

Package ‘gld’

October 23, 2022

Version 2.6.6

Date 2022-10-23

Title Estimation and Use of the Generalised (Tukey) Lambda Distribution

Suggests

Imports stats, graphics, e1071, lmom

Description The generalised lambda distribution, or Tukey lambda distribution, provides a wide variety of shapes with one functional form. This package provides random numbers, quantiles, probabilities, densities and density quantiles for four different types of the distribution, the FKML (Freimer et al 1988), RS (Ramberg and Schmeiser 1974), GPD (van Staden and Loots 2009) and FM5 - see documentation for details. It provides the density function, distribution function, and Quantile-Quantile plots. It implements a variety of estimation methods for the distribution, including diagnostic plots. Estimation methods include the starship (all 4 types), method of L-Moments for the GPD and FKML types, and a number of methods for only the FKML type. These include maximum likelihood, maximum product of spacings, Titterington's method, Moments, Trimmed L-Moments and Distributional Least Absolutes.

License GPL (>= 2)

URL <https://github.com/newystats/gld/>

NeedsCompilation yes

Author Robert King [aut, cre] (<<https://orcid.org/0000-0001-7495-6599>>), Benjamin Dean [aut], Sigbert Klinke [aut], Paul van Staden [aut] (<<https://orcid.org/0000-0002-5710-5984>>)

Maintainer Robert King <Robert.King.Newcastle@gmail.com>

Repository CRAN

Date/Publication 2022-10-23 14:25:06 UTC

R topics documented:

BetaLambdaLambda	2
fit.fkml	3
fit.fkml.moments	6
fit.gpd	8
GeneralisedLambdaDistribution	11
gl.check.lambda	14
gld-Deprecated	17
gld.lmoments	17
gld.moments	19
plot.starship	20
plotgl	22
print.starship	24
qdbl-deprecated	25
qqgl	26
starship	28
starship.adaptivegrid	30
starship.obj	32
Index	34

BetaLambdaLambda	<i>Calculates Beta function for two identical parameters, allowing non-integer negative values</i>
------------------	--

Description

By defining the Beta Function in terms of the Gamma Function,

$$B(a, b) = \frac{\Gamma(a)\Gamma(b)}{\Gamma(a + b)}$$

the function can be defined for non-integer negative values of a and b. The special case of this where $a = b$ is needed to calculate the standard errors of the L Moment estimates of the gpd type of the generalised lambda distribution, so this function carries out that calculation.

Usage

```
BetaLambdaLambda(lambda)
```

Arguments

lambda A vector, each element of which is used for both arguments of the Beta function.

Details

NaN is returned for any negative integer elements of lambda.

Value

A vector the same length as lambda, containing Beta(lambda,lambda)

Author(s)

Robert King, <robert.king.newcastle@gmail.com>, <https://github.com/newstats/>
Paul van Staden

References

<https://github.com/newstats/gld/>

See Also

[beta gamma fit.gpd GeneralisedLambdaDistribution](#)

Examples

```
BetaLambdaLambda(-0.3)
```

fit.fkml	<i>Estimate parameters of the FKML parameterisation of the generalised lambda distribution</i>
----------	--

Description

Estimates parameters of the FKML parameterisation of the Generalised λ Distribution. Five estimation methods are available; Numerical Maximum Likelihood, Maximum Product of Spacings, Titterington's Method, the Starship (also available in the [starship](#) function, which uses the same underlying code as this for the fkml parameterisation), Trimmed L-Moments, L-Moments, Distributional Least Absolutes, and Method of Moments.

Usage

```
fit.fkml(x, method = "ML", t1 = 0, t2 = 0,
  l3.grid = c(-0.9, -0.5, -0.1, 0, 0.1, 0.2, 0.4, 0.8, 1, 1.5),
  l4.grid = l3.grid, record.cpu.time = TRUE, optim.method = "Nelder-Mead",
  inverse.eps = .Machine$double.eps, optim.control=list(maxit=10000),
  optim.penalty=1e20, return.data=FALSE)
```

Arguments

x	Data to be fitted, as a vector
method	A character string, to select the estimation method. One of: ML for numerical Maximum Likelihood, MPS or MSP for Maximum Spacings Product, TM for Titterington's Method, SM for Starship Method, TL for method of Trimmed L-moments, Lmom for method of L-moments, DLA for the method of Distributional Least Absolutes, or Mom for method of Moments.

t1	Number of observations to be trimmed from the left in the conceptual sample, t_1 (A non-negative integer, only used by TL-moment estimation, see details section)
t2	Number of observations to be trimmed from the right in the conceptual sample, t_2 (A non-negative integer, only used by TL-moment estimation, see details section). These two arguments are restricted by $t_1 + t_2 < n$, where n is the sample size
13.grid	A vector of values to form the grid of values of λ_3 used to find a starting point for the optimisation.
14.grid	A vector of values to form the grid of values of λ_4 used to find a starting point for the optimisation.
record.cpu.time	Boolean — should the CPU time used in fitting be recorded in the fitted model object?
optim.method	Optimisation method, use any of the options available under method of <code>optim</code> .
inverse.eps	Accuracy of calculation for the numerical determination of $F(x)$, defaults to <code>.Machine\$double.eps</code> .
optim.control	List of options for the optimisation step. See <code>optim</code> for details.
optim.penalty	The penalty to be added to the objective function if parameter values are proposed outside the allowed region
return.data	Logical: Should the function return the data (from the argument data)?

Details

Maximum Likelihood Estimation of the generalised lambda distribution (`gld`) proceeds by calculating the density of the data for candidate values of the parameters. Because the `gld` is defined by its quantile function, the method first numerically obtains $F(x)$ by inverting $Q(u)$, then obtains the density for that observation.

Maximum Product of Spacings estimation (sometimes referred to as Maximum Spacing Estimation, or Maximum Spacings Product) finds the parameter values that maximise the product of the spacings (the difference between successive depths, $F_\theta(x_{(i+1)}) - F_\theta(x_{(i)})$, where $F_\theta(x)$ is the distribution function for the candidate values of the parameters). See Dean (2013) and Cheng & Amin (1981) for details.

Titterington (1985) remarked that MPS effectively added an “extra observation”; there are N data points in the original sample, but $N + 1$ spacings in the expression maximised in MPS. Instead of using spacings between transformed data points, so method `TM` uses spacings between transformed, adjacently-averaged, data points. The spacings are given by $D_i = F_\theta(z_{(i)}) - F_\theta(z_{(i-1)})$, where $\alpha_1 = z_0 < z_1 < \dots < z_n = \alpha_2$ and $z_i = (x_{(i)} + x_{(i+1)})/2$ for $i = 1, 2, \dots, n-1$ (where α_1 and α_2 are the lower and upper bounds on the support of the distribution). This reduces the number of spacings to n and achieves concordance with the original sample size. See Titterington (1985) and Dean (2013) for details.

The starship is built on the fact that the `gld` is a transformation of the uniform distribution. Thus the inverse of this transformation is the distribution function for the `gld`. The starship method applies different values of the parameters of the distribution to the distribution function, calculates the depths q corresponding to the data and chooses the parameters that make these calculated depths

closest (as measured by the Anderson-Darling statistic) to a uniform distribution. See King & MacGillivray (1999) for details.

TL-Moment estimation chooses the values of the parameters that minimise the difference between the sample Trimmed L-Moments of the data and the Trimmed L-Moments of the fitted distribution. TL-Moments are based on inflating the conceptual sample size used in the definition of L-Moments. The t_1 and t_2 arguments to the function define the extent of trimming of the conceptual sample. Thus, the default values of $t_1=0$ and $t_2=0$ reduce the TL-Moment method to L-Moment estimation. t_1 and t_2 give the number of observations to be trimmed (from the left and right respectively) from the conceptual sample of size $n+t_1+t_2$. These two arguments should be non-negative integers, and $t_1+t_2 < n$, where n is the sample size. See Elamir and Seheult (2003) for more on TL-Moments in general, Asquith, (2007) for TL-Moments of the RS parameterisation of the gld and Dean (2013) for more details on TL-Moment estimation of the gld.

The method of distributional least absolutes (DLA) minimises the sum of absolute deviations between the order statistics and their medians (based on the candidate parameters). See Dean (2013) for more information.

