Package 'LNPar'

May 7, 2026

Title Estimation and Testing for a Lognormal-Pareto Mixture

Version 1.1.3

Description Estimates a lognormal-Pareto mixture by means of the Expectation-Conditional-Maximization-Either algorithm and by maximizing the profile likelihood function. A likelihood ratio test for discriminating between lognormal and Pareto tail is also implemented. See Bee, M. (2022) <doi:10.1007/s11634-022-00497-4>.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.3

Depends R (>= 4.0.0)

LazyData true

RdMacros Rdpack

Imports Rdpack, parallel, stats

NeedsCompilation no

Author Marco Bee [aut, cre] (ORCID: <https://orcid.org/0000-0002-9579-3650>)

Maintainer Marco Bee <marco.bee@unitn.it>

Repository CRAN

Date/Publication 2026-01-20 06:20:03 UTC

Contents

`dLnormParMix` *density of a mixture of a lognormal and a Pareto r.v.*

Description

This function computes the density of a mixture of a lognormal and a Pareto r.v.

Usage

```
dLnormParMix(x, pi, mu, sigma, xmin, alpha)
```

Arguments

`x`	non-negative numerical vector: values where the density has to be evaluated.
`pi`	scalar, $0 < p < 1$: mixing weight.
`mu`	scalar: expected value of the lognormal distribution on the log scale.
`sigma`	positive scalar: standard deviation of the lognormal distribution on the log scale.
`xmin`	positive scalar: threshold.
`alpha`	positive scalar: Pareto shape parameter.

Value

Density of the lognormal-Pareto distribution evaluated at x.

Examples

```
mixDens <- dLnormParMix(5,.5,0,1,4,1.5)
```

dpareto *density of a Pareto r.v.*

Description

This function evaluates the density of a Pareto r.v.s

Usage

```
dpareto(x, xmin, alpha)
```

Arguments

x	numerical vector (>=xmin): values where the density has to be evaluated.
xmin	positive scalar: Pareto scale parameter.
alpha	positive scalar: Pareto shape parameter.

Value

Density of the Pareto distribution evaluated at x.

Examples

```
parDens <- dpareto(5,4,2)
```

ECMEBoot *Bootstrap standard errors for the MLEs of a lognormal-Pareto mixture*

Description

This function draws a bootstrap sample and uses it to estimate the parameters of a lognormal-Pareto mixture distribution. Since this is typically called by LPfitEM, see the help of LPfitEM for examples.

Usage

```
ECMEBoot(x, y, eps, maxiter)
```

Arguments

x	list: sequence of integers 1,...,K, where K is the mumber of datasets. Set x = 1 in case of a single dataset.
y	numerical vector: observed sample.
eps	non-negative scalar: starting value of the log-expectation of the lognormal distribution on the log scale.
maxiter	non-negative integer: maximum number of iterations of the ECME algorithm.

Details

At each bootstrap replication, the mixture is estimated via the ECME algorithm. The function is typically called by LPfitEM.

Value

Estimated parameters obtained from a bootstrap sample.

`ll_lnormparmix` *Log-likelihood with respect to xmin*

Description

This function evaluates the log-likelihood function with respect to xmin for a mixture of a lognormal and a Pareto r.v., assuming to know the numerical values of all the other parameters.

Usage

```
ll_lnormparmix(x, pi, mu, sigma, alpha, y)
```

Arguments

`x`	positive scalar: value of xmin where the function is evaluated.
`pi`	scalar, 0 < pi < 1: mixing weight.
`mu`	scalar: expected value of the lognormal distribution on the log scale.
`sigma`	positive scalar: standard deviation of the lognormal distribution on the log scale.
`alpha`	non-negative scalar: Pareto shape parameter.
`y`	(nx1) vector: random sample from the mixture.

Value

ll numerical value of the log-likelihood function.

