Package ‘PearsonDS’

June 10, 2021

Type Package

Title Pearson Distribution System

Version 1.2

Date 2021-06-10

Maintainer Martin Becker <martin.becker@mx.uni-saarland.de>

Description Implementation of the Pearson distribution system, including full support for the (d,p,q,r)-family of functions for probability distributions and fitting via method of moments and maximum likelihood method.

Suggests gsl

License GPL (>= 2)

LazyLoad yes

Encoding latin1

NeedsCompilation yes

Author Martin Becker [aut, cre] (<https://orcid.org/0000-0003-2336-9751>), Stefan Klößner [aut] (<https://orcid.org/0000-0002-6558-7370>), Joel Heinrich [ctb]

Repository CRAN

Date/Publication 2021-06-10 09:30:02 UTC

R topics documented:

PearsonDS-package 2
empMoments 3
Pearson 4
Pearson0 5
pearsonDiagram 7
pearsonFitM 8
pearsonFitML 9
PearsonI 10
PearsonII 12
PearsonIII 13

PearsonIV 15
pearsonMoments 18
pearsonMSC 19
PearsonV 21
PearsonVI 22
PearsonVII 24

Index **27**

`PearsonDS-package` *Pearson Distribution System*

Description

Implementation of the `d,p,q,r` function family, calculation of moments, and fitting via (empirical) moment matching as well as maximum likelihood method for the Pearson distribution system.

Warning

If at all possible, package `gsl` should be installed. In this case, the functions for Pearson type IV distributions make use of `lngamma_complex` (see `Gamma`). If package `gsl` is not installed, some calculations for Pearson type IV distributions with (more or less) extreme parameters (ie, big `nu` and/or `m`) may slow down by factors of more than 1000.

Author(s)

Martin Becker <martin.becker@mx.uni-saarland.de> and Stefan Klößner <S.Kloessner@mx.uni-saarland.de>

Maintainer: Martin Becker <martin.becker@mx.uni-saarland.de>

References

[1] Abramowitz, M. and I. A. Stegun (1972) *Handbook of mathematical functions*, National Bureau of Standards, Applied Mathematics Series - 55, Tenth Printing, Washington D.C.

[2] Heinrich, J. (2004) *A Guide to the Pearson Type IV Distribution*, Univ. Pennsylvania, Philadelphia, Tech. Rep. CDF/Memo/Statistics/Public/6820 http://www-cdf.fnal.gov/physics/statistics/notes/cdf6820_pearson4.pdf

[3] Hida, Y., X. S. Li and D. H. Bailey (2000) *Algorithms for quad-double precision floating point arithmetic*, Lawrence Berkeley National Laboratory. Paper LBNL-48597.

[4] Johnson, N. L., Kotz, S. and Balakrishnan, N. (1994) Continuous Univariate Distributions, Vol. 1, Wiley Series in Probability and Mathematical Statistics, Wiley

[5] Johnson, N. L., Kotz, S. and Balakrishnan, N. (1994) Continuous Univariate Distributions, Vol. 2, Wiley Series in Probability and Mathematical Statistics, Wiley

[6] Willink, R. (2008) *A Closed-form Expression for the Pearson Type IV Distribution Function*, Aust. N. Z. J. Stat. 50 (2), pp. 199-205

See Also

Pearson for d,p,q,r function family for Pearson distributions, pearsonFitM and pearsonFitML for fitting Pearson distributions, pearsonMSC for model selection, pearsonMoments for calculation of (first four) moments.

Examples

```
## see documentation of individual functions
```

---

empMoments *Empirical Moments*

Description

Calculates the first four empirical moments (mean, variance, skewness, kurtosis) of a numeric vector.

Usage

```
empMoments(x)
```

Arguments

x	(numeric) vector containing the data set.

Value

Named vector of length 4 containing `mean`, `variance`, `skewness` and `kurtosis` (in this order).

Author(s)

Martin Becker <martin.becker@mx.uni-saarland.de>

See Also

PearsonDS-package

Examples

```
## Generate sample with given (theoretical) moments
DATA <- rpearson(25000,moments=c(mean=1,variance=2,skewness=1,kurtosis=5))
## Calculate corresponding empirical moments
empMoments(DATA)
```

Pearson *The Pearson Distribution System*

Description

Density, distribution function, quantile function and random generation for the Pearson distribution system.

Usage

```
dpearson(x, params, moments, log = FALSE, ...)

ppearson(q, params, moments, lower.tail = TRUE, log.p = FALSE, ...)

qpearson(p, params, moments, lower.tail = TRUE, log.p = FALSE, ...)

rpearson(n, params, moments, ...)
```

Arguments

`x, q`	vector of quantiles.
`p`	vector of probabilities.
`n`	number of observations.
`params`	vector/list of parameters for Pearson distribution. First entry gives type of distribution (0 for type 0, 1 for type I, ..., 7 for type VII), remaining entries give distribution parameters (depending on distribution type).
`moments`	optional vector/list of mean, variance, skewness, kurtosis (not excess kurtosis). Overrides `params` with corresponding pearson distribution, if given.
`log, log.p`	logical; if `TRUE`, probabilities `p` are given as `log(p)`.
`lower.tail`	logical; if `TRUE`, probabilities are $P[X \le x]$, otherwise, $P[X > x]$.
`...`	further parameters for underlying functions (currently only used for distributions of type IV).

Details

These are the wrapper functions for the (d,p,q,r)-functions of the Pearson distribution system subclasses.

Value

`dpearson` gives the density, `ppearson` gives the distribution function, `qpearson` gives the quantile function, and `rpearson` generates random deviates.