Moment estimation chooses the values of the parameters that minimise the (sum of the squared) difference between the first four sample moments of the data and the first four moments of the fitted distribution.

Value

`fit.fkml` returns an object of class "starship" (regardless of the estimation method used).

`print` prints the estimated values of the parameters, while `summary.starship` prints these by default, but can also provide details of the estimation process (from the components `grid.results`, `data` and `optim` detailed below).

The value of `fit.fkml` is a list containing the following components:

<code>lambda</code>	A vector of length 4, giving the estimated parameters, in order, λ_1 - location parameter λ_2 - scale parameter λ_3 - first shape parameter λ_4 - second shape parameter
<code>grid.results</code>	output from the grid search
<code>optim</code>	output from the optim search, <code>optim</code> for details
<code>cpu</code>	A vector showing the computing time used, returned if <code>record.cpu.time</code> is TRUE
<code>data</code>	The data, if <code>return.data</code> is TRUE

Author(s)

Robert King, <robert.king.newcastle@gmail.com>, <https://github.com/newystats/>

Ben Dean, University of Newcastle <benjamin.dean@uon.edu.au>

References

Asquith, W. H. (2007), *L-Moments and TL-Moments of the Generalized Lambda Distribution*, Computational Statistics & Data Analysis, **51**, 4484–4496.

Cheng, R.C.H. & Amin, N.A.K. (1981), *Maximum Likelihood Estimation of Parameters in the Inverse Gaussian Distribution, with Unknown Origin*, *Technometrics*, **23**(3), 257–263. <https://www.jstor.org/stable/1267789>

Dean, B. (2013) *Improved Estimation and Regression Techniques with the Generalised Lambda Distribution*, PhD Thesis, University of Newcastle <https://nova.newcastle.edu.au/vital/access/manager/Repository/uon:13503>

Elamir, E. A. H., and Seheult, A. H. (2003), *Trimmed L-Moments*, *Computational Statistics & Data Analysis*, **43**, 299–314.

King, R.A.R. & MacGillivray, H. L. (1999), *A starship method for fitting the generalised λ distributions*, *Australian and New Zealand Journal of Statistics* **41**, 353–374.

Titterton, D. M. (1985), *Comment on ‘Estimating Parameters in Continuous Univariate Distributions’*, *Journal of the Royal Statistical Society, Series B*, **47**, 115–116.

See Also

[starshipGeneralisedLambdaDistribution](#)

Examples

```
example.data <- rgl(200,c(3,1,.4,-0.1),param="fkml")
example.fit <- fit.fkml(example.data,"MSP",return.data=TRUE)
print(example.fit)
summary(example.fit)
plot(example.fit,one.page=FALSE)
```

fit.fkml.moments	<i>Method of moments estimation for the FKML type of the generalised lambda distribution</i>
------------------	--

Description

Estimates parameters of the generalised lambda distribution (FKML type) using the Method of Moments, on the basis of moments calculated from data, or moment values (mean, variance, skewness ratio and kurtosis ratio (note, not the *excess kurtosis*)).

Usage

```
fit.fkml.moments(data,na.rm=TRUE,
  optim.method="Nelder-Mead",
  optim.control= list(), starting.point = c(0,0))
fit.fkml.moments.val(moments=c(mean=0, variance=1, skewness=0,
  kurtosis=3), optim.method="Nelder-Mead", optim.control= list(),
  starting.point = c(0,0))
```

Arguments

data	A vector of data
na.rm	Logical - should NAs be removed from the data - if FALSE, any NAs in the data will cause an error
moments	A vector of length 4, consisting of the mean, variance and moment ratios for skewness and kurtosis (do not subtract 3 from the kurtosis ratio)
optim.method	Optimisation method for <code>optim</code> to use, defaults to Nelder-Mead
optim.control	argument control, passed to <code>optim</code> .
starting.point	a vector of length 2, giving the starting value for λ_3 and λ_4 .

Details

Estimates parameters of the generalised lambda distribution (FKML type) using Method of Moments on the basis of moment values (mean, variance, skewness ratio and kurtosis ratio). Note this is the fourth central moment divided by the second central moment, without subtracting 3. `fit.fkml.moments` will estimate using the method of moments for a dataset, including calculating the sample moments. This function uses `optim` to find the parameters that minimise the sum of squared differences between the skewness and kurtosis sample ratios and their counterpart expressions for those ratios on the basis of the parameters λ_3 and λ_4 . On the basis of these estimates (and the mean and variance), this function then estimates $\hat{\lambda}_2$ and then $\hat{\lambda}_1$.

Note that the first 4 moments don't uniquely identify members of the generalised λ distribution. Typically, for a set of moments that correspond to a unimodal gld, there is another set of parameters that give a distribution with the same first 4 moments. This other distribution has a truncated appearance (that is, the distribution has finite support and the density is non-zero at the end points). See the examples below.

Value

A vector containing the parameters of the FKML type generalised lambda; λ_1 - location parameter λ_2 - scale parameter λ_3 - first shape parameter λ_4 - second shape parameter (See `gld` for more details)

Author(s)

Robert King, <robert.king.newcastle@gmail.com>, <https://github.com/newystats/>

Sigbert Klinke

Paul van Staden

References

Au-Yeung, Susanna W. M. (2003) *Finding Probability Distributions From Moments*, Masters thesis, Imperial College of Science, Technology and Medicine (University of London), Department of Computing

Freimer, M., Mudholkar, G. S., Kollia, G. & Lin, C. T. (1988), *A study of the generalized tukey lambda family*, *Communications in Statistics - Theory and Methods* **17**, 3547–3567.

Lakhany, Asif and Mausser, Helmut (2000) *Estimating the parameters of the generalized lambda distribution*, *Algo Research Quarterly*, **3(3)**:47–58

van Staden, Paul (2013) *Modeling of generalized families of probability distributions in the quantile statistical universe*, PhD thesis, University of Pretoria. <https://repository.up.ac.za/handle/2263/40265>

<https://github.com/newystats/gld/>

See Also

[gld.moments](#)

Examples

```
# Moment estimate
example.data = rgl(n=400,lambda1=c(0,1,0.4,0),
  param="fkml")
fit.fkml.moments(example.data)
# Approximation to the standard normal distribution
norm.approx <- fit.fkml.moments.val(c(0,1,0,3))
norm.approx
# Another distribution with the same moments
another <- fit.fkml.moments.val(c(0,1,0,3),start=c(2,2))
another
# Compared
plotgld(norm.approx$lambda,ylim=c(0,0.75),main="Approximation to the standard normal",
  sub="and another GLD with the same first 4 moments")
plotgld(another$lambda,add=TRUE,col=2)
```

fit.gpd

Estimate parameters of the GPD type generalised lambda distribution

Description

Estimates parameters of the GPD type generalised λ Distribution. Estimation is via method of L moments or the starship method.

The Method of L-Moments estimates for the GPD type are the only estimates for any generalised lambda distribution type with closed form expressions, and the only with algebraic results for standard errors of the estimates.

For further details on the starship method, see [starship](#).

Usage

```
fit.gpd(x, method = "LM", na.rm = TRUE, record.cpu.time = TRUE, return.data = FALSE,
  LambdaZeroEpsilon=1e-15)
fit.gpd.lmom(data, na.rm = TRUE,LambdaZeroEpsilon=1e-15)
fit.gpd.lmom.given(lmom, n = NULL,LambdaZeroEpsilon=1e-15)
```


Arguments

x	Data to be fitted, as a vector
method	A character string, to select the estimation method. The available methods are 1. "LM" for Method of L-Moments, 2. "SM" or "starship" for the Starship method.
na.rm	Logical: Should missing values be removed?
record.cpu.time	Logical: should the CPU time used in fitting be recorded in the fitted model object?
return.data	Logical: Should the function return the data (from the argument x)?
data	Data to be fitted, as a vector
lmoms	A numeric vector containing two L-moments and two L-moment ratios, in the order l_1, l_2, t_3, t_4 .
n	the sample size, defaults to NULL
LambdaZeroEpsilon	tolerance for lambda estimate of zero

Details

The starship method calls the [starship](#) function - see its help for more details.

