Package 'gld'

July 2, 2014

Version 2.2.1

Date 2014/02/19

Title Estimation and use of the generalised (Tukey) lambda distribution

Depends stats

Suggests graphics, lmom, e1071

Author
Robert King <Robert.King@newcastle.edu.au>, Ben Dean <benjamin.dean@uon.edu.au>, Sigbert Klinke

Maintainer Robert King <Robert.King@newcastle.edu.au>

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 for four different parameterisations of the distribution. 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 parameterisations) and a number of methods for only the FKML parameterisation. These include maximum likelihood, maximum product of spacings, Titterington's method, L moments, Trimmed L moments and Distributional Least Absolutes.

License GPL (>= 2)

URL http://tolstoy.newcastle.edu.au/~rking/publ/rprogs/information.html

NeedsCompilation yes

Repository CRAN

Date/Publication 2014-02-19 07:39:51

R topics documented:

`fit.fkml` *Estimate parameters of the FKML parameterisation of the generalised lambda distribution*

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), and Trimmed L-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 Product of Spacings, `TM` for Titterington's Method, `SM` for Starship Method, `TL` for Method of TL-moments, or `DLA` for the method of distributional least absolutes.
`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

`l3.grid` A vector of values to form the grid of values of λ_3 used to find a starting point for the optimisation.

`l4.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 `optim`.

`inverse.eps` Accuracy of calculation for the numerical determination of $F(x)$, defaults to `.Machine$double.eps`.

`optim.control` List of options for the optimisation step. See `optim` 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 $\alpha_1 = z_0 < z_1 < \ldots < z_n = \alpha_2$ and $z_i = (x_{(i)} + x_{(i+1)})/2$ for i = 1,2, ... 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 $g\lambda d$ 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 `t1` and `t2` arguments to the function define the extent of trimming of the conceptual sample. Thus, the default values of `t1=0` and `t2=0` reduce the TL-Moment method to L-Moment estimation.

t1 and t2 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.

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:

lambda	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
grid.results	output from the grid search
optim	output from the optim search, optim for details
cpu	A vector showing the computing time used, returned if record.cpu.time is TRUE
data	The data, if return.data is TRUE

Author(s)

Robert King, <robert.king@newcastle.edu.au>, http://tolstoy.newcastle.edu.au/~rking/

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. http://www.jstor.org/stable/1267789

Dean, B. (2013) *Improved Estimation and Regression Techniques with the Generalised Lambda Distribution*, PhD Thesis, University of Newcastle http://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.

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

See Also

starship GeneralisedLambdaDistribution

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)
```

GeneralisedLambdaDistribution
The Generalised Lambda Distribution

Description

Density, quantile density, 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 `fmkl` (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 = "fkml", lambda5 = NULL, inverse.eps = .Machine$double.eps,
  max.iterations = 500)
dqgl(p, lambda1, lambda2 = NULL, lambda3 = NULL, lambda4 = NULL,
  param = "fkml", lambda5 = NULL)
qdgl(p, lambda1, lambda2 = NULL, lambda3 = NULL, lambda4 = NULL,
  param = "fkml", lambda5 = NULL)
pgl(q, lambda1 = 0, lambda2 = NULL, lambda3 = NULL, lambda4 = NULL,
  param = "fkml", lambda5 = NULL, inverse.eps = .Machine$double.eps,
  max.iterations = 500)
qgl(p, lambda1, lambda2 = NULL, lambda3 = NULL, lambda4 = NULL,
  param = "fkml", lambda5 = NULL)
rgl(n, lambda1=0, lambda2 = NULL, lambda3 = NULL, lambda4 = NULL,
  param = "fkml", 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 `fmkl`, `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

Note that the numbering of the λ parameters for the fmkl *parameterisation is different to that used by Freimer, Mudholkar, Kollia and Lin. Note also that in the* `gpd` *parameterisation, the four parameters are labelled $\alpha, \beta, \delta, \lambda$.*

`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) `fmkl` uses *Freimer, Mudholkar, Kollia and Lin (1988)* (default). `rs` uses *Ramberg and Schmeiser (1974)* `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 `.Machine$double.eps`

`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:

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

for $\beta > 0$ and $0 \leq \delta \leq 1$. (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 `pgl` and `dgl` are the result of numerical solutions to the quantile function, using the Newton-Raphson method. Since the quantile density function, $f(F^{-1}(u))$, does exist, an additional function, `qdgl`, computes this.

The functions `qdgl.fmkl`, `qdgl.rs`, `qdgl.fm5`, `qgl.fmkl`, `qgl.rs` and `qgl.fm5` from versions 1.5 and earlier of the `gld` package have been renamed (and hidden) to `.qdgl.fmkl`, `.qdgl.rs`, `..qdgl.fm5`, `.qgl.fmkl`, `.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 $F^{-1}(u) = x$),

`qdgl` 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.edu.au>, http://tolstoy.newcastle.edu.au/~rking/

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 \& 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 fi ts*, 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.

http://tolstoy.newcastle.edu.au/~rking/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="fmkl",inverse.eps=1e-10)
rgl(21,c(3,2,0.3,-0.1),param="gpd")
# calculate the probabilities less accurately than normal
```