Author(s)

Martin Becker <martin.becker@mx.uni-saarland.de>

See Also

PearsonDS-package, Pearson0, PearsonI, PearsonII, PearsonIII, PearsonIV, PearsonV, PearsonVI, PearsonVII, pearsonFitM, pearsonFitML, pearsonMSC

Examples

```
## Define moments of distribution
moments <- c(mean=1,variance=2,skewness=1,kurtosis=5)
## Generate some random variates
rpearson(5,moments=moments)
## evaluate distribution function
ppearson(seq(-2,3,by=1),moments=moments)
## evaluate density function
dpearson(seq(-2,3,by=1),moments=moments)
## evaluate quantile function
qpearson(seq(0.1,0.9,by=0.2),moments=moments)
```

Pearson0 *The Pearson Type 0 (aka Normal) Distribution*

Description

Density, distribution function, quantile function and random generation for the Pearson type 0 (aka Normal) distribution.

Usage

```
dpearson0(x, mean, sd, params, log = FALSE)

ppearson0(q, mean, sd, params, lower.tail = TRUE, log.p = FALSE)

qpearson0(p, mean, sd, params, lower.tail = TRUE, log.p = FALSE)

rpearson0(n, mean, sd, params)
```

Arguments

x, q	vector of quantiles.
p	vector of probabilities.
n	number of observations.
mean	location parameter (and expectation)
sd	scale parameter (and standard deviation)

params	optional vector/list containing distribution parameters mean and sd (in this order!). Overrides parameters mean and sd, if given.
log, log.p	logical; if TRUE, probabilities p are given as log(p).
lower.tail	logical; if TRUE, probabilities are $P[X \le x]$, otherwise, $P[X > x]$.

Details

Distributions of type 0 have been added to the Pearson Distribution system in order to have the normal distributions not only nested as limits of other distribution types. The functions are only wrappers for dnorm, pnorm, qnorm and rnorm contained in package stats.

Value

dpearson0 gives the density, ppearson0 gives the distribution function, qpearson0 gives the quantile function, and rpearson0 generates random deviates.

Author(s)

Martin Becker <martin.becker@mx.uni-saarland.de>

References

See the references in Normal.

See Also

Normal, PearsonDS-package, Pearson

Examples

```
## define Pearson type 0 parameter set with mean=-1, sd=2
p0pars <- list(mean=-1, sd=2)
## calculate probability density function
dpearson0(-4:1,params=p0pars)
## calculate cumulative distribution function
ppearson0(-4:1,params=p0pars)
## calculate quantile function
qpearson0(seq(0.1,0.9,by=0.2),params=p0pars)
## generate random numbers
rpearson0(5,params=p0pars)
```

pearsonDiagram *Regions of Pearson Distribution Types*

Description

2D-Plot of the regions for the different types of Pearson distributions, depending on (squared) skewness and kurtosis.

Usage

```
pearsonDiagram(max.skewness = sqrt(14), max.kurtosis = 24,
               squared.skewness = TRUE, lwd = 2, legend = TRUE,
               n = 301)
```

Arguments

max.skewness	maximal value for the skewness.
max.kurtosis	maximal value for the kurtosis (not excess kurtosis!).
squared.skewness	plot squared skewness on x-axis (default: TRUE)?
lwd	line width for distributions of type II, III, V, VII.
legend	include legend in the plot (default: TRUE)?
n	number of points for curve.

Details

The label of the x-axis is β_1 for squared skewness and $\sqrt{\beta_1}$ for regular skewness. The label of the y-axis is β_2.

Value

Nothing useful. Function called for its side-effects.

Author(s)

Martin Becker <martin.becker@mx.uni-saarland.de> and Stefan Klößner <S.Kloessner@mx.uni-saarland.de>

References

[1] Johnson, N. L., Kotz, S. and Balakrishnan, N. (1994) Continuous Univariate Distributions, Vol. 1, Wiley Series in Probability and Mathematical Statistics, Wiley

See Also

PearsonDS-package

Examples

```
## Show me the regions for the different distribution types!
pearsonDiagram()
```

---

pearsonFitM — *Method of Moments Estimator for Pearson Distributions*

Description

This function calculates the method of moments estimator for Pearson distribution, ie, it generates a Pearson distribution with moments exactly (up to rounding errors) matching the input moments mean, variance, skewness and kurtosis.

Usage

```
pearsonFitM(mean, variance, skewness, kurtosis, moments)
```

Arguments

`mean`	target mean.
`variance`	target variance.
`skewness`	target skewness.
`kurtosis`	target kurtosis (not excess kurtosis!).
`moments`	optional vector/list of mean, variance, skewness, kurtosis (not excess kurtosis) in this order. Overrides all other input parameters, if given.

Value

List of parameters for Pearson distribution. First entry gives type of distribution (0 for type 0, 1 for type I, ..., 7 for type VII), remaining entries give distribution parameters (depending on distribution type).