The method of L-Moments equates sample L-Moments with expressions for the L-Moments of the GPD type GLD. Closed form expressions exist to give these estimates.

For many values there are two possible estimates for the same L Moment values, one in each of two regions of the GPD GLD parameter space, denoted region A and region B in van Staden (2013). More details on these regions can be found on page 154 of van Staden (2013).

If the 4th L-Moment ratio, τ_4 is less than the minimum value that τ_4 can obtain for the GPD generalised lambda distribution;

$$\tau_4^{(min)} = \frac{12 - 5\sqrt{6}}{12 + 5\sqrt{6}} \approx -0.0102051,$$

there is no possible L-Moment estimate (from either region A or B), and this function returns NA for the estimates.

When estimating from data, or for given L-Moments with n given, standard errors of the estimates are calculated if possible (standard errors are only finite if $\lambda > -0.5$).

If λ is zero, the GPD gld is a special case the Quantile Based Skew Logistic Distribution. If the estimated λ is within LambdaZeroEpsilon of zero, standard errors for alpha, beta and delta are calculated for the Quantile Based Skew Logistic Distribution and NA is returned as the standard error of λ .

Value

These functions return an object of `class` "GldGPDFit". It is a list, containing these components (optional components noted here);

estA The estimate in region A. This will be NULL if there is no estimate in region A

estB	The estimate in region B. This will be NULL if there is no estimate in region B
warn	(only if estA and estB are both NULL), the reason there are no estimates. If this is the case, the function also issues a warning .
cpu	A vector showing the computing time used, returned if record.cpu.time is TRUE (only for fit.gpd).
data	The data, if return.data is TRUE (only for fit.gpd).
param	The character "gpd", indicating the GPD type of the generalised lambda distribution.
starship	The value returned by starship , if the starship method is used.

Each of the estimate elements (if they are not NULL) are either a vector of length 4, or a 4 by 2 matrix if standard errors are calculated. The elements of the vector, or rows of the matrix are the estimated parameters, in order;

alpha	location parameter
beta	scale parameter
delta	skewness parameter
lambda	kurtosis parameter

The columns of the matrix are the parameter, and its standard error.

Author(s)

Robert King, <robert.king.newcastle@gmail.com>, <https://github.com/newystats/>

Paul van Staden

References

Van Staden, Paul J., & M.T. Loots. (2009), *Method of L-moment Estimation for the Generalized Lambda Distribution*. In Proceedings of the Third Annual ASEARC Conference. Callaghan, NSW 2308 Australia: School of Mathematical and Physical Sciences, University of Newcastle.

See Also

[GeneralisedLambdaDistribution](#)

Examples

```
fit.gpd.lmom.given(c(1,.3,.6,.8))
example.data = rgl(n=300,c(5,2,0.8,-0.1),param="gpd")
fit.gpd(example.data)
fit.gpd(example.data,method="starship")
```

 GeneralisedLambdaDistribution

The Generalised Lambda Distribution

Description

Density, density quantile, distribution function, quantile function and random generation for the generalised lambda distribution (also known as the asymmetric lambda, or Tukey lambda). Provides for four different parameterisations, the fmk1 (recommended), the rs, the gpd and a five parameter version of the FMKL, the fm5.

Usage

```

dgl(x, lambda1 = 0, lambda2 = NULL, lambda3 = NULL, lambda4 = NULL,
    param = "fkm1", lambda5 = NULL, inverse.eps = .Machine$double.eps,
    max.iterations = 500)
dqgl(p, lambda1, lambda2 = NULL, lambda3 = NULL, lambda4 = NULL,
    param = "fkm1", lambda5 = NULL)
pql(q, lambda1 = 0, lambda2 = NULL, lambda3 = NULL, lambda4 = NULL,
    param = "fkm1", lambda5 = NULL, inverse.eps = .Machine$double.eps,
    max.iterations = 500)
qgl(p, lambda1, lambda2 = NULL, lambda3 = NULL, lambda4 = NULL,
    param = "fkm1", lambda5 = NULL)
rgl(n, lambda1=0, lambda2 = NULL, lambda3 = NULL, lambda4 = NULL,
    param = "fkm1", lambda5 = NULL)
  
```

Arguments

x, q	vector of quantiles.
p	vector of probabilities.
n	number of observations.
lambda1	This can be either a single numeric value or a vector. If it is a vector, it must be of length 4 for parameterisations fmk1, rs and gpd and of length 5 for parameterisation fm5. If it is a vector, it gives all the parameters of the generalised lambda distribution (see below for details) and the other lambda arguments must be left as NULL. If it is a a single value, it is λ_1 , the location parameter of the distribution (α for the gpd parameterisation). The other parameters are given by the following arguments <i>Note that the numbering of the λ parameters for the fmk1 parameterisation is different to that used by Freimer, Mudholkar, Kollia and Lin. Note also that in the gpd parameterisation, the four parameters are labelled $\alpha, \beta, \delta, \lambda$.</i>
lambda2	λ_2 - scale parameter (β for gpd)
lambda3	λ_3 - first shape parameter (δ , a skewness parameter for gpd)
lambda4	λ_4 - second shape parameter (λ , a tail-shape parameter for gpd)

lambda5	λ_5 - a skewing parameter, in the fm5 parameterisation
param	choose parameterisation (see below for details) fmk1 uses <i>Freimer, Mudholkar, Kollia and Lin (1988)</i> (default). rs uses <i>Ramberg and Schmeiser (1974)</i> gpd uses GPD parameterisation, see van Staden and Loots (2009) fm5 uses the 5 parameter version of the FMKL parameterisation (paper to appear)
inverse.eps	Accuracy of calculation for the numerical determination of $F(x)$, defaults to <code>.Machine\$double.eps</code> . You may wish to make this a larger number to speed things up for large samples.
max.iterations	Maximum number of iterations in the numerical determination of $F(x)$, defaults to 500

Details

The generalised lambda distribution, also known as the asymmetric lambda, or Tukey lambda distribution, is a distribution with a wide range of shapes. The distribution is defined by its quantile function ($Q(u)$), the inverse of the distribution function. The `gld` package implements three parameterisations of the distribution. The default parameterisation (the FMKL) is that due to *Freimer, Mudholkar, Kollia and Lin (1988)* (see references below), with a quantile function:

$$Q(u) = \lambda_1 + \frac{\frac{u^{\lambda_3} - 1}{\lambda_3} - \frac{(1-u)^{\lambda_4} - 1}{\lambda_4}}{\lambda_2}$$

for $\lambda_2 > 0$.

A second parameterisation, the RS, chosen by setting `param="rs"` is that due to *Ramberg and Schmeiser (1974)*, with the quantile function:

$$Q(u) = \lambda_1 + \frac{u^{\lambda_3} - (1-u)^{\lambda_4}}{\lambda_2}$$

This parameterisation has a complex series of rules determining which values of the parameters produce valid statistical distributions. See [gl.check.lambda](#) for details.

Another parameterisation, the GPD, chosen by setting `param="gpd"` is due to van Staden and Loots (2009), with a quantile function:

$$Q(u) = \alpha + \beta \left((1-\delta) \frac{(u^\lambda - 1)}{\lambda} - \delta \frac{((1-u)^\lambda - 1)}{\lambda} \right)$$

for $\beta > 0$ and $-1 \leq \delta \leq 1$. (where the parameters appear in the `par` argument to the function in the order $\alpha, \beta, \delta, \lambda$). This parameterisation has simpler L-moments than other parameterisations and δ is a skewness parameter and λ is a tailweight parameter.

Another parameterisation, the FM5, chosen by setting `param="fm5"` adds an additional skewing parameter to the FMKL parameterisation. This uses the same approach as that used by Gilchrist (2000) for the RS parameterisation. The quantile function is

$$F^{-1}(u) = \lambda_1 + \frac{\frac{(1-\lambda_5)(u^{\lambda_3} - 1)}{\lambda_3} - \frac{(1+\lambda_5)((1-u)^{\lambda_4} - 1)}{\lambda_4}}{\lambda_2}$$

for $\lambda_2 > 0$ and $-1 \leq \lambda_5 \leq 1$.

