

# Package ‘Davies’

May 26, 2016

**Title** The Davies Quantile Function

**Version** 1.1-9

**Author** Robin K. S. Hankin

**Description** Various utilities for the Davies distribution.

**Maintainer** Robin K. S. Hankin <hankin.robin@gmail.com>

**License** GPL-2

**Repository** CRAN

**Date/Publication** 2016-05-26 09:21:09

**NeedsCompilation** no

## R topics documented:

Davies . . . . .	2
davies.moment . . . . .	3
davies.start . . . . .	5
expected.gld . . . . .	6
fit.davies.p . . . . .	7
Gld . . . . .	8
least.squares . . . . .	9
likelihood . . . . .	10
objective . . . . .	11
ozturk . . . . .	12
plotcf . . . . .	13
rstupid . . . . .	14
twolines.vert . . . . .	15
x00m700p4 . . . . .	16
<b>Index</b>	<b>17</b>

---

Davies

*The Davies distribution*

---

### Description

Density, distribution function, quantile function and random generation for the Davies distribution

### Usage

```
ddavies(x, params)
pdavies(x, params)
qdavies(p, params)
rdavies(n, params)
ddavies.p(x, params)
```

### Arguments

x	quantile
p	vector of probabilities
n	number of observations. If <code>length(n) &gt; 1</code> , the length is taken to be the number required
params	A three-member vector holding $C$ , $\lambda_1$ and $\lambda_2$

### Details

The Davies distribution is defined in terms of its quantile function:

$$Cp^{\lambda_1}/(1-p)^{\lambda_2}$$

It does not have a closed-form probability density function or cumulative density function, so numerical solution is used.

### Value

Function `ddavies()` gives the density, `pdavies()` gives the distribution function, `qdavies()` gives the quantile function, and `rdavies()` generates random deviates.

### Author(s)

Robin K. S. Hankin

### References

R. K. S. Hankin and A. Lee 2006. "A new family of non-negative distributions" *Australia and New Zealand Journal of Statistics*, 48(1):67–78

**See Also**

[Gld](#), [fit.davies.p](#), [least.squares](#), [skewness](#)

**Examples**

```

params <- c(10,0.1,0.1)
x <- seq(from=4,to=20,by=0.2)
p <- seq(from=1e-3,to=1-1e-3,len=50)

rdavies(n=5,params)
least.squares(rdavies(100,params))
plot(pdavies(x,params))

plot(p,qdavies(p,params))
plot(x,ddavies(x,params),type="b")

```

---

davies.moment

*Moments of the Davies distribution*

---

**Description**

Moments of order statistics of random variables drawn from a Davies distribution

**Usage**

```

davies.moment(n=1 , i=1 , order=1 , params)
M(order,params)
mu(params)
expected.value(n,i,params)
expected.value.approx(n,i,params)
variance(params)
skewness(params)
kurtosis(params)

```

**Arguments**

params	A three-member vector holding $C$ , $\lambda_1$ and $\lambda_2$
n	The number of observations
i	Return information about the $i$ -th order statistic (ie $i = 1$ means the smallest, $i = n$ means the biggest)
order	The order (eg order=2 gives the square)

## Details

Function `davies.moment(n,i,order=r)` gives the  $r$ -th moment of the  $i$ -th order statistic of  $n$  observations. The following aliases are just newbie wrappers with  $n = i = 1$  (ie moments of one observation from a Davies distribution):

- `M()` gives the  $r$ -th moment for  $n = i = 1$
- `mu()` gives the first moment of a Davies distribution (ie the mean)
- `variance()` gives the second *central* moment of a Davies distribution
- `skewness()` gives the normalized skewness of a Davies distribution
- `kurtosis()` gives the normalized kurtosis of a Davies distribution

## Author(s)

Robin K. S. Hankin

## See Also

[expected.value](#), [expected.gld](#)

## Examples

```
params <- c(10,0.1,0.1)
davies.moment(n=100,i=99,2,params) # ie the second moment of the 99th smallest
                                   # observation of 100 drawn from a Davies
                                   # distribution with parameters p

mean(rdavies(1e6,params))-mu(params)

#now reproduce the S-K graph:

f <- function(x,y){c(skewness(c(1,x,y)),kurtosis(c(1,x,y)))}
g <- function(j,vector,pp,qq=1){points(t(sapply(vector,f,y=j)),type="l",col="black",lty=qq)}

vector <- c((0:300)/100 , (0:300)/10000 , seq(from=3,to=10,len=100))
vector <- sort(unique(vector))

plot(t(sapply((0:10)/10,f,y=0)),
      xlim=c(-3,3),ylim=c(0,10),
      type="n",xlab="skewness",ylab="kurtosis")
g(0.001,vector,"red",qq=1)
g(0.01,vector,"yellow",qq=2)
g(0.02,vector,"green",qq=3)
g(0.05,vector,"blue",qq=4)
g(0.1 ,vector,"purple",qq=5)
g(0.14,vector,"black",qq=6)

x <- seq(from=-3,to=3,len=30)
points(x,x^2+1,type="l",lwd=2)
```

```
leg.txt <- expression(lambda[2]==0.001,
                      lambda[2]==0.01,lambda[2]==0.02,lambda[2]==0.05,
                      lambda[2]==0.1,lambda[2]==0.14)
legend(-1.1,10,leg.txt,col="black",lty=1:6)
```