Author(s)

Martin Becker <martin.becker@mx.uni-saarland.de>

References

[1] Johnson, N. L., Kotz, S. and Balakrishnan, N. (1994) Continuous Univariate Distributions, Vol. 1, Wiley Series in Probability and Mathematical Statistics, Wiley

[2] Johnson, N. L., Kotz, S. and Balakrishnan, N. (1994) Continuous Univariate Distributions, Vol. 2, Wiley Series in Probability and Mathematical Statistics, Wiley

See Also

`PearsonDS-package`, `Pearson`, `pearsonFitML`, `pearsonMoments`, `pearsonMSC`

Examples

```
## Define moments of distribution
moments <- c(mean=1,variance=2,skewness=1,kurtosis=5)
## find Pearson distribution with these parameters
ppar <- pearsonFitM(moments=moments)
print(unlist(ppar))
## check moments
pearsonMoments(params=ppar)
```

---

pearsonFitML *Maximum Likelihood Estimation of Pearson Distributions*

Description

This function tries to find the Maximum Likelihood estimator within the Pearson distribution system. ML estimation is done for all sub-classes of the distribution system via numerical optimization (with `nlminb`). The sub-class with the optimal likelihood function value and the corresponding parameters are returned.

Usage

```
pearsonFitML(x, ...)
```

Arguments

`x`	empirical data (numerical vector) for MLE.
`...`	parameters for `nlminb`.

Details

Starting values for each sub-class are found in a three-step procedure. First, the empirical moments of the input vector are calculated. In the second step, the moments are altered, such that the moment restrictions for the current sub-class are fulfilled (if necessary), and the method of moments estimator is calculated to obtain starting values for the optimizer. In the last step, the starting values are adjusted (if necessary) in order to assure that the whole sample lies in the support of the distribution.

Value

List of parameters for Pearson distribution. First entry gives type of distribution (0 for type 0, 1 for type I, ..., 7 for type VII), remaining entries give distribution parameters (depending on distribution type).

Note

The implementation is VERY preliminary (and slow). No analytical results are used, ie. no analytical solutions for ML estimators and no analytical gradients. Most of the distribution types (0, II, III, V, VII) should rather be neglected (for speed reasons), because they will contain the MLE with probability of 0.

Author(s)

Martin Becker <martin.becker@mx.uni-saarland.de>

References

[1] Johnson, N. L., Kotz, S. and Balakrishnan, N. (1994) Continuous Univariate Distributions, Vol. 1, Wiley Series in Probability and Mathematical Statistics, Wiley

[2] Johnson, N. L., Kotz, S. and Balakrishnan, N. (1994) Continuous Univariate Distributions, Vol. 2, Wiley Series in Probability and Mathematical Statistics, Wiley

See Also

PearsonDS-package, Pearson, pearsonFitM, pearsonMSC, pearsonMoments

Examples

```
## Generate sample
DATA <- rpearson(1000,moments=c(mean=1,variance=2,skewness=1,kurtosis=5))
## find Pearson distribution with these parameters
ppar <- pearsonFitML(DATA)
print(unlist(ppar))
## compare with method of moments estimator
print(unlist(pearsonFitM(moments=empMoments(DATA))))
```

PearsonI *The Pearson Type I (aka Beta) Distribution*

Description

Density, distribution function, quantile function and random generation for the Pearson type I (aka Beta) distribution.

Usage

```
dpearsonI(x, a, b, location, scale, params, log = FALSE)

ppearsonI(q, a, b, location, scale, params, lower.tail = TRUE,
          log.p = FALSE)

qpearsonI(p, a, b, location, scale, params, lower.tail = TRUE,
          log.p = FALSE)

rpearsonI(n, a, b, location, scale, params)
```

Arguments

x, q	vector of quantiles.
p	vector of probabilities.
n	number of observations.
a	first shape parameter of Pearson type I distribution.
b	second shape parameter of Pearson type I distribution.
location	location parameter of Pearson type I distribution.
scale	scale parameter of Pearson type I distribution.
params	vector/list of length 4 containing parameters a, b, location, scale for Pearson type I distribution (in this order!).
log, log.p	logical; if TRUE, probabilities p are given as log(p).
lower.tail	logical; if TRUE, probabilities are $P[X \le x]$, otherwise, $P[X > x]$.

Details

Essentially, Pearson type I distributions are (location-scale transformations of) Beta distributions, the above functions are thus simple wrappers for dbeta, pbeta, qbeta and rbeta contained in package stats. The probability density function with parameters a, b, scale= s and location= λ is given by

$$f(x) = \frac{1}{|s|}\frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)}\left(\frac{x-\lambda}{s}\right)^{a-1}\left(1-\frac{x-\lambda}{s}\right)^{b-1}$$

for $a > 0$, $b > 0$, $s \neq 0$, $0 < \frac{x-\lambda}{s} < 1$.

Value

dpearsonI gives the density, ppearsonI gives the distribution function, qpearsonI gives the quantile function, and rpearsonI generates random deviates.

Note

Negative values for scale are not excluded, albeit negative skewness is usually obtained by switching a and b (such that a>b) and not by using negative values for scale (and a<b).

Author(s)

Martin Becker <martin.becker@mx.uni-saarland.de>

References

See the references in Beta.

See Also

Beta, PearsonDS-package, Pearson

Examples

```
## define Pearson type I parameter set with a=2, b=3, location=1, scale=2
pIpars <- list(a=2, b=3, location=1, scale=2)
## calculate probability density function
dpearsonI(seq(1,3,by=0.5),params=pIpars)
## calculate cumulative distribution function
ppearsonI(seq(1,3,by=0.5),params=pIpars)
## calculate quantile function
qpearsonI(seq(0.1,0.9,by=0.2),params=pIpars)
## generate random numbers
rpearsonI(5,params=pIpars)
```

PearsonII — *The Pearson Type II (aka Symmetric Beta) Distribution*

Description

Density, distribution function, quantile function and random generation for the Pearson type II (aka symmetric Beta) distribution.