---

gl.check.lambda *Function to check the validity of parameters of the generalized lambda distribution*

Description

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

Usage

```
gl.check.lambda(lambdas, lambda2 = NULL, lambda3 = NULL, lambda4 = NULL, param = "fkml",
  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 `fmkl` 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 a single value, it is λ_1, the location parameter of the distribution and the other parameters are given by the following arguments *Note that the numbering of the λ parameters for the* fmkl *parameterisation is different to that used by Freimer, Mudholkar, Kollia and Lin.*
`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: `fmkl` uses *Freimer, Mudholkar, Kollia and Lin (1988)* (default). `gpd` uses the GPD type *van Staden and Loots (2009)* `rs` uses *Ramberg and Schmeiser (1974)* `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 λ_1, λ_3,λ_4 and any positive real values of λ_2.

The FM5 parameterisation gives statistical distribution for any real values of λ_1, λ_3, λ_4, any positive real values of λ_2 and values of λ_5 that satisfy $-1 \leq \lambda_5 \leq 1$.

The GPD type gives a valid distribution for any real values of α and λ, any positive real values of β and values of δ that satisfy $0 \leq \delta \leq 1$

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

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

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.edu.au>, http://tolstoy.newcastle.edu.au/~rking/

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.

http://tolstoy.newcastle.edu.au/~rking/gld/

See Also

The generalized lambda functions GeneralisedLambdaDistribution

Examples

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

---

plot.starship — *Plots to compare a fitted generalised lambda distribution to data*

Description

Plots to compare estimated Generalised Lambda Distribution parameters to data. This works for the fitted objects created by starship and fit.fkml.

Usage

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

Arguments

x	An object of class starship.
data	Data to which the gld was fitted. Leave this as NULL if the return.data argument was TRUE in the starship call that created x.
ask	Ask for user input before next plot — passed to par(ask). Does not permanently change this setting. Ignored if one.page is TRUE
one.page	Put the two plots on one page using par(mfrow=c(2,1)). Does not permanently change this setting.
breaks	Control the number of histogram bins — passed to hist.
histogram.title	Main title for histogram — passed to hist(main=).
...	arguments passed to plot AND hist

Details

summary Gives the details of the starship.adaptivegrid and optim steps.

The class is named starship, after the first estimation method implemented in this package, but this plot is available for any estimated generalised lambda parameters.

Author(s)

Robert King, <robert.king@newcastle.edu.au>, http://tolstoy.newcastle.edu.au/~rking/

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

http://tolstoy.newcastle.edu.au/~rking/gld/

See Also

starship,fit.fkml

Examples

```
set.seed(2308)
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,one.page=FALSE)
```

plotgl — *Plots of density and distribution function for the generalised lambda distribution*

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 seperate 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 `fmkl` 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 a single value, it is λ_1, the location parameter of the distribution and the other parameters are given by the following arguments

Note that the numbering of the λ parameters for the fmkl *parameterisation is different to that used by Freimer, Mudholkar, Kollia and Lin.*

`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: `fmkl` uses *Freimer, Mudholkar, Kollia and Lin (1988)* (default). `rs` uses *Ramberg and Schmeiser (1974)* `fm5` 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 `plot`). Defaults to FALSE (new plot) if both `add` and `new.plot` are NULL.

`truncate` for `plotgld`, 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 *details*
xlab	X axis label
ylab	Y axis label
quant.probs	Quantiles of distribution to return (see *value* 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.edu.au>, http://tolstoy.newcastle.edu.au/~rking/

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.

http://tolstoy.newcastle.edu.au/~rking/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")
plotglc(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 estimation of Generalised Lambda Distribution*

Description

Print (or summarise) the results of estimation of the parameters of the Generalised Lambda Distribution, from either `starship` or `fit.fkml`

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 *significant* digits, see `print.default`.
`...`	arguments passed to `print`

Details

summary Gives the details of the starship.adaptivegrid and optim steps.

The class is named starship, after the first estimation method implemented in this package, but is used for any estimated generalised lambda parameters.

Author(s)

Robert King, <robert.king@newcastle.edu.au>, http://tolstoy.newcastle.edu.au/~rking/

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.

http://tolstoy.newcastle.edu.au/~rking/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)
```

