Package ‘zipfextR’

October 31, 2017

Type Package

Title Zipf Extended Distributions

Version 0.5.0

Author Ariel Duarte-López and Marta Pérez-Casany

Maintainer Ariel Duarte-López <aduarte@ac.upc.edu>

Description Three extensions of the Zipf model.

License GPL-3

Depends R (>= 2.0.1)

Imports VGAM (>= 0.9.8), tolerance(>= 1.2.0)

Encoding UTF-8

LazyData true

URL https://github.com/ardlop/zipfextR

BugReports https://github.com/ardlop/zipfextR/issues

RoxygenNote 6.0.1

Suggests testthat

R topics documented:

moezipf *The Marshal-Olkin Extended Zipf Distribution (MOEZipf).*

Description

Probability Mass Function, Cumulative Function, Quantile Function and Random Generation of the MOEZipf distribution with parameter α and β.

Usage

```
dmoezipf(x, alpha, beta, log = FALSE)

pmoezipf(q, alpha, beta, log.p = FALSE, lower.tail = TRUE)

qmoezipf(p, alpha, beta, log.p = FALSE, lower.tail = TRUE)

rmoezipf(n, alpha, beta)
```

Arguments

`x, q`	Vector of positive integer values.
`alpha`	Value of the α parameter ($\alpha > 1$).
`beta`	Value of the β parameter ($\beta > 0$).
`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, $P[X > x]$.
`p`	Vector of probabilities.
`n`	Number of random numbers to return.

Details

The *probability mass function* at a positive integer value x of the MOEZipf distribution with parameters α and β is computed as follows:

$$p(x|\alpha,\beta) = \frac{x^{-\alpha}\beta\zeta(\alpha)}{[\zeta(\alpha) - \bar{\beta}\zeta(\alpha,x)][\zeta(\alpha) - \bar{\beta}\zeta(\alpha,x+1)]}, \alpha > 1, \beta > 0,$$

where $\zeta(\alpha)$ is the Riemann-zeta function at α, $\zeta(\alpha, x)$ is the Hurtwitz zeta function with arguments α and x, and $\bar{\beta} = 1 - \beta$.

The *cumulative distribution function*, $F_\alpha(x)$, at a given positive real value x, is calcuted from the survival function $S(x)$ as:

$$F(x) = 1 - S(x),$$

the survival function $S(x)$ is equal to:

$$S(x) = \frac{\beta\zeta(\alpha, x+1)}{\zeta(\alpha) - \bar{\beta}\zeta(\alpha, x+1)}, \forall x > 0$$

The *quantiles* of a MOEZipf distribution for a given probability vector `p`, are obtained by computing the quantiles associated to a Zipf distribution with the same parameter α, and probability vector equal to:

$$p\prime = \frac{p\,\beta}{1 + p\,(\beta - 1)}$$

Value

`dmoezipf` gives the probability mass function, `pmoezipf` gives the cumulative function, `qmoezipf` gives the quantile function, and `rmoezipf` generates random deviates.

References

Young, D. S. (2010). *Tolerance: an R package for estimating tolerance intervals*. Journal of Statistical Software, 36(5), 1-39.

Casellas, A. (2013) *La distribució Zipf Estesa segons la transformació Marshall-Olkin*. Universitat Politécnica de Catalunya.

Pérez-Casany, M. and Casellas, A. (2013) *Marshall-Olkin Extended Zipf Distribution*. arXiv preprint arXiv:1304.4540.

Duarte-López, A., Prat-Pérez, A., & Pérez-Casany, M. (2015, August). *Using the Marshall-Olkin Extended Zipf Distribution in Graph Generation*. In European Conference on Parallel Processing (pp. 493-502). Springer International Publishing.

Examples

```
dmoezipf(1:10, 2.5, 1.3)
pmoezipf(1:10, 2.5, 1.3)
qmoezipf(0.56, 2.5, 1.3)
rmoezipf(10, 2.5, 1.3)
```

`moezipfFit` *MOEZipf parameters estimation.*

Description

For a given count data set, usually of the type of ranking data or frequencies of frequencies data, estimates the parameters of the MOEZipf distribution by means of the maximum likelihood method.

Usage

```
moezipfFit(data, init_alpha, init_beta, level = 0.95, ...)
```

Arguments

`data`	Matrix of count data.
`init_alpha`	Initial value of α parameter ($\alpha > 1$).
`init_beta`	Initial value of β parameter ($\beta > 0$).
`level`	Confidence level used to calculate the intervals (default 0.95).
`...`	Further arguments to the generic functions. In case of the function *moezipfR.fit* the extra arguments are passing to the optim function.