Usage

```
dpearsonII(x, a, location, scale, params, log = FALSE)

ppearsonII(q, a, location, scale, params, lower.tail = TRUE,
           log.p = FALSE)

qpearsonII(p, a, location, scale, params, lower.tail = TRUE,
           log.p = FALSE)

rpearsonII(n, a, location, scale, params)
```

Arguments

`x, q`	vector of quantiles.
`p`	vector of probabilities.
`n`	number of observations.
`a`	shape parameter of Pearson type II distribution.
`location`	location parameter of Pearson type II distribution.
`scale`	scale parameter of Pearson type II distribution.
`params`	vector/list of length 3 containing parameters `a`, `location`, `scale` for Pearson type II distribution (in this order!).
`log, log.p`	logical; if `TRUE`, probabilities `p` are given as `log(p)`.
`lower.tail`	logical; if `TRUE`, probabilities are $P[X \le x]$, otherwise, $P[X > x]$.

Details

Essentially, Pearson type II distributions are (location-scale transformations of) symmetric Beta distributions, the above functions are thus simple wrappers for `dbeta`, `pbeta`, `qbeta` and `rbeta` contained in package `stats`. The probability density function with parameters `a`, `scale`= s and `location`= λ is given by

$$f(x) = \frac{1}{|s|}\frac{\Gamma(2a)}{\Gamma(a)^2}\left(\frac{x-\lambda}{s}\cdot\left(1-\frac{x-\lambda}{s}\right)\right)^{a-1}$$

for $a > 0$, $s \neq 0$, $0 < \frac{x-\lambda}{s} < 1$.

Value

`dpearsonII` gives the density, `ppearsonII` gives the distribution function, `qpearsonII` gives the quantile function, and `rpearsonII` generates random deviates.

Author(s)

Martin Becker <martin.becker@mx.uni-saarland.de>

References

See the references in Beta.

See Also

Beta, PearsonDS-package, Pearson

Examples

```
## define Pearson type II parameter set with a=2, location=1, scale=2
pIIpars <- list(a=2, location=1, scale=2)
## calculate probability density function
dpearsonII(seq(1,3,by=0.5),params=pIIpars)
## calculate cumulative distribution function
ppearsonII(seq(1,3,by=0.5),params=pIIpars)
## calculate quantile function
qpearsonII(seq(0.1,0.9,by=0.2),params=pIIpars)
## generate random numbers
rpearsonII(5,params=pIIpars)
```

`PearsonIII` *The Pearson Type III (aka Gamma) Distribution*

Description

Density, distribution function, quantile function and random generation for the Pearson type III (aka Gamma) distribution.

Usage

```
dpearsonIII(x, shape, location, scale, params, log = FALSE)

ppearsonIII(q, shape, location, scale, params, lower.tail = TRUE,
            log.p = FALSE)

qpearsonIII(p, shape, location, scale, params, lower.tail = TRUE,
            log.p = FALSE)

rpearsonIII(n, shape, location, scale, params)
```

Arguments

`x, q`	vector of quantiles.
`p`	vector of probabilities.
`n`	number of observations.
`shape`	shape parameter of Pearson type III distribution.
`location`	location parameter of Pearson type III distribution.
`scale`	scale parameter of Pearson type III distribution.
`params`	vector/list of length 3 containing parameters `shape`, `location`, `scale` for Pearson type III distribution (in this order!).
`log, log.p`	logical; if `TRUE`, probabilities `p` are given as `log(p)`.
`lower.tail`	logical; if `TRUE`, probabilities are $P[X \leq x]$, otherwise, $P[X > x]$.

Details

Essentially, the above functions are wrappers for `dgamma`, `pgamma`, `qgamma` and `rgamma` contained in package `stats`. As a minor (but important) extension, negative `scale` parameters (which reflect the distribution at `location`) are permitted to allow for negative skewness. The probability density function with parameters `shape`$= a$, `scale`$= s$ and `location`$= \lambda$ is thus given by

$$f(x) = \frac{1}{|s|^a \Gamma(a)} |x - \lambda|^{a-1} e^{-\frac{x-\lambda}{s}}$$

for $s \neq 0$, $a > 0$ and $\frac{x-\lambda}{s} \geq 0$.

Value

`dpearsonIII` gives the density, `ppearsonIII` gives the distribution function, `qpearsonIII` gives the quantile function, and `rpearsonIII` generates random deviates.

Author(s)

Martin Becker <martin.becker@mx.uni-saarland.de>

References

See the references in GammaDist.

See Also

GammaDist, PearsonDS-package, Pearson

Examples

```
## define Pearson type III parameter set with shape=3, location=1, scale=-2
pIIIpars <- list(shape=3, location=1, scale=-0.5)
## calculate probability density function
dpearsonIII(-4:1,params=pIIIpars)
## calculate cumulative distribution function
ppearsonIII(-4:1,params=pIIIpars)
## calculate quantile function
qpearsonIII(seq(0.1,0.9,by=0.2),params=pIIIpars)
## generate random numbers
rpearsonIII(5,params=pIIIpars)
```

PearsonIV *The Pearson Type IV Distribution*

Description

Density, distribution function, quantile function and random generation for the Pearson type IV distribution.