Examples

```
y <- rLnormParMix(100,.5,0,1,4,1.5)
llMix <- ll_lnormparmix(5,.5,0,1,4,y)
```

LPfitEM *Estimating a lognormal-Pareto mixture via the ECME algorithm*

Description

This function fits a lognormal-Pareto mixture by means of the ECME algorithm.

Usage

```
LPfitEM(y, eps, maxiter, qxmin0 = 0.5, nboot = 0)
```

Arguments

y	numerical vector: random sample from the mixture.
eps	non-negative scalar: tolerance for the stopping rule.
maxiter	non-negative integer: maximum number of iterations of the ECME algorithm.
qxmin0	scalar, $0 < \text{qxmin0} < 1$: quantile level used for determining the starting value of xmin. Defaults to 0.5.
nboot	non-negative integer: number of bootstrap replications used for estimating the standard errors. If omitted, no standard errors are computed.

Details

Estimation of a lognormal-Pareto mixture via the ECME algorithm. Standard errors are computed via non-parametric bootstrap.

Value

A list with the following elements:

pars: estimated parameters (p, mu, sigma, xmin, alpha).

loglik: maximized log-likelihood.

thRank: estimated rank of xmin.

niter: number of iterations.

npareto: estimated number of Pareto observations.

postProb: matrix of posterior probabilities.

bootEst: matrix of estimated parameters at each bootstrap replication.

bootstd: bootstrap standard errors of the estimators.

Examples

```
ysim <- rLnormParMix(100,.9,0,1,5,1)
mixFit <- LPfitEM(ysim,eps=1e-10,maxiter=1000,nboot=0)
```

`LPfitProf` *Profile likelihood estimation of a lognormal-Pareto mixture*

Description

This function fits a lognormal-Pareto mixture by maximizing the profile log-likelihood.

Usage

```
LPfitProf(y, minRank, nboot)
```

Arguments

`y`	numerical vector: random sample from the mixture.
`minRank`	integer: minimum possible rank of the threshold.
`nboot`	number of bootstrap replications used for estimating the standard errors. If omitted, no standard errors are computed.

Details

Estimation is implemented as in Bee (2022). As of standard errors, at each bootstrap replication the mixture is estimated with thresholds equal to ys(minRank), ys(minRank+1),..., ys(n), where n is the sample size and ys is the sample sorted in ascending order. The latter procedure is implemented via parallel computing. If the algorithm does not converge in 1000 iterations, a message is displayed.

Value

A list with the following elements:

xmin: estimated threshold.

prior: estimated mixing weight.

postProb: matrix of posterior probabilities.

alpha: estimated Pareto shape parameter.

mu: estimated expectation of the lognormal distribution on the lognormal scale.

sigma: estimated standard deviation of the lognormal distribution on the lognormal scale.

loglik: maximized log-likelihood.

nit: number of iterations.

npareto: estimated number of Pareto observations.

bootstd: bootstrap standard errors of the estimators.

References

Bee M (2024). "On discriminating between lognormal and Pareto tail: an unsupervised mixture-based approach." *Advances in Data Analysis and Classification*, **18**, 251-269.

Examples

```
mixFit <- LPfitProf(TN2016,90,0)
```

`LPtest` *Profile-based testing for a Pareto tail*

Description

This function draws a bootstrap sample from the null (lognormal) distribution and computes the test for the null hypothesis of a pure lognormal distribution versus the alternative of a lognormal-Pareto mixture, where the parameters of the latter are estimated via maximum profile likelihood. To be only called from ParallelTest. Estimation unde rthe alternative is perfromed

Usage

```
LPtest(x, n, muNull, sigmaNull, minRank)
```

Arguments

`x`	list: sequence of integers 1,...,K, where K is the mumber of datasets. Set x = 1 in case of a single dataset.
`n`	sample size.
`muNull`	lognormal expected value under the null hypothesis.
`sigmaNull`	lognormal standard deviation under the null hypothesis.
`minRank`	minimum possible rank of the threshold.

