Package 'ParetoPosStable'

September 2, 2015

Type Package

Title Computing, Fitting and Validating the PPS Distribution

Version 1.1

Date 2015-09-02

Maintainer Antonio Jose Saez-Castillo <ajsaez@ujaen.es>

Depends R (>= 3.0.0)

Imports graphics, grDevices, stats, ADGofTest, lmom, foreach, doParallel, parallel

Description Statistical functions to describe a Pareto Positive Stable (PPS) distribution and fit it to real data. Graphical and statistical tools to validate the fits are included.

License GPL-2 | GPL-3

LazyData true

NeedsCompilation no

Author Antonio Jose Saez-Castillo [aut, cre],
Faustino Prieto [aut],
Jose Maria Sarabia [aut]

Repository CRAN

Date/Publication 2015-09-02 15:52:04

R topics documented:

`coef.PPSfit` *Parameter estimates in a PPSfit Object*

Description

`coef` returns the parameter estimates in a `PPSfit` Object

Usage

```
## S3 method for class 'PPSfit'
coef(object, ...)
```

Arguments

`object`	A `PPSfit` object, typically from `PPS.fit()`.
`...`	Other arguments.

Value

A list with the parameter estimates.

See Also

`PPS.fit`.

Examples

```
x <- rPPS(50, 1.2, 100, 2.3)
fit <- PPS.fit(x)
coef(fit)
```

forbes400 *Forbes 400 list. The Richest People in America in 2012*

Description

The Forbes 400 or 400 Richest Americans is a list published by Forbes magazine of the wealthiest 400 Americans, ranked by net worth. The dataset presents results about 2012.

Format

A data frame with 400 observations on the following 2 variables.

- `Name`members of the list names.
- `NetWorth`net worth in 2012 in $ billion.

GoF *Goodness of fit tests for the Pareto Positive Stable (PPS) distribution*

Description

Kolmogorov-Smirnov, Anderson-Darling and PPS goodness of fit tests to validate a PPS fit (typically from `PPS.fit()`).

Usage

```
GoF(PPSfit, k = 2000, parallel = TRUE, ncores = 2, ...)
```

Arguments

`PPSfit`	A `PPSfit` Object.
`k`	The number of iterations in the bootstrap procedure to approximate the p-values.
`parallel`	A logical argument specifying if parallelization is allowed in the bootstrap iteration procedure.
`ncores`	is the number of cores that we use if parallel is TRUE.
`...`	Other arguments.

Details

It returns the Kolmogorov-Smirnov, the Anderson-Darling tests and a specific test for PPS distributions. p-values are approximated by a bootstrap procedure. The specific goodness of fit test for PPS distributions is based on the linearity of the survival function vs. the scaled observations in a double log-log scale (see Sarabia and Prieto, 2009).

Value

A list with the values of the tests statistics and the approximated p-values.

References

Sarabia, J.M and Prieto, F. (2009). The Pareto-positive stable distribution: A new descriptive model for city size data, *Physica A: Statistical Mechanics and its Applications*, **388**(19), 4179-4191.

See Also

PPS.fit, plot.PPSfit

Examples

```
x <- rPPS(50, 1.2, 100, 2.3)
fit <- PPS.fit(x)
GoF(fit, k = 50)
```

---

logLik.PPSfit *Log-likelihood value of a PPSfit Object*

Description

It returns the log-likelihood value of a `PPSfit` Object

Usage

```
## S3 method for class 'PPSfit'
logLik(object, ...)
```

Arguments

`object`	A `PPSfit` Object.
`...`	Other arguments.

Value

The log-likelihood.

References

Sarabia, J.M and Prieto, F. (2009). The Pareto-positive stable distribution: A new descriptive model for city size data, *Physica A: Statistical Mechanics and its Applications*, **388**(19), 4179-4191.

See Also

PPS.fit

Examples

```
x <- rPPS(50, 1.2, 100, 2.3)
fit <- PPS.fit(x)
logLik(fit)
```

---

pareto.fit *Fitting a Pareto distribution*

Description

It is an auxiliar function for fitting a Pareto distribution as a particular case of a Pareto Positive Stable distribution, allowing the scale parameter to be held fixed if desired.