Details

The argument `data` is a matrix where, for each row, the first column contains a count, and the second column contains its corresponding frequency.

The log-likelihood function is computed by means of the following equation:

$$l(\alpha, \beta; x) = -\alpha \sum_{i=1}^{m} f_a(x_i) log(x_i) + N(log(\beta) + \log(\zeta(\alpha)))$$

$$-\sum_{i=1}^{m} f_a(x_i) log[(\zeta(\alpha) - \bar{\beta}\zeta(\alpha, x_i)(\zeta(\alpha) - \bar{\beta}\zeta(\alpha, x_i + 1)))],$$

where N is the sample size $N = \sum_{i=1}^{m} x_i f_a(x_i)$, m is the number of different values x_i in the sample, and $f_a(x_i)$ is the absolute frequency of x_i.

The function *optim* is used to estimate the parameters.

Value

Returns a *moezipfR* object composed by the maximum likelihood parameter estimations, their standard deviation, their confidence intervals and the log-likelihood value.

See Also

`zipfExtR_getDataMatrix`, `moezipf_getInitialValues`.

Examples

```
data <- rmoezipf(100, 2.5, 1.3)
data <- zipfExtR_getDataMatrix(data)
obj <- moezipfFit(data, 1.001, 0.001)
```

`moezipfMean` *Expected value.*

Description

Computes the expected value of the MOEZipf distribution for given values of parameters α and β.

Usage

```
moezipfMean(alpha, beta, tolerance = 10^(-4))
```

Arguments

`alpha`	Value of the α parameter ($\alpha > 2$).
`beta`	Value of the β parameter ($\beta > 0$).
`tolerance`	Tolerance used in the calculations (default = 10^{-4}).

Details

The expected value of the MOEZipf distribution only exists for α values strictly greater than 2. In this case, if Y is a random variable that follows a MOEZipf distribution with parameters α and β, the expected value is computed as:

$$E(Y) = \sum_{x=1}^{\infty} \frac{\beta\zeta(\alpha)x^{-\alpha+1}}{[\zeta(\alpha) - \bar{\beta}\zeta(\alpha, x)][\zeta(\alpha) - \bar{\beta}\zeta(\alpha, x+1)]}\,, \alpha > 2\,, \beta > 0$$

The mean is computed calculating the partial sums of the serie, and it stops when two consecutive partial sums differs less than the `tolerance` value. The last partial sum is returned.

Value

A positive real value corresponding to the mean value of the distribution.

Examples

```
moezipfMean(2.5, 1.3)
moezipfMean(2.5, 1.3, 10^(-3))
```

`moezipfMoments` *Distribution Moments.*

Description

General function to compute the k-th moment of the distribution, for any $k \geq 1$ when it exists. Note that the k-th moment exists if and only if $\alpha > k + 1$. When k = 1, this function returns the same value as the moezipfMean function.

Usage

```
moezipfMoments(k, alpha, beta, tolerance = 10^(-4))
```

Arguments

`k`	Order of the moment to compute.
`alpha`	Value of the α parameter ($\alpha > k + 1$).
`beta`	Value of the β parameter ($\beta > 0$).
`tolerance`	Tolerance used in the calculations (default = 10^{-4}).

Details

The k-th moment of the MOEZipf distribution is finite for α values strictly greater than $k+1$. For a random variable Y that follows a MOEZipf distribution with parameters α and β, the k-th moment is computed as:

$$E(Y^k) = \sum_{x=1}^{\infty} \frac{\beta\zeta(\alpha)x^{-\alpha+k}}{[\zeta(\alpha) - \bar{\beta}\zeta(\alpha, x)][\zeta(\alpha) - \bar{\beta}\zeta(\alpha, x+1)]}\,, \alpha \geq k+1\,, \beta > 0$$

The k-th moment is computed calculating the partial sums of the serie, and it stops when two consecutive partial sums differs less than the `tolerance` value. The last partial sum is returned.

Value

A positive real value corresponding to the k-th moment of the distribution.

Examples

```
moezipfMoments(3, 4.5, 1.3)
moezipfMoments(3, 4.5, 1.3,  1*10^(-3))
```

`moezipfVariance` *Variance.*

Description

Computes the variance of the MOEZipf distribution for given values of α and β.