The distribution is defined by its quantile function and its distribution and density functions do not exist in closed form. Accordingly, the results from `qgl` and `dgl` are the result of numerical solutions to the quantile function, using the Newton-Raphson method. Since the density quantile function, $f(F^{-1}(u))$, does exist, an additional function, `qdg1`, computes this.

The functions `qdg1.fmk1`, `qdg1.rs`, `qdg1.fm5`, `qgl.fmk1`, `qgl.rs` and `qgl.fm5` from versions 1.5 and earlier of the `gld` package have been renamed (and hidden) to `.qdg1.fmk1`, `.qdg1.rs`, `..qdg1.fm5`, `.qgl.fmk1`, `.qgl.rs` and `.qgl.fm5` respectively. See the code for more details

Value

`dgl` gives the density (based on the quantile density and a numerical solution to $F^{-1}(u) = x$),
`qdg1` gives the quantile density,
`pgl` gives the distribution function (based on a numerical solution to $F^{-1}(u) = x$),
`qgl` gives the quantile function, and
`rgl` generates random deviates.

Author(s)

Robert King, <robert.king.newcastle@gmail.com>, <https://github.com/newystats/>

References

- Freimer, M., Mudholkar, G. S., Kollia, G. & Lin, C. T. (1988), *A study of the generalized tukey lambda family*, Communications in Statistics - Theory and Methods **17**, 3547–3567.
- Gilchrist, Warren G. (2000), *Statistical Modelling with Quantile Functions*, Chapman and Hall
- Karian, Z.A., Dudewicz, E.J., and McDonald, P. (1996), *The extended generalized lambda distribution system for fitting distributions to data: history, completion of theory, tables, applications, the “Final Word” on Moment fits*, Communications in Statistics - Simulation and Computation **25**, 611–642.
- Karian, Zaven A. and Dudewicz, Edward J. (2000), *Fitting statistical distributions: the Generalized Lambda Distribution and Generalized Bootstrap methods*, Chapman & Hall
- Ramberg, J. S. & Schmeiser, B. W. (1974), *An approximate method for generating asymmetric random variables*, Communications of the ACM **17**, 78–82.
- Van Staden, Paul J., & M.T. Loots. (2009), *Method of L-moment Estimation for the Generalized Lambda Distribution*. In Proceedings of the Third Annual ASEARC Conference. Callaghan, NSW 2308 Australia: School of Mathematical and Physical Sciences, University of Newcastle.
- <https://github.com/newystats/gld/>

Examples

```
qgl(seq(0,1,0.02),0,1,0.123,-4.3)
pgl(seq(-2,2,0.2),0,1,-.1,-.2,param="fmk1")
```

gl.check.lambda	<i>Function to check the validity of parameters of the generalized lambda distribution</i>
-----------------	--

Description

Checks the validity of parameters of the generalized lambda. The tests are simple for the FMKL, FM5 and GPD types, and much more complex for the RS parameterisation.

Usage

```
gl.check.lambda(lambdas, lambda2 = NULL, lambda3 = NULL, lambda4 = NULL, param = "fkm1",
               lambda5 = NULL, vect = FALSE)
```

Arguments

lambdas	This can be either a single numeric value or a vector. If it is a vector, it must be of length 4 for parameterisations fkm1 or rs and of length 5 for parameterisation fm5. If it is a vector, it gives all the parameters of the generalised lambda distribution (see below for details) and the other lambda arguments must be left as NULL. If it is a single value, it is λ_1 , the location parameter of the distribution and the other parameters are given by the following arguments <i>Note that the numbering of the λ parameters for the fkm1 parameterisation is different to that used by Freimer, Mudholkar, Kollia and Lin.</i>
lambda2	λ_2 - scale parameter (β for gpd)
lambda3	λ_3 - first shape parameter (δ , skewness parameter for gpd)
lambda4	λ_4 - second shape parameter (λ , kurtosis parameter for gpd)
lambda5	λ_5 - a skewing parameter, in the fm5 parameterisation
param	choose parameterisation: fkm1 uses <i>Freimer, Mudholkar, Kollia and Lin (1988)</i> (default). rs uses <i>Ramberg and Schmeiser (1974)</i> fm5 uses the 5 parameter version of the FMKL parameterisation (paper to appear)
vect	A logical, set this to TRUE if the parameters are given in the vector form (it turns off checking of the format of lambdas and the other lambda arguments)

Details

See [GeneralisedLambdaDistribution](#) for details on the generalised lambda distribution. This function determines the validity of parameters of the distribution.

The FMKL parameterisation gives a valid statistical distribution for any real values of $\lambda_1, \lambda_3, \lambda_4$ and any positive real values of λ_2 .

The FM5 parameterisation gives statistical distribution for any real values of $\lambda_1, \lambda_3, \lambda_4$, any positive real values of λ_2 and values of λ_5 that satisfy $-1 \leq \lambda_5 \leq 1$.

For the RS parameterisation, the combinations of parameters value that give valid distributions are the following (the region numbers in the table correspond to the labelling of the regions in *Ramberg and Schmeiser (1974)* and *Karian, Dudewicz and McDonald (1996)*):

region	λ_1	λ_2	λ_3	λ_4	note
1	all	< 0	< -1	> 1	
2	all	< 0	> 1	< -1	
3	all	> 0	≥ 0	≥ 0	one of λ_3 and λ_4 must be non-zero
4	all	< 0	≤ 0	≤ 0	one of λ_3 and λ_4 must be non-zero
5	all	< 0	> -1 and < 0	> 1	equation 1 below must also be satisfied
6	all	< 0	> 1	> -1 and < 0	equation 2 below must also be satisfied

Equation 1

$$\frac{(1 - \lambda_3)^{1-\lambda_3} (\lambda_4 - 1)^{\lambda_4-1}}{(\lambda_4 - \lambda_3)^{\lambda_4-\lambda_3}} < -\frac{\lambda_3}{\lambda_4}$$

Equation 2

$$\frac{(1 - \lambda_4)^{1-\lambda_4} (\lambda_3 - 1)^{\lambda_3-1}}{(\lambda_3 - \lambda_4)^{\lambda_3-\lambda_4}} < -\frac{\lambda_4}{\lambda_3}$$

The GPD type gives a valid distribution provided β is positive and $0 \leq \delta \leq 1$.

Value

This logical function takes on a value of TRUE if the parameter values given produce a valid statistical distribution and FALSE if they don't

Note

The complex nature of the rules in this function for the RS parameterisation are the reason for the invention of the FMKL parameterisation and its status as the default parameterisation in the other generalized lambda functions.

Author(s)

Robert King, <robert.king.newcastle@gmail.com>, <https://github.com/newystats/>

References

Freimer, M., Mudholkar, G. S., Kollia, G. & Lin, C. T. (1988), *A study of the generalized tukey lambda family*, Communications in Statistics - Theory and Methods **17**, 3547–3567.

Karian, Z.E., Dudewicz, E.J., and McDonald, P. (1996), *The extended generalized lambda distribution system for fitting distributions to data: history, completion of theory, tables, applications, the "Final Word" on Moment fits*, Communications in Statistics - Simulation and Computation **25**, 611–642.

Ramberg, J. S. & Schmeiser, B. W. (1974), *An approximate method for generating asymmetric random variables*, Communications of the ACM **17**, 78–82.

<https://github.com/newystats/gld/>

See Also

The generalized lambda functions [GeneralisedLambdaDistribution](#)

Examples

```
gld.check.lambda(c(0,1,.23,4.5),vect=TRUE) ## TRUE
gld.check.lambda(c(0,-1,.23,4.5),vect=TRUE) ## FALSE
gld.check.lambda(c(0,1,0.5,-0.5),param="rs",vect=TRUE) ## FALSE
gld.check.lambda(c(0,2,1,3.4,1.2),param="fm5",vect=TRUE) ## FALSE
```

gld-Deprecated	<i>Deprecated functions</i>
----------------	-----------------------------

Description

qgdl: This calculates the density quantile function of the GLD, so it has been renamed [dqgl](#).

gld.lmoments	<i>Calculate L-Moments of the GPD type generalised lambda distribution for given parameter values</i>
--------------	---

Description

Calculates the first four L-Moments of the GPD type generalised λ distribution for given parameter values.