Value

A list with the following elements:

LR: observed value of the llr test.

References

Bee M (2024). "On discriminating between lognormal and Pareto tail: an unsupervised mixture-based approach." *Advances in Data Analysis and Classification*, **18**, 251-269.

Examples

```
n = 100
muNull = mean(log(TN2016))
sigmaNull = sd(log(TN2016))
minRank = 90
res = LPtest(1,n,muNull,sigmaNull,minRank)
```

LPtestEM *ECME-based testing for a Pareto tail*

Description

This function draws a bootstrap sample from the null (lognormal) distribution and computes the test for the null hypothesis of a pure lognormal distribution versus the alternative of a lognormal-Pareto mixture, where the parameters of the latter are estimated by means of the ECME algorithm. To be only called from ParallelTestEM.

Usage

```
LPtestEM(x, n, muNull, sigmaNull)
```

Arguments

x	list: sequence of integers 1,...,K, where K is the mumber of datasets. Set x = 1 in case of a single dataset.
n	sample size.
muNull	log-expectation value under the null hypothesis.
sigmaNull	log-standard deviation under the null hypothesis.

Value

A list with the following elements:

LR: observed value of the llr test.

Examples

```
n = 100
muNull = mean(log(TN2016))
sigmaNull = sd(log(TN2016))
res = LPtestEM(1,n,muNull,sigmaNull)
```

ParallelTest *Profile-based testing for a Pareto tail*

Description

This function computes the bootstrap test for the null hypothesis of a pure lognormal distribution versus the alternative of a lognormal-Pareto mixture, where the parameters of the latter are estimated via maximum profile likelihood. Implemented via parallel computing.

Usage

```
ParallelTest(nboot, y, obsTest, minRank)
```

Arguments

nboot	number of bootstrap replications.
y	observed data.
obsTest	value of the test statistics computed with the data under analysis.
minRank	minimum possible rank of the threshold.

Value

A list with the following elements:

LR: nboot simulated values of the llr test under the null hypothesis.

pval: p-value of the test.

Examples

```
minRank = 90
mixFit <- LPfitProf(TN2016,minRank,0)
ell1 <- mixFit$loglik
estNull <- c(mean(log(TN2016)),sd(log(TN2016)))
ellNull <- sum(log(dlnorm(TN2016,estNull[1],estNull[2])))
obsTest <- 2*(ell1-ellNull)
nboot = 2
TestRes = ParallelTest(nboot,TN2016,obsTest,minRank)
```

ParallelTestEM *ECME-based testing for a Pareto tail*

Description

This function computes the bootstrap test for the null hypothesis of a pure lognormal distribution versus the alternative of a lognormal-Pareto mixture, where the parameters of the latter are estimated by means of the ECME algorithm. likelihood. Implemented via parallel computing.

Usage

```
ParallelTestEM(nboot, y, obsTest)
```

Arguments

nboot	number of bootstrap replications.
y	observed data.
obsTest	value of the test statistics computed with the data under analysis.

Value

A list with the following elements:

LR: nboot simulated values of the llr test under the null hypothesis.

pval: p-value of the test.

Examples

```
minRank = 90
mixFit <- LPfitEM(TN2016,1e-12,1000)
ell1 <- mixFit$loglik
estNull <- c(mean(log(TN2016)),sd(log(TN2016)))
ellNull <- sum(log(dlnorm(TN2016,estNull[1],estNull[2])))
obsTest <- 2*(ell1-ellNull)
nboot = 2
TestRes = ParallelTestEM(nboot,TN2016,obsTest)
```

`par_logn_mix_known` *Estimate the parameters of a lognormal-Pareto density, assuming a known threshold*

Description

This function estimates the parameters of a Pareto and a lognormal density, assuming a known threshold.