Usage

```
moezipfVariance(alpha, beta, tolerance = 10^(-4))
```

Arguments

`alpha`	Value of the α parameter ($\alpha > 3$).
`beta`	Value of the β parameter ($\beta > 0$).
`tolerance`	Tolerance used in the calculations. (default = 10^{-4})

Details

The variance of the distribution only exists for α strictly greater than 3. It is calculated as:

$$Var[Y] = E[Y^2] - (E[Y])^2$$

Value

A positive real value corresponding to the variance of the distribution.

Examples

```
moezipfVariance(3.5, 1.3)
```

`moezipf_getInitialValues` *Calculates initial values for the α and β parameters.*

Description

The initial value of the parameters are computed using the empirical absolute frequencies of values one and two. The selection of robust initial values allows to reduce the number of iterations which in turn, reduces the computation time. In the case where one of the two first positive integer values does not appear in the data set, the default values are set equal to α = 1.0001 and β = 0.0001.

Usage

```
moezipf_getInitialValues(data)
```

Arguments

`data`	Matrix of count data.

Details

The argument `data` is a matrix where, for each row, the first column corresponds to a count, and the second column contains its corresponding frequency.

To obtain the initial value for α and β, one will assume that the data come from a Zipf(α) distribution. Thus, the initial value for β is set equal to one, and the inital value for α, denoted by α_0, is obtained equating the ratio of the theoretical probabilities at one and two to the corresponding emprirical ratio. Thus,

$$\alpha_0 = log_2(\frac{f_1}{f_2})$$

where f_1 and f_2 are the absolute frequencies of one and two in the sample.

Value

Returns the initial value for parameters α and β.

References

Güney, Y., Tuaç, Y., & Arslan, O. (2016). Marshall–Olkin distribution: parameter estimation and application to cancer data. Journal of Applied Statistics, 1-13.

See Also

`zipfExtR_getDataMatrix`

Examples

```
data <- rmoezipf(100, 2.5, 1.3)
data <- zipfExtR_getDataMatrix(data)
initials <- moezipf_getInitialValues(data)
```

`zipfExtR_getDataMatrix` *Convert a sample vector to a frequency matrix.*

Description

Converts a sequence of values into a matrix of frequencies.

Usage

```
zipfExtR_getDataMatrix(values)
```

Arguments

`values`	Vector of positive integer values.

Value

The matrix of frequencies associated to the vector `values`.

Examples

```
data <- rmoezipf(100, 2.5, 1.3)
zipfExtR_getDataMatrix(data)
```

zpe *The Zipf-Poisson Extreme Distribution (ZPE).*

Description

Probability Mass Function, Cumulative Function of the ZPE distribution with parameter α and β.

Usage

```
dzpe(x, alpha, beta, log = FALSE)

pzpe(q, alpha, beta, log.p = FALSE, lower.tail = TRUE)

qzpe(p, alpha, beta, log.p = FALSE, lower.tail = TRUE)

rzpe(n, alpha, beta)
```

Arguments

`x, q`	Vector of positive integer values.
`alpha`	Value of the α parameter ($\alpha > 1$).
`beta`	Value of the β parameter ($\beta > 0$).
`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, $P[X > x]$.
`p`	Vector of probabilities.
`n`	Number of random numbers to return.

Details

The *probability mass function* at a positive integer value x of the ZPE distribution with parameters α and β is computed as follows:

$$p(x|\alpha, \beta) = \frac{e^{\beta(1-\frac{\zeta(\alpha,x)}{\zeta(\alpha)})}(e^{\beta\frac{x^{-alpha}}{\zeta(\alpha)}} - 1)}{e^{beta} - 1}, \alpha > 1, -\infty < \beta < +\infty,$$

where $\zeta(\alpha)$ is the Riemann-zeta function at α, $\zeta(\alpha, x)$ is the Hurtwitz zeta function with arguments α and x.

The *cumulative distribution function*, $F_{\alpha,\beta}(x)$, at a given positive integer value x, is calcuted as:

$$F(x) = \frac{e^{beta(1-\frac{\zeta(\alpha,x+1)}{\zeta(\alpha)})} - 1}{e^{beta} - 1}$$

The *quantiles* of a ZPE distribution for a given probability vector `p`, are obtained by computing the quantiles associated to a Zipf distribution with the same parameter α, and probability vector equal to:

$$p\prime = \frac{log(p\,(e^{\beta} - 1) + 1)}{\beta}$$

Value

`dzpe` gives the probability mass function, `pzpe` gives the cumulative function. `qzpe` gives the quantile function, and `rzpe` generates random deviates.