Usage

```
dpearsonIV(x, m, nu, location, scale, params, log = FALSE)

ppearsonIV(q, m, nu, location, scale, params, lower.tail = TRUE,
           log.p = FALSE, tol = 1e-08, ...)

qpearsonIV(p, m, nu, location, scale, params, lower.tail = TRUE,
           log.p = FALSE, tol = 1e-08, ...)

rpearsonIV(n, m, nu, location, scale, params)
```

Arguments

x, q	vector of quantiles.
p	vector of probabilities.
n	number of observations.
m	first shape parameter of Pearson type IV distribution.

nu	second shape parameter (skewness) of Pearson type IV distribution.
location	location parameter of Pearson type IV distribution.
scale	scale parameter of Pearson type IV distribution.
params	vector/list of length 4 containing parameters m, nu, location, scale for Pearson type IV distribution (in this order!).
log, log.p	logical; if TRUE, probabilities p are given as log(p).
lower.tail	logical; if TRUE, probabilities are $P[X \le x]$, otherwise, $P[X > x]$.
tol	relative tolerance for evaluation of hypergeometric function 2F1 (ppearsonIV) or absolute target q-error for Newton method (qpearsonIV).
...	further parameters for underlying hypergeometric function.

Details

The Pearson type IV distribution with location parameter location$= \lambda$, scale parameter scale$= a$, and shape parameters m and ν can be obtained by its probability density function

$$f(x) = \frac{\left|\frac{\Gamma(m+\frac{\nu}{2}i)}{\Gamma(m)}\right|^2}{aB(m-\frac{1}{2},\frac{1}{2})}\left[1+\left(\frac{x-\lambda}{a}\right)^2\right]^{-m} e^{-\nu \arctan\left(\frac{x-\lambda}{a}\right)}$$

for $a > 0$, $m > \frac{1}{2}$, $\nu \neq 0$ ($\nu = 0$ corresponds to the Pearson type VII distribution family).

The normalizing constant, which involves the complex Gamma function, is calculated with lngamma_complex (see Gamma) of package gsl, if package gsl is installed. Section 5.1 of [2] contains an algorithm (C code) for the calculation of the normalizing constant, which is used otherwise, but this will be very slow for large absolute values of ν.

The generation of random numbers (rpearsonIV) uses the C code from section 7 of [2]. It is (thus) restricted to distributions with $m > 1$.

For the cumulative distribution function (ppearsonIV), numerical integration of the density function is used, if package gsl is not available. If package gsl is installed, three different methods are used, depending on the parameter constellation (the corresponding parameter regions were obtained by comprehensive benchmarks):

- numerical integration of the density function
- cdf representation of Heinrich [2]
- cdf representation of Willink [4]

The hypergeometric functions involved in the latter two representations are approximated via partial sums of the corresponding series (see [1], 15.1.1, p. 556). Depending on the parameter constellation, transformation 15.3.5 of [1] (p. 559) is applied for Heinrich's method. The evaluation of the partial sums is first carried out in (ordinary) double arithmetic. If cancellation reduces accuracy beyond tol, the evaluation is redone in double-double arithmetics. If cancellation still reduces accuracy beyond tol, the evaluation is again redone in quad-double arithmetic. Code for double-double and quad-double arithmetics is based on [3]. For Willink's representation, the hypergeometric function in the denominator of R in equation (10) is evaluated via complex gamma functions (see [1], 15.1.20, p. 556), which is fast and much more stable. A warning is issued if the approximation of the hypergeometric function seems to fail (which should not happen, since numerical integration should be carried out for critical parameter constellations).

The quantile function (qpearsonIV) is obtained via Newton's method. The Newton iteration begins in the (single) mode of the distribution, which is easily calculated (see [2], section 3). Since the mode of the distribution is the only inflection point of the cumulative distribution function, convergence is guaranteed. The Newton iteration stops when the target q-error is reached (or after a maximum of 30 iterations).

Value

dpearsonIV gives the density, ppearsonIV gives the distribution function, qpearsonIV gives the quantile function, and rpearsonIV generates random deviates.

Warning

If at all possible, package gsl should be installed. Otherwise, calculations for distributions with (more or less) extreme parameters may slow down by factors of more than 1000 and/or may become unstable.

Note

Many calculations are done in logarithms to avoid IEEE 754 underflow/overflow issues.

The description of quad-double arithmetics in [3] contains minor errors: in algorithm 9 (p. 6), lines 9 and 10 should be interchanged; in algorithm 14 (p. 10), $k < 2$ should be replaced with $k < 3$ (line 10) and $k < 3$ should be replaced with $k < 4$ (line 11).

Author(s)

Martin Becker <martin.becker@mx.uni-saarland.de>, Stefan Klößner <S.Kloessner@mx.uni-saarland.de> and Joel Heinrich

References

[1] Abramowitz, M. and I. A. Stegun (1972) *Handbook of mathematical functions*, National Bureau of Standards, Applied Mathematics Series - 55, Tenth Printing, Washington D.C.

[2] Heinrich, J. (2004) *A Guide to the Pearson Type IV Distribution*, Univ. Pennsylvania, Philadelphia, Tech. Rep. CDF/Memo/Statistics/Public/6820 http://www-cdf.fnal.gov/physics/statistics/notes/cdf6820_pearson4.pdf

[3] Hida, Y., X. S. Li and D. H. Bailey (2000) *Algorithms for quad-double precision floating point arithmetic*, Lawrence Berkeley National Laboratory. Paper LBNL-48597.

[4] Willink, R. (2008) *A Closed-form Expression for the Pearson Type IV Distribution Function*, Aust. N. Z. J. Stat. 50 (2), pp. 199-205

See Also

PearsonDS-package, Pearson

Examples

```
## define Pearson type IV parameter set with m=5.1, nu=3, location=0.5, scale=2
pIVpars <- list(m=5.1, nu=3, location=0.5, scale=2)
## calculate probability density function
dpearsonIV(-2:2,params=pIVpars)
## calculate cumulative distribution function
ppearsonIV(-2:2,params=pIVpars)
## calculate quantile function
qpearsonIV(seq(0.1,0.9,by=0.2),params=pIVpars)
## generate random numbers
rpearsonIV(5,params=pIVpars)
```

---

pearsonMoments *Moments of Pearson Distribution*

Description

Calculates the first four moments (mean, variance, skewness, kurtosis) of a Pearson distribution.