Usage

```
pareto.fit(x, estim.method = "MLE", sigma = NULL, start, ...)
```

Arguments

`x`	The vector of observations.
`estim.method`	The estimation method, "MLE" or "OLS"
`sigma`	The value of the scale parameter, if it is known; if the value is `NULL`, the parameter is estimated.
`start`	Unused argument from `PPS.fit`.
`...`	Other arguments.

Details

This function is called by `PPS.fit()` when `Pareto` argument is `TRUE`.

Value

A `PPSfit` Object.

References

Sarabia, J.M and Prieto, F. (2009). The Pareto-positive stable distribution: A new descriptive model for city size data, *Physica A: Statistical Mechanics and its Applications*, **388**(19), 4179-4191.

See Also

`PPS.fit`, `coef.PPSfit`, `print.PPSfit`, `plot.PPSfit`, `GoF`

ParetoPosStable — *Computing, Fitting and Validating the PPS Distribution*

Description

Statistical functions to describe a Pareto Positive Stable (PPS) distribution and fit it to real data. Graphical and statistical tools to validate the fits are included.

Details

The main function you're likely to need from **ParetoPosStable** is `PPS.fit`, in order to obtain a PPS fit from data. Validation can be obtain with `GoF` and `plot`.

References

Sarabia, J.M and Prieto, F. (2009). The Pareto-positive stable distribution: A new descriptive model for city size data, *Physica A: Statistical Mechanics and its Applications*, **388**(19), 4179-4191.

plot.PPSfit — *Plots to validate a Pareto Positive Stable (PPS) fit*

Description

Plots to validate a PPS fit (typically from `PPS.fit()`) with different comparisons between empirical and theoretical functions.

Usage

```
## S3 method for class 'PPSfit'
plot(x, which = 1:4, ask = prod(par("mfcol")) <
  length(which) && dev.interactive(), ylim, breaks, ...)
```

Arguments

`x`	a `PPSfit` Object.
`which`	values from 1 to 4 indicating the type of plot.
`ask`	an argument to control the plot window.
`ylim`	optional argument to control the y limits of the histogram. It is included to prevent non-desired scales on the y-axis.
`breaks`	optional argument to control the breakpoints of the histogram. See `hist` help for the details. It is included to prevent non-desired scales on the y-axis.
`...`	other arguments.

Details

The plots return:

1. The histogram of the observations and the fitted PPS density (`which = 1`). Optional `ylim` and `breaks` arguments are provided to prevent frequent imbalances between density and histogram scales in real data: they work as the analogue arguments of the default `hist` function. 2. The empirical distribution function of data and the cumulative distribution function of the fitted model (`which = 2`). 3. A rank-size plot in log-log scale to check the Pareto or power-law behaviour of data (`which = 3`). In the X-axis the log of the observations appears; in the Y-axis, the log of the empirical survival function. If the scatter-plot is around a straight line, then the observations exhibit a power law behaviour. The plot also includes the curve with the theoretical survival function of the model specified in the first argument class `PPSfit`: only when `nu` is 1, that curve is going to be a straight line. 4. A plot in a double log-log scale to check the adequacy of data to the PPS model (`which = 4`). On one hand, the X-axis shows the double log of the observations divided by the scale parameter, while the Y-axis shows the log of minus the log of the empirical survival function. On the other hand, the straight line determined by the linear relation between the survival function and the scaled data in a double log-log scale, in relation to the argument class `PPSfit` is added. The proximity of the points in the scatter-plot to that straight line is an evidence in favour of a PPS behaviour of data.

References

Sarabia, J.M and Prieto, F. (2009). The Pareto-positive stable distribution: A new descriptive model for city size data, *Physica A: Statistical Mechanics and its Applications*, **388**(19), 4179-4191.