Usage

```
gld.lmoments(pars,order=1:4,ratios=TRUE,type="GPD",param=NULL)
```

Arguments

pars	A vector of length 4, giving the parameters of the GPD type generalised lambda distribution, consisting of; <ul style="list-style-type: none"> • α location parameter • $\beta > 0$ scale parameter • $0 \leq \delta \leq 1$ skewness parameter • λ kurtosis parameter
order	Integers to select the orders of L-moments to calculate. Currently this function only calculates for orders 1 to 4.
type	choose the type of generalised lambda distribution. Currently gld.lmoments only supports GPD which uses <i>van Staden and Loots (2009)</i> (default).
ratios	Logical. TRUE gives L-moment ratios for skewness and kurtosis (τ_3 and τ_4) (and all higher orders), FALSE gives the requested L-moments instead.
param	alias for the type argument. The type argument is preferred.

Details

The GPD type generalised λ distribution was introduced by van Staden and Loots (2009). It has explicit parameters for skewness and kurtosis, and closed form estimates for L-moment estimates of the parameters.

In the limit, as the kurtosis parameter, λ , goes to zero, the distribution approaches the skew logistic distribution of van Staden and King (2013). See the `sld` package for this distribution.

Value

A vector containing the selected L-moments of the GPD type generalised lambda. If `ratio` is true, the vector contains L-Moment ratios for orders 3 and over, otherwise all values are L-Moments.

Author(s)

Robert King, <robert.king.newcastle@gmail.com>, <https://github.com/newystats/>

Paul van Staden

References

van Staden, P.J. and King, Robert A.R. (2015) *The quantile-based skew logistic distribution*, Statistics and Probability Letters **96** 109–116. doi:10.1016/j.spl.2014.09.001

van Staden, Paul J. 2013 *Modeling of generalized families of probability distribution in the quantile statistical universe*. PhD thesis, University of Pretoria. <https://repository.up.ac.za/handle/2263/40265>

Van Staden, Paul J., & M.T. Loots. (2009), *Method of L-moment Estimation for the Generalized Lambda Distribution*. In Proceedings of the Third Annual ASEARC Conference. Callaghan, NSW 2308 Australia: School of Mathematical and Physical Sciences, University of Newcastle.

Quantile based Skew logistic distribution

Generalised Lambda Distribution

<https://github.com/newystats/gld/>

See Also

`sld` package

Examples

```
gld.lmoments(c(0,1,0.5,0.23))
gld.lmoments(c(0,1,0,0.23))
gld.lmoments(c(0,1,0.5,0.7))
```

gld.moments	<i>Calculate moments of the FKML type of the generalised lambda distribution for given parameter values</i>
-------------	---

Description

Calculates the mean, variance, skewness ratio and kurtosis ratio of the generalised λ distribution for given parameter values.

Usage

```
gld.moments(par, type="fkml", ratios=TRUE)
```

Arguments

par	A vector of length 4, giving the parameters of the generalised lambda distribution, consisting of; λ_1 location parameter λ_2 - scale parameter λ_3 - first shape parameter λ_4 - second shape parameter
type	choose the type of generalised lambda distribution. Currently gld.moments only supports fkml which uses <i>Freimer, Kollia, Mudholkar, and Lin (1988)</i> (default).
ratios	Logical. TRUE to give moment ratios for skewness and kurtosis, FALSE to give the third and fourth central moments instead.

Details

The FKML type of the generalised λ distribution was introduced by Freimer et al (1988) who gave expressions for the moments. In the limit, as the shape parameters (λ_3 and λ_4) go to zero, the distribution is defined using limit results. The moments in these limiting cases were given by van Staden (2013). This function calculates the first 4 moments.

See pages 96–97 of van Staden (2013) for the full expressions for these moments.

Value

A vector containing the first four moments of the FKML type generalized lambda. If ratio is true, the vector contains the mean, variance, skewness ratio and kurtosis ratio. If ratio is false, the vector contains the mean, variance, third central moment and fourth central moment.

Author(s)

Robert King, <robert.king.newcastle@gmail.com>, <https://github.com/newystats/>

Sigbert Klinke

Paul van Staden

References

Au-Yeung, Susanna W. M. (2003) *Finding Probability Distributions From Moments*, Masters thesis, Imperial College of Science, Technology and Medicine (University of London), Department of Computing

Freimer, M., Kollia, G., Mudholkar, G. S., & Lin, C. T. (1988), *A study of the generalized tukey lambda family*, Communications in Statistics - Theory and Methods **17**, 3547–3567.

Lakhany, Asif and Mausser, Helmut (2000) *Estimating the parameters of the generalized lambda distribution*, Algo Research Quarterly, **3(3)**:47–58

van Staden, Paul J. (2013) *Modeling of generalized families of probability distributions in the quantile statistical universe*, PhD thesis, University of Pretoria. <https://repository.up.ac.za/handle/2263/40265>

<https://github.com/newystats/gld/>

See Also

[fit.fkml.moments.val](#)

Examples

```
gld.moments(c(0, 1.463551, 0.1349124, 0.1349124))
gld.moments(c(0, 1.813799, 0, 0))
gld.moments(c(0, 1, 0, 3))
```

plot.starship

Plots to compare a fitted generalised lambda distribution to data

Description

Plots to compare a fitted Generalised Lambda Distribution to data. This works with fitted gld objects from [starship](#) or [fit.fkml](#).

Usage

```
## S3 method for class 'starship'
plot(x, data, ask = NULL, one.page = FALSE,
     breaks = "Sturges", plot.title = "default", ...)
```

Arguments

x	An object of class starship . These are produced by the fitting functions fit.fkml and starship .
data	Data to which the gld was fitted. Leave this as NULL if the return.data argument was TRUE in the call that created (starship or fit.fkml)

ask	Ask for user input before next plot. The default of NULL changes to TRUE if one.page is FALSE and plot is called interactively, otherwise it changes to FALSE this is then passed to <code>par(ask)</code> . Does not permanently change this setting. The argument is ignored if one.page is TRUE
one.page	If TRUE, put the two plots on one page using <code>par(mfrow=c(2,1))</code> . Does not permanently change this setting.
breaks	Control the number of histogram bins — passed to <code>hist</code> .
plot.title	Main title for histogram and QQ — passed to <code>hist(main=)</code> and <code>qqgl(main=)</code> . If you set this to "default", it will include the fitting method and gld type, for example "Starship fit of FMKL type GLD".
...	arguments passed to <code>plot</code> AND <code>hist</code>

Details

This function produces a Q-Q plot of the data against the fitted generalised lambda distribution and a histogram of the data overlaid with the fitted probability density function.

Author(s)

Robert King, <robert.king.newcastle@gmail.com>, <https://github.com/newystats/>

References

Freimer, M., Mudholkar, G. S., Kollia, G. & Lin, C. T. (1988), *A study of the generalized tukey lambda family*, Communications in Statistics - Theory and Methods **17**, 3547–3567.

Ramberg, J. S. & Schmeiser, B. W. (1974), *An approximate method for generating asymmetric random variables*, Communications of the ACM **17**, 78–82.