Usage

```
pearsonMoments(params, moments)
```

Arguments

`params`	vector/list of parameters for Pearson distribution. First entry gives type of distribution (0 for type 0, 1 for type I, ..., 7 for type VII), remaining entries give distribution parameters (depending on distribution type).
`moments`	optional vector/list of mean, variance, skewness, kurtosis (not excess kurtosis). Overrides `params` with corresponding pearson distribution, if given.

Value

Named vector of length 4 containing `mean`, `variance`, `skewness` and `kurtosis` (in this order).

Note

Optional parameter `moments` is merely for testing purposes. Of course, `pearsonMoments` should reproduce its input (when neglecting rounding errors) if `moments` is given.

`kurtosis` is the kurtosis of the distribution, not the excess kurtosis!

Author(s)

Martin Becker <martin.becker@mx.uni-saarland.de>

References

[1] Johnson, N. L., Kotz, S. and Balakrishnan, N. (1994) Continuous Univariate Distributions, Vol. 1, Wiley Series in Probability and Mathematical Statistics, Wiley

[2] Johnson, N. L., Kotz, S. and Balakrishnan, N. (1994) Continuous Univariate Distributions, Vol. 2, Wiley Series in Probability and Mathematical Statistics, Wiley

See Also

`PearsonDS-package`, `Pearson0`, `PearsonI`, `PearsonII`, `PearsonIII`, `PearsonIV`, `PearsonV`, `PearsonVI`, `PearsonVII`,

Examples

```
## Define moments of distribution
moments <- c(mean=1,variance=2,skewness=1,kurtosis=5)
## Are the moments reproduced?
pearsonMoments(moments=moments)
```

---

pearsonMSC — *Log-Likelihoods and Model Selection Criteria for different Pearson distribution types*

Description

This function performs (as `pearsonFitML`) an ML estimation for all sub-classes of the Pearson distribution system via numerical optimization (with `nlminb`) for model selection purposes. Apart from calculating the log-likelihood values as well as the values of some common model selection criteria (pure ML, AIC, AICc, BIC, HQC) for the different sub-classes, model selection is done for each of the criteria and the parameter estimates for each distribution sub-class are returned.

Usage

```
pearsonMSC(x, ...)
```

Arguments

`x`	empirical data (numerical vector) for MLE.
`...`	parameters for `nlminb`.

Details

For the ML estimation, see the details of `pearsonFitML`. The considered Model Selection Criteria (MSCs) are 'pure' Maximum Likelihood (`ML`), Akaike Information Criterion (`AIC`), corrected AIC (`AICc`), Bayes Information Criterion (`BIC`, also known as Schwarz Criterion), and Hannan-Quinn-Criterion (`HQC`). The definitions used for the different MSCs are

- for `ML`: $-2 \cdot \ln L(\theta)$

- for AIC: $-2 \cdot \ln L(\theta) + 2 \cdot k$
- for AICc: $-2 \cdot \ln L(\theta) + 2 \cdot k \cdot \frac{n}{n-k-1}$
- for BIC: $-2 \cdot \ln L(\theta) + k \cdot \ln(n)$
- for HQC: $-2 \cdot \ln L(\theta) + 2 \cdot k \cdot \ln(\ln(n))$

where $\ln L(\theta)$ denotes the log-Likelihood, n denotes the number of observations (ie, the length of x) and k denotes the number of parameters of the distribution (sub-class).

The best model *minimizes* the corresponding MSC function values.

Value

A list containing

MSCs: a matrix with rows ML, AIC, AICc, SIC, HQC for the different model selection criteria and columns 0, I, II, III, IV, V, VI,VII for the different distribution types containing the values of the criterion/distribution type-combinations.

logLik: a vector with the log-likelihood values for the different distribution types.

FittedDistributions: a list with vectors of the parameter estimates (preceeded by the distribution type number) for the 8 Pearson distribution sub-classes.

Best: a list with components ML, AIC, AICc, SIC, and HQC containing the 'best' distributions (distribution type number and parameter estimates) for the different Model Selection Criteria.

Note

The implementation is still preliminary (and slow). No analytical results are used, ie. no analytical solutions for ML estimators and no analytical gradients.

Author(s)

Martin Becker <martin.becker@mx.uni-saarland.de>

See Also

PearsonDS-package, Pearson, pearsonFitML

Examples

```
## Generate sample
DATA <- rpearson(500,moments=c(mean=1,variance=2,skewness=1,kurtosis=5))
## Call pearsonMSC for model selection
MSC  <- pearsonMSC(DATA,control=list(iter.max=1e5,eval.max=1e5))
## log-Likelihood values for all distribution sub-classes
print(MSC$logLik)
## Values for all MSCs and distribution sub-classes
print(MSC$MSCs)
## Model selection for all MSCs
print(MSC$Best)
```

PearsonV *The Pearson Type V (aka Inverse Gamma) Distribution*

Description

Density, distribution function, quantile function and random generation for the Pearson type V (aka Inverse Gamma) distribution.

