareto distributions*

Description

Calculates integral of probability density functions for convoluted Pareto distributions

Usage

```
pdf_integral(a, x0 = 1, n = 0, discrete = TRUE, xmin = 0, quiet = TRUE)
```

Arguments

a	Shape parameter of Pareto disitribution
x0	Lower cut-off for classic Pareto distribution
n	Number of convolutions
discrete	Calculate integral for discrete (TRUE) or continuous (FALSE) probability density function
xmin	Lower limit of integral (generally 0, 1, or x0)
quiet	If FALSE, progress is displayed as screen output

Note

Values for discrete integrals differ from values from continuous integrals. This function can also take a long time to execute, particularly for continuous integrals.

Examples

```
pdf_integral (a = 2.1, x0 = 5, n = 0)
```

`pparetoconv` *Estimate probability of observed KS statistic from synthetic models*

Description

Estimate probability of observed KS statistic from synthetic models

Usage

```
pparetoconv(m, x0, n, neach = 10, quiet = TRUE)
```

Arguments

`m`	A `poweRlaw::displ` object containing data to be modelled
`x0`	Initial guess for lower limit of Pareto distribution
`n`	Initial guess for number of convolutions
`neach`	Number of test statistics to be generated for each different model
`quiet`	If FALSE, display progress messages on screen

Value

Single value representing probability of observing given value of `ks`

`rplconv` *Generate random deviates for power laws*

Description

Generate random deviates for power laws

Usage

```
rplconv(n, x0, alpha, discrete_max = 10000)
```

Arguments

n	Number of deviates
x0	Traditional cutoff parameter of Pareto distribution
alpha	Primary shape parameter of Pareto distribution
discrete_max	Upper limit for generation (see poweRlaw::rpldis for details)

Value

Vector of n random deviates

Note

This is directly cribbed from poweRlaw::rpldis, modified only to generate random deviates for all values including 0<x<xmin.

Index