See Also

`PPS.fit`

Examples

```
data(forbes400)
fit <- PPS.fit(forbes400$NetWorth)
par(mfrow=c(2,2))
plot(fit)
dev.off()
plot(fit, which = 1, breaks = seq(0, 60, length.out = 60))
```

PPS *The Pareto Positive Stable (PPS) distribution*

Description

Density, distribution function, hazard function, quantile function and random generation for the Pareto Positive Stable (PPS) distribution with parameters `lam`, `sc` and `v`.

Usage

```
dPPS(x, lam, sc, v, log = FALSE)

hPPS(x, lam, sc, v)

pPPS(x, lam, sc, v, lower.tail = TRUE, log.p = FALSE)

qPPS(p, lam, sc, v, lower.tail = TRUE, log.p = FALSE)

rPPS(n, lam, sc, v)
```

Arguments

x	vector of quantiles.
lam	vector of (non-negative) first shape parameters.
sc	vector of (non-negative) scale parameters.
v	vector of (non-negative) second shape parameters.
log	logical; if TRUE, probabilities/densities p are returned as $log(p)$.
lower.tail	logical; if TRUE (default), probabilities are $P[X \leq x]$, otherwise, $P[X > x]$.
log.p	logical; if TRUE, probabilities/densities p are returned as $log(p)$.
p	vector of probabilities.
n	number of random values to return.

Details

The PPS distribution has density

$$f(x) = \lambda\nu[log(x/\sigma)]^(\nu - 1)exp(-\lambda[log(x/\sigma)]^{\nu})/x,$$

cumulative distribution function

$$F(x) = 1 - exp(-\lambda[log(x/\sigma)^{\nu}]),$$

quantile function

$$Q(p) = \sigma exp([-(1/\lambda)log(1-p)]^(1/\nu))$$

and hazard function

$$\lambda\nu(log(x/\sigma))^(\nu - 1)x^(- 1).$$

See Sarabia and Prieto (2009) for the details about the numbers random generation.

Value

dPPS gives the (log) density, pPPS gives the (log) distribution function, qPPS gives the quantile function, and rpois generates random samples. Invalid parameters will result in return value NaN, with a warning. The length of the result is determined by n for rPPS, and is the common length of the numerical arguments for the other functions.

References

Sarabia, J.M and Prieto, F. (2009). The Pareto-positive stable distribution: A new descriptive model for city size data, *Physica A: Statistical Mechanics and its Applications*, **388**(19), 4179-4191.

Examples

```
print(x <- sort(rPPS(10, 1.2, 100, 2.3)))
dPPS(x, 1.2, 100, 2.3)
pPPS(x, 1.2, 100, 2.3)
qPPS(pPPS(x, 1.2, 100, 2.3), 1.2, 100, 2.3)
hPPS(x, 1.2, 100, 2.3)
```

PPS.fit *Fitting the Pareto Positive Stable (PPS) distribution*

Description

`PPS.fit()` returns the fit of a PPS distribution to real data, allowing the scale parameter to be held fixed if desired.

Usage

```
PPS.fit(x, estim.method = "MLE", sigma = NULL, start, Pareto = FALSE, ...)
```

Arguments

`x`	a vector of observations
`estim.method`	the method of parameter estimation. It may be "MLE", "iMLE", "OLS", or "LMOM".
`sigma`	the value of the scale parameter, if it is known; if the value is `NULL`, the parameter is estimated.
`start`	a list with the initial values of the parameters for some of the estimation methods.
`Pareto`	a logical argument to constrain the PPS fit to a Pareto fit when the value is `TRUE`.
`...`	other arguments.

Details

The maximum likelihood method implemented by the direct optimization of the log-likelihood is given by `estim.method = "MLE"`. The numerical algorithm to search the optimum is the "Nelder-Mead" method implemented in the `optim` function, considering as initial values those given in the `start` argument or, if it is missing, those provided by the OLS method.

A different approximation of the maximum likelihood estimates is given by `estim method = "iMLE"`; it is an iterative methodology where `optimize()` function provides the optimum scale parameter value, while the `uniroot()` function solve normal equations for that given scale parameter.