Usage

```
dpearsonV(x, shape, location, scale, params, log = FALSE)

ppearsonV(q, shape, location, scale, params, lower.tail = TRUE,
          log.p = FALSE)

qpearsonV(p, shape, location, scale, params, lower.tail = TRUE,
          log.p = FALSE)

rpearsonV(n, shape, location, scale, params)
```

Arguments

`x, q`	vector of quantiles.
`p`	vector of probabilities.
`n`	number of observations.
`shape`	shape parameter of Pearson type V distribution.
`location`	location parameter of Pearson type V distribution.
`scale`	scale parameter of Pearson type V distribution.
`params`	vector/list of length 3 containing parameters `shape`, `location`, `scale` for Pearson type V distribution (in this order!).
`log, log.p`	logical; if `TRUE`, probabilities `p` are given as `log(p)`.
`lower.tail`	logical; if `TRUE`, probabilities are $P[X \le x]$, otherwise, $P[X > x]$.

Details

The Pearson type V distributions are essentially Inverse Gamma distributions. Thus, all functions are implemented via calls to the corresponding functions for Gamma distributions, ie. `dgamma`, `pgamma`, `qgamma` and `rgamma` in package `stats`. Negative `scale` parameters (which reflect the distribution at `location`) are permitted to allow for negative skewness. The probability density function with parameters `shape`$= a$, `scale`$= s$ and `location`$= \lambda$ is given by

$$f(x) = \frac{|s|^a}{\Gamma(a)}|x-\lambda|^{-a-1}e^{-\frac{s}{x-\lambda}}$$

for $s \neq 0$, $a > 0$ and $\frac{s}{x-\lambda} > 0$.

Value

dpearsonV gives the density, ppearsonV gives the distribution function, qpearsonV gives the quantile function, and rpearsonV generates random deviates.

Note

Since package version 0.98, the parameter scale corresponds to the usual scale parameter of the Inverse Gamma distribution (not the reciprocal value, which was implemented [incorrectly!] until package version 0.97).

Author(s)

Martin Becker <martin.becker@mx.uni-saarland.de>

References

See the references in GammaDist.

See Also

GammaDist, PearsonDS-package, Pearson

Examples

```
## define Pearson type V parameter set with shape=3, location=1, scale=-2
pVpars <- list(shape=3, location=1, scale=-2)
## calculate probability density function
dpearsonV(-4:1,params=pVpars)
## calculate cumulative distribution function
ppearsonV(-4:1,params=pVpars)
## calculate quantile function
qpearsonV(seq(0.1,0.9,by=0.2),params=pVpars)
## generate random numbers
rpearsonV(5,params=pVpars)
```

PearsonVI *The Pearson Type VI (aka Beta Prime) Distribution*

Description

Density, distribution function, quantile function and random generation for the Pearson type VI (aka Beta prime) distribution.

Usage

```
dpearsonVI(x, a, b, location, scale, params, log = FALSE)

ppearsonVI(q, a, b, location, scale, params, lower.tail = TRUE,
           log.p = FALSE)

qpearsonVI(p, a, b, location, scale, params, lower.tail = TRUE,
           log.p = FALSE)

rpearsonVI(n, a, b, location, scale, params)
```

Arguments

`x, q`	vector of quantiles.
`p`	vector of probabilities.
`n`	number of observations.
`a`	first shape parameter of Pearson type VI distribution.
`b`	second shape parameter of Pearson type VI distribution.
`location`	location parameter of Pearson type VI distribution.
`scale`	scale parameter of Pearson type VI distribution.
`params`	vector/list of length 4 containing parameters `a`, `b`, `location`, `scale` for Pearson type VI distribution (in this order!).
`log, log.p`	logical; if `TRUE`, probabilities `p` are given as `log(p)`.
`lower.tail`	logical; if `TRUE`, probabilities are $P[X \le x]$, otherwise, $P[X > x]$.

Details

Pearson type VI distributions are (location-scale transformations of) Beta prime distributions, and Beta prime distributions are scaled F-distributions. The above functions are thus implemented via calls to `df`, `pf`, `qf` and `rf` (contained in package `stats`). The probability density function with parameters `a`, `b`, `scale`$= s$ and `location`$= \lambda$ is given by

$$f(x) = \frac{\Gamma(a+b)}{|s|\Gamma(a)\Gamma(b)} \left(\frac{x-\lambda}{s}\right)^{a-1} \left(1 + \frac{x-\lambda}{s}\right)^{-a-b}$$

for $a > 0$, $b > 0$, $s \neq 0$, $\frac{x-\lambda}{s} > 0$.

Value

`dpearsonVI` gives the density, `ppearsonVI` gives the distribution function, `qpearsonVI` gives the quantile function, and `rpearsonVI` generates random deviates.

Note

Negative values for `scale` are permitted to allow for negative skewness.

Author(s)

Martin Becker <martin.becker@mx.uni-saarland.de>

References

See the references in FDist.

See Also

FDist, PearsonDS-package, Pearson

Examples

```
## define Pearson type VI parameter set with a=2, b=3, location=1, scale=2
pVIpars <- list(a=2, b=3, location=1, scale=2)
## calculate probability density function
dpearsonVI(seq(1,6,by=1),params=pVIpars)
## calculate cumulative distribution function
ppearsonVI(seq(1,6,by=1),params=pVIpars)
## calculate quantile function
qpearsonVI(seq(0.1,0.9,by=0.2),params=pVIpars)
## generate random numbers
rpearsonVI(5,params=pVIpars)
```

PearsonVII *The Pearson Type VII (aka Student's t) Distribution*

Description

Density, distribution function, quantile function and random generation for the Pearson type VII (aka Student's t) distribution.