---

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 = "fkml", lambda5 = NULL, abline = TRUE, lambda.pars1 = NULL, lambda.pars2 = NULL,
  param2 = "fkml", 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 `fmkl` 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 a single value, it is λ_1, the location parameter of the distribution and the other parameters are given by the following arguments

Note that the numbering of the λ parameters for the fmkl *parameterisation is different to that used by Freimer, Mudholkar, Kollia and Lin.*

`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: `fmkl` uses *Freimer, Mudholkar, Kollia and Lin (1988)* (default). `rs` uses *Ramberg and Schmeiser (1974)* `fm5` uses the 5 parameter version of the FMKL parameterisation (paper to appear)

`abline` A logical value, TRUE adds a line through the origian 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 distibution with parameters λ_1, λ_2, λ_3, λ_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.edu.au>, http://tolstoy.newcastle.edu.au/~rking/

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

http://tolstoy.newcastle.edu.au/~rking/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 — *Carry out the "starship" estimation method for the generalised lambda distribution*

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. For the fkml parameterisation, this function calls `fit.fkml` to estimate.

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, `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`).

If it is left as NULL, the default grid depends on the parameterisation. For `fmkl`, both lcvect and ldvect default to:

-0.9 -0.5 -0.1 0.0 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. It is possible to use this function to obtain starship estimates in the other regions of the plane where the `rs` parameterisation is valid (see `gl.check.lambda` for details). Just set the values of initgrid to include those regions.

For `gpd`, the defaults are: δ:

0.3 0.5 0.7

and λ:

-1.5 -0.5 0.0 0.2 0.4 0.8 1.5 5

For `fm5`, both lcvect and ldvect default to:

-1.5 -1 -.5 -0.1 0 0.1 0.2 0.4 0.8 1 1.5

and `levect` 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 `.Machine$double.eps`

`param` choose parameterisation: `fmkl` uses *Freimer, Mudholkar, Kollia and Lin (1988)* (default). `rs` uses *Ramberg and Schmeiser (1974)* `fm5` uses the 5 parameter version of the FMKL parameterisation (paper to appear) `gpd` uses the *van Staden and Loots (2009)*, or gpd parameterisation.

`optim.control` List of options for the optimisation step. See `optim` for details. If left as NULL, 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)? Not implemented for fkml parameterisation

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`

The `fkml` parameterisation starship uses seperate (faster) code. See `fit.fkml` for details.

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`, `data` and `optim` detailed below).

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

lambda
: A vector of length 4 (or 5, for the *fm5* 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 *fm5* parameterisation)
: In the *gpd* 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

data
: The data, if `return.data` is TRUE

Author(s)

Robert King, <robert.king@newcastle.edu.au>, http://tolstoy.newcastle.edu.au/~rking/

Ben Dean

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.

http://tolstoy.newcastle.edu.au/~rking/gld/

See Also

starship.adaptivegrid, starship.obj

Examples

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

---

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: fmkl uses *Freimer, Mudholkar, Kollia and Lin (1988)* (default). rs uses *Ramberg and Schmeiser (1974)* fm5 uses the 5 parameter version of the FMKL parameterisation (paper to appear)

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 function carries out a grid-based search. This was the original method of King \& 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 starship.obj. See King \& 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.edu.au>, http://tolstoy.newcastle.edu.au/~rking/

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.

http://tolstoy.newcastle.edu.au/~rking/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 funciton 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 = "fmkl")
```

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: fmkl uses *Freimer, Mudholkar, Kollia and Lin (1988)* (default). rs uses *Ramberg and Schmeiser (1974)*

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 function returns that objective function. It is provided as a seperate 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.edu.au>, http://tolstoy.newcastle.edu.au/~rking/
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.

http://tolstoy.newcastle.edu.au/~rking/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,"fmkl")
```

Index