King, R.A.R. & MacGillivray, H. L. (1999), *A starship method for fitting the generalised λ distributions*, Australian and New Zealand Journal of Statistics **41**, 353–374

<https://github.com/newystats/gld/>

See Also

[starship](#),

Examples

```
data <- rgl(100,0,1,.2,.2)
starship.result <- starship(data,optim.method="Nelder-Mead",initgrid=list(lcvect=(0:4)/10,
ldvect=(0:4)/10),return.data=TRUE)
plot(starship.result)
```

plotgl	<i>Plots of density and distribution function for the generalised lambda distribution</i>
--------	---

Description

Produces plots of density and distribution function for the generalised lambda distribution. Although you could use `plot(function(x) dgl(x))` to do this, the fact that the density and quantiles of the generalised lambda are defined in terms of the depth, u , means that a separate function that uses the depths to produce the values to plot is more efficient

Usage

```
plotgld(lambda1 = 0, lambda2 = NULL, lambda3 = NULL, lambda4 = NULL,
        param = "fmkl", lambda5 = NULL, add = NULL, truncate = 0,
        bnw = FALSE, col.or.type = 1, granularity = 10000, xlab = "x",
        ylab = NULL, quant.probs = seq(0,1,.25), new.plot = NULL, ...)
plotglc(lambda1 = 0, lambda2 = NULL, lambda3 = NULL, lambda4 = NULL,
        param = "fmkl", lambda5 = NULL, granularity = 10000, xlab = "x",
        ylab = "cumulative probability", add = FALSE, ...)
```

Arguments

lambda1	This can be either a single numeric value or a vector. If it is a vector, it must be of length 4 for parameterisations <code>fmkl</code> or <code>rs</code> and of length 5 for parameterisation <code>fm5</code> . If it is a vector, it gives all the parameters of the generalised lambda distribution (see below for details) and the other lambda arguments must be left as <code>NULL</code> . If it is a single value, it is λ_1 , the location parameter of the distribution and the other parameters are given by the following arguments <i>Note that the numbering of the λ parameters for the <code>fmkl</code> parameterisation is different to that used by Freimer, Mudholkar, Kollia and Lin.</i>
lambda2	λ_2 - scale parameter
lambda3	λ_3 - first shape parameter
lambda4	λ_4 - second shape parameter
lambda5	λ_5 - a skewing parameter, in the <code>fm5</code> parameterisation
param	choose parameterisation: <code>fmkl</code> uses <i>Freimer, Mudholkar, Kollia and Lin (1988)</i> (default). <code>rs</code> uses <i>Ramberg and Schmeiser (1974)</i> <code>fm5</code> uses the 5 parameter version of the FMKL parameterisation (paper to appear)
add	a logical value describing whether this should add to an existing plot (using lines) or produce a new plot (using <code>plot</code>). Defaults to <code>FALSE</code> (new plot) if both <code>add</code> and <code>new.plot</code> are <code>NULL</code> .
truncate	for <code>plotgld</code> , a minimum density value at which the plot should be truncated.

bnw	a logical value, true for a black and white plot, with different densities identified using line type (lty), false for a colour plot, with different densities identified using line colour (col)
col.or.type	Colour or type of line to use
granularity	Number of points to calculate quantiles and density at — see <i>details</i>
xlab	X axis label
ylab	Y axis label
quant.probs	Quantiles of distribution to return (see <i>value</i> below). Set to NULL to suppress this return entirely.
new.plot	a logical value describing whether this should produce a new plot (using plot), or add to an existing plot (using lines). Ignored if add is set.
...	arguments that get passed to plot if this is a new plot

Details

The generalised lambda distribution is defined in terms of its quantile function. The density of the distribution is available explicitly as a function of depths, u , but not explicitly available as a function of x . This function calculates quantiles and depths as a function of depths to produce a density plot plotgld or cumulative probability plot plotglc.

The plot can be truncated, either by restricting the values using xlim — see par for details, or by the truncate argument, which specifies a minimum density. This is recommended for graphs of densities where the tail is very long.

Value

A number of quantiles from the distribution, the default being the minimum, maximum and quartiles.

Author(s)

Robert King, <robert.king.newcastle@gmail.com>, <https://github.com/newystats/>

References

Freimer, M., Mudholkar, G. S., Kollia, G. & Lin, C. T. (1988), *A study of the generalized tukey lambda family*, Communications in Statistics - Theory and Methods **17**, 3547–3567.

Ramberg, J. S. & Schmeiser, B. W. (1974), *An approximate method for generating asymmetric random variables*, Communications of the ACM **17**, 78–82.

Karian, Z.E. & Dudewicz, E.J. (2000), *Fitting Statistical Distributions to Data: The generalised Lambda Distribution and the Generalised Bootstrap Methods*, CRC Press.

<https://github.com/newystats/gld/>

See Also

[GeneralisedLambdaDistribution](#)

Examples

```

plotgld(0,1.4640474,.1349,.1349,main="Approximation to Standard Normal",
sub="But you can see this isn't on infinite support")

plotgld(1.42857143,1,.7,.3,main="The whale")
plotgld(1.42857143,1,.7,.3)
plotgld(0,-1,5,-0.3,param="rs")
plotgld(0,-1,5,-0.3,param="rs",xlim=c(1,2))
# A bizarre shape from the RS paramterisation
plotgld(0,1,5,-0.3,param="fmkl")
plotgld(10/3,1,.3,-1,truncate=1e-3)

plotgld(0,1,.0742,.0742,col.or.type=2,param="rs",
main="All distributions have the same moments",
sub="The full Range of all distributions is shown")
plotgld(0,1,6.026,6.026,col.or.type=3,new.plot=FALSE,param="rs")
plotgld(0,1,35.498,2.297,col.or.type=4,new.plot=FALSE,param="rs")
legend(0.25,3.5,lty=1,col=c(2,3,4),legend=c("(0,1,.0742,.0742)",
"(0,1,6.026,6.026)", "(0,1,35.498,2.297)"),cex=0.9)
# An illustration of problems with moments as a method of characterising shape

```

print.starship

Print (or summarise) the results of a starship estimation

Description

Print (or summarise) the results of a [starship](#) estimation of the parameters of the Generalised Lambda Distribution

Usage

```

## S3 method for class 'starship'
summary(object, ...)

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

```

Arguments

x	An object of class starship .
object	An object of class starship .
digits	minimal number of <i>significant</i> digits, see print.default .
...	arguments passed to print

Details

summary Gives the details of the [starship.adaptivegrid](#) and `optim` steps.

Author(s)

Robert King, <robert.king.newcastle@gmail.com>, <https://github.com/newystats/>
Darren Wraith

References

- Freimer, M., Mudholkar, G. S., Kollia, G. & Lin, C. T. (1988), *A study of the generalized tukey lambda family*, Communications in Statistics - Theory and Methods **17**, 3547–3567.
- Ramberg, J. S. & Schmeiser, B. W. (1974), *An approximate method for generating asymmetric random variables*, Communications of the ACM **17**, 78–82.
- King, R.A.R. & MacGillivray, H. L. (1999), *A starship method for fitting the generalised λ distributions*, Australian and New Zealand Journal of Statistics **41**, 353–374
- Owen, D. B. (1988), *The starship*, Communications in Statistics - Computation and Simulation **17**, 315–323.
- <https://github.com/newystats/gld/>

See Also

[starship](#), [starship.adaptivegrid](#), [starship.obj](#)

Examples

```
data <- rgl(100,0,1,.2,.2)
starship.result <- starship(data,optim.method="Nelder-Mead",initgrid=list(lcvect=(0:4)/10,
ldvect=(0:4)/10))
print(starship.result)
summary(starship.result,estimation.details=TRUE)
```

qdgl-deprecated	<i>Deprecated function for density quantile function of gld. See qdgl instead</i>
-----------------	---

Description

See qdgl help instead.

Usage

```
qdgl(p, lambda1, lambda2 = NULL, lambda3 = NULL, lambda4 = NULL,
param = "fkml", lambda5 = NULL)
```

Arguments

p	See qdgl help instead.
lambda1	See qdgl help instead.
lambda2	See qdgl help instead.
lambda3	See qdgl help instead.
lambda4	See qdgl help instead.
param	See qdgl help instead.
lambda5	See qdgl help instead.

Value

See qdgl help instead.

qqgl *Quantile-Quantile plot against the generalised lambda distribution*

Description

qqgl produces a Quantile-Quantile plot of data against the generalised lambda distribution, or a Q-Q plot to compare two sets of parameter values for the generalised lambda distribution. It does for the generalised lambda distribution what [qqnorm](#) does for the normal.