Usage

```
dpearsonVII(x, df, location, scale, params, log = FALSE)

ppearsonVII(q, df, location, scale, params, lower.tail = TRUE,
            log.p = FALSE)

qpearsonVII(p, df, location, scale, params, lower.tail = TRUE,
            log.p = FALSE)

rpearsonVII(n, df, location, scale, params)
```

Arguments

`x, q`	vector of quantiles.
`p`	vector of probabilities.
`n`	number of observations.
`df`	degrees of freedom of Pearson type VII distribution
`location`	location parameter of Pearson type VII distribution.
`scale`	scale parameter of Pearson type VII distribution.
`params`	vector/list of length 3 containing parameters `df`, `location`, `scale` for Pearson type VII distribution (in this order!).
`log, log.p`	logical; if `TRUE`, probabilities `p` are given as `log(p)`.
`lower.tail`	logical; if `TRUE`, probabilities are $P[X \le x]$, otherwise, $P[X > x]$.

Details

The Pearson type VII distribution is a simple (location-scale) transformation of the well-known Student's t distribution; the probability density function with parameters `df`$= n$, `location`$= \lambda$ and `scale`$= s$ is given by

$$f(x) = \frac{1}{|s|} \frac{\Gamma(\frac{n+1}{2})}{\sqrt{n\pi}\Gamma(\frac{n}{2})} \left(1 + \frac{(\frac{x-\lambda}{s})^2}{n}\right)^{-\frac{n+1}{2}}$$

for $s \neq 0$. The above functions are thus only wrappers for `dt`, `pt`, `qt` and `rt` contained in package `stats`.

Value

`dpearsonVII` gives the density, `ppearsonVII` gives the distribution function, `qpearsonVII` gives the quantile function, and `rpearsonVII` generates random deviates.

Author(s)

Martin Becker <martin.becker@mx.uni-saarland.de>

References

See the references in `TDist`.

See Also

`TDist`, `PearsonDS-package`, `Pearson`

Examples

```
## define Pearson type VII parameter set with df=7, location=1, scale=1
pVIIpars <- list(df=7, location=1, scale=1)
## calculate probability density function
dpearsonVII(-2:4,params=pVIIpars)
## calculate cumulative distribution function
```

```
ppearsonVII(-2:4,params=pVIIpars)
## calculate quantile function
qpearsonVII(seq(0.1,0.9,by=0.2),params=pVIIpars)
## generate random numbers
rpearsonVII(5,params=pVIIpars)
```

Index

∗ **distribution**
- empMoments, 3
- Pearson, 4
- Pearson0, 5
- pearsonDiagram, 7
- PearsonDS-package, 2
- pearsonFitM, 8
- pearsonFitML, 9
- PearsonI, 10
- PearsonII, 12
- PearsonIII, 13
- PearsonIV, 15
- pearsonMoments, 18
- pearsonMSC, 19
- PearsonV, 21
- PearsonVI, 22
- PearsonVII, 24

∗ **package**
- PearsonDS-package, 2

Beta, *11*, *13*

curve, *7*

dpearson (Pearson), 4
dpearson0 (Pearson0), 5
dpearsonI (PearsonI), 10
dpearsonII (PearsonII), 12
dpearsonIII (PearsonIII), 13
dpearsonIV (PearsonIV), 15
dpearsonV (PearsonV), 21
dpearsonVI (PearsonVI), 22
dpearsonVII (PearsonVII), 24

empMoments, 3

FDist, *24*

Gamma, *2*, *16*
GammaDist, *15*, *22*

nlminb, *9*, *19*
Normal, *6*

Pearson, *3*, 4, *6*, *8*, *10*, *11*, *13*, *15*, *17*, *20*, *22*, *24*, *25*
Pearson0, *5*, 5, *19*
pearsonDiagram, 7
PearsonDS (PearsonDS-package), 2
PearsonDS-package, 2
pearsonFitM, *3*, *5*, 8, *10*
pearsonFitML, *3*, *5*, *8*, 9, *19*, *20*
PearsonI, *5*, 10, *19*
PearsonII, *5*, 12, *19*
PearsonIII, *5*, 13, *19*
PearsonIV, *5*, 15, *19*
pearsonMoments, *3*, *8*, *10*, 18
pearsonMSC, *3*, *5*, *8*, *10*, 19
PearsonV, *5*, *19*, 21
PearsonVI, *5*, *19*, 22
PearsonVII, *5*, *19*, 24
ppearson (Pearson), 4
ppearson0 (Pearson0), 5
ppearsonI (PearsonI), 10
ppearsonII (PearsonII), 12
ppearsonIII (PearsonIII), 13
ppearsonIV (PearsonIV), 15
ppearsonV (PearsonV), 21
ppearsonVI (PearsonVI), 22
ppearsonVII (PearsonVII), 24

qpearson (Pearson), 4
qpearson0 (Pearson0), 5
qpearsonI (PearsonI), 10
qpearsonII (PearsonII), 12
qpearsonIII (PearsonIII), 13
qpearsonIV (PearsonIV), 15
qpearsonV (PearsonV), 21
qpearsonVI (PearsonVI), 22
qpearsonVII (PearsonVII), 24

rpearson *(*Pearson*)*, 4
rpearson0 *(*Pearson0*)*, 5
rpearsonI *(*PearsonI*)*, 10
rpearsonII *(*PearsonII*)*, 12
rpearsonIII *(*PearsonIII*)*, 13
rpearsonIV *(*PearsonIV*)*, 15
rpearsonV *(*PearsonV*)*, 21
rpearsonVI *(*PearsonVI*)*, 22
rpearsonVII *(*PearsonVII*)*, 24

TDist, *25*