References

Young, D. S. (2010). *Tolerance: an R package for estimating tolerance intervals*. Journal of Statistical Software, 36(5), 1-39.

Examples

```
dzpe(1:10, 2.5, -1.5)
pzpe(1:10, 2.5, -1.5)
qzpe(0.56, 2.5, 1.3)
rzpe(10, 2.5, 1.3)
```

`zpeFit` *ZPE parameters estimation.*

Description

For a given count data set, usually of the type of ranking data or frequencies of frequencies data, estimates the parameters of the MOEZipf distribution by means of the maximum likelihood method.

Usage

```
zpeFit(data, init_alpha, init_beta, ...)
```

Arguments

`data`	Matrix of count data.
`init_alpha`	Initial value of α parameter ($\alpha > 1$).
`init_beta`	Initial value of β parameter ($\beta \in [0, 1]$).
`...`	Further arguments to the generic functions. In case of the function *moezipfR.fit* the extra arguments are passing to the optim function.

Details

The argument `data` is a matrix where, for each row, the first column contains a count, and the second column contains its corresponding frequency.

The log-likelihood function is computed by means of the following equation:

The function *optim* is used to estimate the parameters.

Value

Returns a *moezipfR* object composed by the maximum likelihood parameter estimations, their standard deviation, their confidence intervals and the log-likelihood value.

See Also

`zipfExtR_getDataMatrix`, `moezipf_getInitialValues`.

Examples

```
data <- rmoezipf(100, 2.5, 1.3)
data <- zipfExtR_getDataMatrix(data)
obj <- zpeFit(data, 1.001, 0.001)
```

`zpeMoments` *Distribution Moments.*

Description

General function to compute the k-th moment of the distribution, for any $k \geq 1$ when it exists. Note that the k-th moment exists if and only if $\alpha > k + 1$. When k = 1, this function returns the same value as the moezipfMean function.

Usage

```
zpeMoments(k, alpha, beta, tolerance = 10^(-4))
```

Arguments

`k`	Order of the moment to compute.
`alpha`	Value of the α parameter ($\alpha > k + 1$).
`beta`	Value of the β parameter ($\beta > 0$).
`tolerance`	Tolerance used in the calculations (default = 10^{-4}).

Details

The k-th moment of the MOEZipf distribution is finite for α values strictly greater than $k+1$. For a random variable Y that follows a MOEZipf distribution with parameters α and β, the k-th moment is computed as:

$$E(Y^k) = \sum_{x=1}^{\infty} \frac{\beta\zeta(\alpha)x^{-\alpha+k}}{[\zeta(\alpha) - \bar{\beta}\zeta(\alpha, x)][\zeta(\alpha) - \bar{\beta}\zeta(\alpha, x+1)]}\,, \alpha \geq k+1\,, \beta > 0$$

The k-th moment is computed calculating the partial sums of the serie, and it stops when two consecutive partial sums differs less than the `tolerance` value. The last partial sum is returned.

Value

A positive real value corresponding to the k-th moment of the distribution.

Examples

```
moezipfMoments(3, 4.5, 1.3)
moezipfMoments(3, 4.5, 1.3,  1*10^(-3))
```

`zpss` *The Zipf-Poisson Stop Sum Distribution (Z-PSS).*

Description

Probability Mass Function, Cumulative Function of the Z-PSS distribution with parameter α and λ.

Usage

```
dzpss(x, alpha, lambda, log = FALSE, isTruncated = FALSE)

pzpss(q, alpha, lambda, log.p = FALSE, lower.tail = TRUE,
  isTruncated = FALSE)
```

Arguments

`x, q`	Vector of positive integer values.
`alpha`	Value of the α parameter ($\alpha > 1$).
`lambda`	Value of the λ parameter ($\lambda >= 0$).
`log, log.p`	Logical; if TRUE, probabilities p are given as log(p).
`isTruncated`	Logical; if TRUE, the truncated version of the distribution is returned.
`lower.tail`	Logical; if TRUE (default), probabilities are $P[X \leq x]$, otherwise, $P[X > x]$.

`zpssFit` *ZPSS parameters estimation.*

Description

For a given count data set, usually of the type of ranking data or frequencies of frequencies data, estimates the parameters of the MOEZipf distribution by means of the maximum likelihood method.