Usage

```
qqgl(y = NULL, lambda1 = 0, lambda2 = NULL, lambda3 = NULL, lambda4 = NULL,
     param = "fkm1", lambda5 = NULL, abline = TRUE, lambda.pars1 = NULL, lambda.pars2 = NULL,
     param2 = "fkm1", points.for.2.param.sets = 4000, ...)
```

Arguments

y	The data sample
lambda1	This can be either a single numeric value or a vector. If it is a vector, it must be of length 4 for parameterisations fmk1 or rs and of length 5 for parameterisation fm5. If it is a vector, it gives all the parameters of the generalised lambda distribution (see below for details) and the other lambda arguments must be left as NULL. Alternatively, leave lambda1 as the default value of 0 and use the lambda.pars1 argument instead. If it is a single value, it is λ_1 , the location parameter of the distribution and the other parameters are given by the following arguments <i>Note that the numbering of the λ parameters for the fmk1 parameterisation is different to that used by Freimer, Mudholkar, Kollia and Lin.</i>
lambda2	λ_2 - scale parameter
lambda3	λ_3 - first shape parameter

lambda4	λ_4 - second shape parameter
lambda5	λ_5 - a skewing parameter, in the fm5 parameterisation
param	choose parameterisation: fmk1 uses <i>Freimer, Mudholkar, Kollia and Lin (1988)</i> (default). rs uses <i>Ramberg and Schmeiser (1974)</i> fm5 uses the 5 parameter version of the FMKL parameterisation (paper to appear)
abline	A logical value, TRUE adds a line through the origin with a slope of 1 to the plot
lambda.pars1	Parameters of the generalised lambda distribution (see lambda1 to lambda4 for details).
lambda.pars2	Second set of parameters of the generalised lambda distribution (see lambda1 to lambda4 for details. Use lambda.pars1 and lambda.pars2 to produce a QQ plot comparing two generalised lambda distributions
param2	parameterisation to use for the second set of parameter values
points.for.2.param.sets	Number of quantiles to use in a Q-Q plot comparing two sets of parameter values
...	graphical parameters, passed to qqplot

Details

See [gld](#) for more details on the Generalised Lambda Distribution. A Q-Q plot provides a way to visually assess the correspondence between a dataset and a particular distribution, or between two distributions.

Value

A list of the same form as that returned by [qqline](#)

x	The x coordinates of the points that were/would be plotted, corresponding to a generalised lambda distribution with parameters $\lambda_1, \lambda_2, \lambda_3, \lambda_4$.
y	The original y vector, i.e., the corresponding y coordinates, or a corresponding set of quantiles from a generalised lambda distribution with the second set of parameters

Author(s)

Robert King, <robert.king.newcastle@gmail.com>, <https://github.com/newystats/>

References

King, R.A.R. & MacGillivray, H. L. (1999), *A starship method for fitting the generalised λ distributions*, Australian and New Zealand Journal of Statistics **41**, 353–374

<https://github.com/newystats/gld/>

See Also

[gld](#), [starship](#)

Examples

```
qqgl(rgl(100,0,1,0,-.1),0,1,0,-.1)
qqgl(lambda1=c(0,1,0.01,0.01),lambda.pars2=c(0,.01,0.01,0.01),param2="rs",pch=".")
```

starship	<i>Carry out the “starship” estimation method for the generalised lambda distribution</i>
----------	---

Description

Estimates parameters of the generalised lambda distribution on the basis of data, using the starship method. The starship method is built on the fact that the generalised lambda distribution ([gld](#)) is a transformation of the uniform distribution. This method finds the parameters that transform the data closest to the uniform distribution. This function uses a grid-based search to find a suitable starting point (using [starship.adaptivegrid](#)) then uses [optim](#) to find the parameters that do this.

Usage

```
starship(data, optim.method = "Nelder-Mead", initgrid = NULL,
inverse.eps = .Machine$double.eps, param="FMKL", optim.control=NULL, return.data=FALSE)
```

Arguments

data	Data to be fitted, as a vector
optim.method	Optimisation method for optim to use, defaults to Nelder-Mead
initgrid	Grid of values of λ_3 and λ_4 to try, in starship.adaptivegrid . This should be a list with elements, <code>lcvect</code> , a vector of values for λ_3 , <code>ldvect</code> , a vector of values for λ_4 and <code>levect</code> , a vector of values for λ_5 (<code>levect</code> is only required if <code>param</code> is <code>fm5</code>). If it is left as <code>NULL</code> , the default grid depends on the parameterisation. For <code>fmk1</code> , both <code>lcvect</code> and <code>ldvect</code> default to:

```
-1.5 -1 -0.5 -0.1 0 0.1 0.2 0.4 0.8 1 1.5
```

(`levect` is `NULL`).

For `rs`, both `lcvect` and `ldvect` default to:

```
0.1 0.2 0.4 0.8 1 1.5
```

(`levect` is `NULL`). Note that this restricts the estimates to only part of the region of the λ_3, λ_4 plane.

For `gpd`, the defaults are: δ :

```
0.3 0.5 0.7
```

and λ :

-1.5 -0.5 0 .2 .4 0.8 1.5 5

For fm5, both lcvect and ldvect default to:

-1.5 -1 -0.5 -0.1 0 0.1 0.2 0.4 0.8 1 1.5

and lcvect defaults to:

-0.5 0.25 0 0.25 0.5

inverse.eps	Accuracy of calculation for the numerical determination of $F(x)$, defaults to <code>.Machine\$double.eps</code>
param	choose parameterisation: <code>fmk1</code> uses <i>Freimer, Mudholkar, Kollia and Lin (1988)</i> (default). <code>rs</code> uses <i>Ramberg and Schmeiser (1974)</i> <code>fm5</code> uses the 5 parameter version of the FMKL parameterisation (paper to appear)
optim.control	List of options for the optimisation step. See <code>optim</code> for details. If left as <code>NULL</code> , the parscale control is set to scale λ_1 and λ_2 by the absolute value of their starting points.
return.data	Logical: Should the function return the data (from the argument data)?

Details

The starship method is described in King & MacGillivray, 1999 (see references). It is built on the fact that the generalised lambda distribution (`gld`) is a transformation of the uniform distribution. Thus the inverse of this transformation is the distribution function for the `gld`. The starship method applies different values of the parameters of the distribution to the distribution function, calculates the depths q corresponding to the data and chooses the parameters that make the depths closest to a uniform distribution.

The closeness to the uniform is assessed by calculating the Anderson-Darling goodness-of-fit test on the transformed data against the uniform, for a sample of size `length(data)`.

This is implemented in 2 stages in this function. First a grid search is carried out, over a small number of possible parameter values (see `starship.adaptivegrid` for details). Then the minimum from this search is given as a starting point for an optimisation of the Anderson-Darling value using `optim`, with method given by `optim.method`

See `GeneralisedLambdaDistribution` for details on parameterisations.

Value

`starship` returns an object of class `"starship"`.

`print` prints the estimated values of the parameters, while `summary.starship` prints these by default, but can also provide details of the estimation process (from the components `grid.results` and `optim` detailed below).

An object of class `"starship"` is a list containing at least the following components:

lambda	A vector of length 4 (or 5, for the <i>fm5</i> parameterisation), giving the estimated parameters, in order, λ_1 - location parameter λ_2 - scale parameter λ_3 - first shape parameter λ_4 - second shape parameter (See gld for details of the parameters in the <i>fm5</i> parameterisation) In the <i>gpd</i> parameterisation, the parameters are labelled: α - location parameter β - scale parameter δ - skewness parameter λ - tailweight parameter
grid.results	output from the grid search - see starship.adaptivegrid for details
optim	output from the optim search - optim for details

Author(s)

Robert King, <robert.king.newcastle@gmail.com>, <https://github.com/newystats/>
Darren Wraith

References

- Freimer, M., Mudholkar, G. S., Kollia, G. & Lin, C. T. (1988), *A study of the generalized tukey lambda family*, Communications in Statistics - Theory and Methods **17**, 3547–3567.
- Ramberg, J. S. & Schmeiser, B. W. (1974), *An approximate method for generating asymmetric random variables*, Communications of the ACM **17**, 78–82.
- King, R.A.R. & MacGillivray, H. L. (1999), *A starship method for fitting the generalised λ distributions*, Australian and New Zealand Journal of Statistics **41**, 353–374
- Owen, D. B. (1988), *The starship*, Communications in Statistics - Computation and Simulation **17**, 315–323.
- <https://github.com/newystats/gld/>

See Also

[starship.adaptivegrid](#), [starship.obj](#)

Examples

```
exampledata <- rgl(300,c(0,1,0.2,0))
starship(exampledata)
```

starship.adaptivegrid *Carry out the “starship” estimation method for the generalised lambda distribution using a grid-based search*

Description

Calculates estimates for the generalised lambda distribution on the basis of data, using the starship method. The starship method is built on the fact that the generalised lambda distribution ([gld](#)) is a transformation of the uniform distribution. This method finds the parameters that transform the data closest to the uniform distribution. This function uses a grid-based search.