Usage

```
par_logn_mix_known(y, prior1, th, alpha, mu, sigma)
```

Arguments

`y`	non-negative numerical vector: random sample from the mixture.
`prior1`	scalar (0<prior1<1): starting value of the prior probability.
`th`	positive scalar: threshold.
`alpha`	non-negative scalar: starting value of the Pareto shape parameter.
`mu`	scalar: starting value of the lognormal parameter mu.
`sigma`	positive scalar: starting value of the lognormal parameter sigma.

Value

A list with the following elements:

xmin: estimated threshold.

prior: estimated mixing weight.

post: matrix of posterior probabilities.

alpha: estimated Pareto shape parameter.

mu: estimated expectation of the lognormal distribution on the lognormal scale.

sigma: estimated standard deviation of the lognormal distribution on the lognormal scale.

loglik: maximized log-likelihood.

nit: number of iterations.

Examples

```
mixFit <- par_logn_mix_known(TN2016, .5, 4700, 3, 7, 1.2)
```

`ProfBoot` *Bootstrap standard errors for the estimators of a lognormal-Pareto mixture*

Description

This function draws a bootstrap sample and uses it to estimate the parameters of a lognormal-Pareto mixture distribution. Since this is typically called by LPfitProf, see the help of LPfitProf for examples.

Usage

```
ProfBoot(x, y, minRank, p0, alpha0, mu0, Psi0)
```

Arguments

`x`	list: sequence of integers 1,...,K, where K is the mumber of datasets. Set x = 1 in case of a single dataset.
`y`	numerical vector: observed sample.
`minRank`	positive integer: minimum possible rank of the threshold.
`p0`	(0<p0<1): starting value of the mixing weight.
`alpha0`	non-negative scalar: starting value of the Pareto shape parameter.
`mu0`	scalar: starting value of the log-expectation of the lognormal distribution on the log scale.
`Psi0`	non-negative scalar: starting value of the log-variance of the lognormal distribution on the log scale.

Details

At each bootstrap replication, the mixture is estimated with thresholds equal to ys(minRank), ys(minRank+1),..., ys(n), where n is the sample size and ys is the sample in ascending order. The function is typically called by LPfitProf.

Value

Estimated parameters obtained from a bootstrap sample.

References

Bee, M. (2022), "On discriminating between lognormal and Pareto tail: a mixture-based approach", Advances in Data Analysis and Classification, https://doi.org/10.1007/s11634-022-00497-4

rLnormParMix — *Random number simulation for a mixture of a lognormal and a Pareto r.v.*

Description

This function simulates random numbers for a mixture of a lognormal and a Pareto r.v.

Usage

```
rLnormParMix(n, p, mu, sigma, xmin, alpha)
```

Arguments

n	positive integer: number of simulated random numbers.
p	scalar, $0 < p < 1$: mixing weight.
mu	scalar: expected value of the lognormal distribution on the log scale.
sigma	positive scalar: standard deviation of the lognormal distribution on the log scale.
xmin	positive scalar: threshold.
alpha	non-negative scalar: Pareto shape parameter.

Value

n iid random numbers from the lognormal-Pareto distribution.

Examples

```
ySim <- rLnormParMix(100,.5,0,1,4,1.5)
```

rpareto — *Random number generation for a Pareto r.v.*

Description

This function simulates random numbers for a Pareto r.v.

Usage

```
rpareto(n, xmin, alpha)
```

Arguments

n	positive integer: number of simulated random numbers.
xmin	positive scalar: Pareto scale parameter.
alpha	non-negative scalar: Pareto shape parameter.

Value

n iid random numbers from the Pareto distribution.

Examples

```
ySim <- rpareto(5,4,1.5)
```

TN2016 *Number of employees in year 2016 in all the firms of the Trento district*

Description

A dataset containing the number of employees in year 2016 in all the firms of the Trento district in Northern Italy.

Usage

```
TN2016
```

Format

A numerical vector with 183 rows and 1 column.

Source

https://dati.trentino.it/

Index