The regression estimates ("OLS") searchs an optimum scale value (in a OLS criterion) by the optimize() function. Then the rest of the parameters are estimated also by OLS, as appears in Sarabia and Prieto (2009).

In the L-moments method ("LMOM") estimates are obtained searching parameters that equal the first three sample and theoretical L-moments by means of the "Nelder-Mead" algorithm implemented in optim(); the initial values are given in the start argument or, if it is missing, provided by the "iMLE".

Value

A PPSfit Object, a list with

- estimateparameter estimates.
- loglikthe log-likelihood value.
- nthe number of observations.
- obsthe observations.
- obsNamethe name of the variable with the observations.
- estim.methodthe method of parameter estimation.

When this last value is "LMOM" the function also returns details about the convergence of the numerical method involved (convergence value).

References

Sarabia, J.M and Prieto, F. (2009). The Pareto-positive stable distribution: A new descriptive model for city size data, *Physica A: Statistical Mechanics and its Applications*, **388**(19), 4179-4191. Hosking, J. R. M. (1990). L-moments: analysis and estimation of distributions using linear combinations of order statistics. *Journal of the Royal Statistical Society, Series B*, **52**, 105-124.

See Also

coef.PPSfit, print.PPSfit, plot.PPSfit, GoF

Examples

```
data(turkey)
fit <- PPS.fit(turkey$Pop2000)
print(fit)
coef(fit)
se(fit, k = 100, parallel = FALSE)
logLik(fit)
par(mfrow=c(2,2))
plot(fit)
GoF(fit, k = 100, parallel = FALSE)
```

`print.PPSfit` *Printing a PPSfit Object*

Description

It prints its argument (typically from `PPS.fit()`), returning some of the most important aspects.

Usage

```
## S3 method for class 'PPSfit'
print(x, digits = max(3, getOption("digits") - 3), ...)
```

Arguments

`x`	a `PPSfit` object.
`digits`	the number of digits to be printed.
`...`	other arguments.

References

Sarabia, J.M and Prieto, F. (2009). The Pareto-positive stable distribution: A new descriptive model for city size data, *Physica A: Statistical Mechanics and its Applications*, **388**(19), 4179-4191.

See Also

`PPS.fit`

Examples

```
x <- rPPS(50, 1.2, 100, 2.3)
fit <- PPS.fit(x)
print(fit)
```

`se` *Approximated standard errors of Pareto Positive Stable (PPS) parameter estimates*

Description

It approximates the stantard errors of PPS parameter estimates by bootstrapping.

Usage

```
se(PPSfit, k = 2000, parallel = TRUE, ncores = 2, ...)
```

Arguments

PPSfit	a PPSfit Object, typically from PPS.fit().
k	the number of steps in the bootstrapping procedure.
parallel	A logical argument specifying if parallelization is allowed in the bootstrapping procedure.
ncores	is the number of cores that we use if parallel is TRUE
...	other arguments.

Details

The function simulates k samples from the model given in the PPSfit argument, fits them with the same method of estimation and uses the parameter estimates to approximate the standard errors.

Value

A list with the standard errors.

References

Sarabia, J.M and Prieto, F. (2009). The Pareto-positive stable distribution: A new descriptive model for city size data, *Physica A: Statistical Mechanics and its Applications*, **388**(19), 4179-4191.

See Also

PPS.fit

Examples

```
x <- rPPS(50, 1.2, 100, 2.3)
fit <- PPS.fit(x)
coef(fit)
se(fit, k = 50)
```

---

turkey *Population of Turkish cities and towns*

Description

Census population of Turkish cities and towns of more than 20,000 inhabitants in 1990 and 2000.

Format

A data frame with 280 observations on the following 4 variables.

- Namecities and towns names.
- Adm.abbreviated province name of cities and towns.
- Pop1990the population in 1990.
- Pop2000the population in 2000. #'

Source

http://www.citypopulation.de/Turkey-C20.html

Index