Usage

```
starship.adaptivegrid(data, initgrid, inverse.eps = 1e-08, param="FMKL")
```

Arguments

data	Data to be fitted, as a vector
initgrid	A list with elements, lcvect, a vector of values for λ_3 , ldvect, a vector of values for λ_4 and levect, a vector of values for λ_5 (levect is only required if param is fm5). The parameter values given in initgrid are not checked with gl.check.lambda .
inverse.eps	Accuracy of calculation for the numerical determination of $F(x)$, defaults to 10^{-8}
param	choose parameterisation: fmk1 uses <i>Freimer, Mudholkar, Kollia and Lin (1988)</i> (default). rs uses <i>Ramberg and Schmeiser (1974)</i> fm5 uses the 5 parameter version of the FMKL parameterisation (paper to appear)

Details

The starship method is described in King and MacGillivray, 1999 (see references). It is built on the fact that the generalised lambda distribution ([gld](#)) is a transformation of the uniform distribution. Thus the inverse of this transformation is the distribution function for the gld. The starship method applies different values of the parameters of the distribution to the distribution function, calculates the depths q corresponding to the data and chooses the parameters that make the depths closest to a uniform distribution.

The closeness to the uniform is assessed by calculating the Anderson-Darling goodness-of-fit test on the transformed data against the uniform, for a sample of size `length(data)`.

This function carries out a grid-based search. This was the original method of King and MacGillivray, 1999, but you are advised to instead use [starship](#) which uses a grid-based search together with an optimisation based search.

See [GeneralisedLambdaDistribution](#) for details on parameterisations.

Value

response	The minimum “response value” — the result of the internal goodness-of-fit measure. This is the return value of <code>starship.obj</code> . See King and MacGillivray, 1999 for more details
lambda	A vector of length 4 giving the values of λ_1 to λ_4 that produce this minimum response, i.e. the estimates

Author(s)

Robert King, <robert.king.newcastle@gmail.com>, <https://github.com/newystats/>

Darren Wraith

References

Freimer, M., Mudholkar, G. S., Kollia, G. & Lin, C. T. (1988), *A study of the generalized tukey lambda family*, Communications in Statistics - Theory and Methods **17**, 3547–3567.

Ramberg, J. S. & Schmeiser, B. W. (1974), *An approximate method for generating asymmetric random variables*, Communications of the ACM **17**, 78–82.

King, R.A.R. & MacGillivray, H. L. (1999), *A starship method for fitting the generalised λ distributions*, Australian and New Zealand Journal of Statistics **41**, 353–374

Owen, D. B. (1988), *The starship*, Communications in Statistics - Computation and Simulation **17**, 315–323.

<https://github.com/newystats/gld/>

See Also

[starship](#), [starship.obj](#)

Examples

```
data <- rgl(100,0,1,.2,.2)
starship.adaptivegrid(data,list(lcvect=(0:4)/10,ldvect=(0:4)/10))
```

starship.obj

Objective function that is minimised in starship estimation method

Description

The starship is a method for fitting the generalised lambda distribution. See [starship](#) for more details.

This function is the objective function minimised in the methods. It is a goodness of fit measure carried out on the depths of the data.

Usage

```
starship.obj(par, data, inverse.eps, param = "fml")
```

Arguments

par	parameters of the generalised lambda distribution, a vector of length 4, giving λ_1 to λ_4 . See GeneralisedLambdaDistribution for details on the definitions of these parameters
data	Data — a vector
inverse.eps	Accuracy of calculation for the numerical determination of $F(x)$, defaults to 10^{-8}
param	choose parameterisation: fml uses <i>Freimer, Mudholkar, Kollia and Lin (1988)</i> (default). rs uses <i>Ramberg and Schmeiser (1974)</i>

Details

The starship method is described in King and MacGillivray, 1999 (see references). It is built on the fact that the generalised lambda distribution ([gld](#)) is a transformation of the uniform distribution. Thus the inverse of this transformation is the distribution function for the gld. The starship method applies different values of the parameters of the distribution to the distribution function, calculates the depths q corresponding to the data and chooses the parameters that make the depths closest to a uniform distribution.

The closeness to the uniform is assessed by calculating the Anderson-Darling goodness-of-fit test on the transformed data against the uniform, for a sample of size `length(data)`.

This function returns that objective function. It is provided as a separate function to allow users to carry out minimisations using [optim](#) or other methods. The recommended method is to use the [starship](#) function.

Value

The Anderson-Darling goodness of fit measure, computed on the transformed data, compared to a uniform distribution. *Note that this is NOT the goodness-of-fit measure of the generalised lambda distribution with the given parameter values to the data.*

Author(s)

Robert King, <robert.king.newcastle@gmail.com>, <https://github.com/newystats/> Darren Wraith

References

Freimer, M., Mudholkar, G. S., Kollia, G. & Lin, C. T. (1988), *A study of the generalized tukey lambda family*, Communications in Statistics - Theory and Methods **17**, 3547–3567.

Ramberg, J. S. & Schmeiser, B. W. (1974), *An approximate method for generating asymmetric random variables*, Communications of the ACM **17**, 78–82.

King, R.A.R. & MacGillivray, H. L. (1999), *A starship method for fitting the generalised λ distributions*, Australian and New Zealand Journal of Statistics **41**, 353–374

Owen, D. B. (1988), *The starship*, Communications in Statistics - Computation and Simulation **17**, 315–323.

<https://github.com/newystats/gld/>

See Also

[starship](#), [starship.adaptivegrid](#)

Examples

```
data <- rgl(100,0,1,.2,.2)
starship.obj(c(0,1,.2,.2),data,inverse.eps=1e-10,"fmk1")
```

Index

- * **aplot**
 - plotgl, 22
 - qqgl, 26
- * **distribution**
 - fit.fkml, 3
 - fit.fkml.moments, 6
 - fit.gpd, 8
 - GeneralisedLambdaDistribution, 11
 - gl.check.lambda, 14
 - gld.lmoments, 17
 - gld.moments, 19
 - plot.starship, 20
 - plotgl, 22
 - print.starship, 24
 - qqgl, 26
 - starship, 28
 - starship.adaptivegrid, 30
 - starship.obj, 32
- * **hplot**
 - plotgl, 22
 - qqgl, 26
- * **math**
 - BetaLambdaLambda, 2
- beta, 3
- BetaLambdaLambda, 2
- class, 5, 9, 29
- dgl (GeneralisedLambdaDistribution), 11
- dqgl, 17
- dqgl (GeneralisedLambdaDistribution), 11
- fit.fkml, 3, 20
- fit.fkml.moments, 6
- fit.fkml.moments.val, 20
- fit.gpd, 3, 8
- gamma, 3
- GeneralisedLambdaDistribution, 3, 6, 10, 11, 14, 17, 23, 29, 31, 32
- GeneralizedLambdaDistribution (GeneralisedLambdaDistribution), 11
- gl.check.lambda, 12, 14, 31
- gld, 4, 7, 27–31, 33
- gld (GeneralisedLambdaDistribution), 11
- gld-Deprecated, 17
- gld.lmoments, 17
- gld.moments, 8, 19
- hist, 21
- optim, 4, 5, 7, 28–30, 33
- par, 21
- pgl (GeneralisedLambdaDistribution), 11
- plot, 21
- plot.starship, 20
- plotgl, 22
- plotglc (plotgl), 22
- plotgld (plotgl), 22
- print, 24
- print.default, 24
- print.starship, 24
- qdgl (qdgl-deprecated), 25
- qdgl-deprecated, 25
- qgl (GeneralisedLambdaDistribution), 11
- qqgl, 21, 26
- qqline, 27
- qqnorm, 26
- qqplot, 27
- rgl (GeneralisedLambdaDistribution), 11
- starship, 3, 6, 8–10, 20, 21, 24, 25, 27, 28, 31–33
- starship.adaptivegrid, 24, 25, 28–30, 30, 33
- starship.obj, 25, 30, 32, 32
- summary.starship, 5, 29

`summary.starship (print.starship)`, [24](#)

`warning`, [10](#)