Usage

```
zpssFit(data, init_alpha, init_lambda, isTruncated = FALSE, ...)
```

Arguments

`data`	Matrix of count data.
`init_alpha`	Initial value of α parameter ($\alpha > 1$).
`init_lambda`	Initial value of β parameter ($\beta \geq 0$).
`isTruncated`	Logical; if TRUE, the truncated version of the distribution is returned.
`...`	Further arguments to the generic functions. In case of the function *moezipfR.fit* the extra arguments are passing to the optim function.

Details

The argument `data` is a matrix where, for each row, the first column contains a count, and the second column contains its corresponding frequency.

The log-likelihood function is computed by means of the following equation:

The function *optim* is used to estimate the parameters.

Value

Returns a *moezipfR* object composed by the maximum likelihood parameter estimations, their standard deviation, their confidence intervals and the log-likelihood value.

See Also

`zipfExtR_getDataMatrix`, `moezipf_getInitialValues`.

Examples

```
data <- rmoezipf(100, 2.5, 1.3)
data <- zipfExtR_getDataMatrix(data)
obj <- zpssFit(data, 1.001, 0.001)
```

---

zpssMean *Expected value of the Z-PSS distribution.*

Description

Computes the expected value of the Z-PSS distribution for given values of parameters α and λ.

Usage

```
zpssMean(alpha, lambda, isTruncated = FALSE)
```

Arguments

alpha	Value of the α parameter ($\alpha > 2$).
lambda	Value of the λ parameter ($\lambda > 0$).
isTruncated	Logical; if TRUE Use the zero-truncated version of the distribution to calculate the expected value (default = FALSE).

Details

The expected value of the Z-PSS distribution only exists for α values strictly greater than 2. The value is derive from $E[Y] = E[N]\,E[X]$ where E[X] is the mean value of the Zipf distribution and E[N] is the expected value of a Poisson one. The resulting expression is set to be equal to:

$$E[Y] = \lambda\,\frac{\zeta(\alpha - 1)}{\zeta(\alpha)}$$

. Particularlly, if one is dealing with the zero-truncated version of the Z-PSS distribution. This values es calculated as:

$$E[Y^{ZT}] = \frac{\lambda\,\zeta(\alpha - 1)}{\zeta(\alpha)\,(1 - e^{-\lambda})}$$

Value

A positive real value corresponding to the mean value of the distribution.

References

Sarabia Alegría, JM. and Gómez Déniz, E. and Vázquez Polo, F. Estadística actuarial: teoría y aplicaciones. Pearson Prentice Hall.

Examples

```
zpssMean(2.5, 1.3)
zpssMean(2.5, 1.3, TRUE)
```

`zpssVariance` *Variance of the Z-PSS distribution.*

Description

Computes the variance of the Z-PSS distribution for given values of parameters α and λ.

Usage

```
zpssVariance(alpha, lambda, isTruncated = FALSE)
```

Arguments

`alpha`	Value of the α parameter ($\alpha > 3$).
`lambda`	Value of the λ parameter ($\lambda \geq 0$).
`isTruncated`	Logical; if TRUE Use the zero-truncated version of the distribution to calculate the expected value (default = FALSE).

Details

The variance of the Z-PSS distribution only exists for α values strictly greater than 3. The value is derive from $Var[Y] = E[N]\,Var[X] + E[X]^2\,Var[N]$ where E[X] and E[N] ares the expected value of the Zipf and the Poisson distributions respectively. In the same way the values of Var[X] and Var[N] stand for the variances of the Zipf and the Poisson distributions. The resulting expression is set to be equal to:

$$Var[Y] = \lambda\,\frac{\zeta(\alpha-2)}{\zeta(\alpha)}$$

. Particularlly, the variance of the zero-truncated version of the Z-PSS distribution is calculated as:

$$Var[Y^{ZT}] = \frac{\lambda\,\zeta(\alpha)\,\zeta(\alpha-2)\,(1-e^{-\lambda}) - \lambda^2\,\zeta(\alpha-1)^2\,e^{-\lambda}}{\zeta(\alpha)^2\,(1-e^{-\lambda})^2}$$

Value

A positive real value corresponding to the variance of the distribution.

References

Sarabia Alegría, JM. and Gómez Déniz, E. and Vázquez Polo, F. Estadística actuarial: teoría y aplicaciones. Pearson Prentice Hall.

Examples

```
zpssVariance(4.5, 2.3)
zpssVariance(4.5, 2.3, TRUE)
```

Index