---

davies.start	<i>start value for Davies minimization routines</i>
--------------	---

---

### Description

Gives a “start” value for the optimization routines. Uses heuristics that seem to work.

### Usage

```
davies.start(x, threeps=c(0.1,0.5,0.9), small = 0.01)
```

### Arguments

x	dataset to be used
threeps	a three-element vector representing the quantiles to be balanced. The default values balance the first and ninth deciles and the median. These seem to work for me pretty well; YMMV
small	a “small” value to be used for $\lambda_1$ and $\lambda_2$ because using exactly zero is inappropriate

### Details

Returns a “start” value of the parameters for use in one of the Davies fitting routines `maximum.likelihood()` or `least.squares()`.

Uses three heuristic methods (one assuming  $\lambda_1 = \lambda_2$ , one with  $\lambda_1 = 0$ , and one with  $\lambda_2 = 0$ ). Returns the best one of the three, as measured by `objective()`.

### Author(s)

Robin K. S. Hankin

### See Also

[least.squares](#), [maximum.likelihood](#), [objective](#)

**Examples**

```

d <- rchisq(40,1)
davies.start(d)
least.squares(d)

params <- c(10 , 0.1 , -0.1)
x <- rdavies(100 , params)
davies.start(x)

f <- function(threeps){objective(davies.start(x, threeps),x)}

(jj<-optim(c(0.1,0.5,0.9),f))
davies.start(x,jj$par)
least.squares(x)

#not bad at all.

```

---

expected.gld

*expected value of the Generalized Lambda Distribution*

---

**Description**

Returns the expected value of the Generalized Lambda Distribution

**Usage**

```

expected.gld(n=1, i=1, params)
expected.gld.approx(n=1, i=1, params)

```

**Arguments**

n	Number of observations
i	Order statistic: $i = 1$ means the smallest of $n$ , and $n = i$ means the largest
params	The four parameters of a GLD distribution

**Details**

expected.gld and expected.approx return the exact and approximate values of the expected value of a Generalized Lambda Distribution RV.

Exploits the fact that the gld quantile function is the sum of a constant and two davies quantile functions

**Author(s)**

Robin K. S. Hankin

**References**

A. Ozturk and R. F. Dale, “Least squares estimation of the parameters of the generalized lambda distribution”, *Technometrics* 1985, 27(1):84 [it does not appear to be possible, as of R-2.9.1, to render the diacritic marks in the first author’s names in a nicely portable way]

**See Also**

[Gld](#), [expected.value](#)

**Examples**

```
params <- c(4.114,0.1333,0.0193,0.1588)
mean(rgld(1000,params))
expected.gld(n=1,i=1,params)
expected.gld.approx(n=1,i=1,params)

f <- function(n){apply(matrix(rgld(n+n,params),2,n),2,min)}
#ie f(n) gives the smaller of 2 rgld RVs, n times.

mean(f(1000))
expected.gld(n=2,i=1,params)
expected.gld.approx(n=2,i=1,params)

plot(1:100,expected.gld.approx(n=100,i=1:100,params)-expected.gld(n=100,i=1:100,params))
# not bad, eh? ...yyeeeeesss, but the parameters given by Ozturk give
# an almost zero second derivative for d(qgld)/dp, so the good agreement
# isn't surprising really. Observe that the error is minimized at about
# p=0.2, where the point of inflection is.
```

---

fit.davies.p

*Fits and plots Davies distributions to datasets*


---

**Description**

A newbie wrapper (and pretty-printer) for `maximum.likelihood()` and `least.squares()`. Draws an empirical quantile function (`fit.davies.p()`) or PDF (`fit.davies.q()`) and the dataset

**Usage**

```
fit.davies.p(x , print.fit=FALSE, use.q=TRUE , params=NULL, small=1e-5 , ...)
fit.davies.q(x , print.fit=FALSE, use.q=TRUE , params=NULL, ...)
```

**Arguments**

x	dataset to be fitted and plotted
print.fit	Boolean with TRUE meaning print details of the fit
use.q	Boolean with TRUE meaning use <code>least.squares()</code> (rather than <code>maximum.likelihood()</code> )

params	three-element vector holding the three parameters of the davies dataset. If NULL, determine the parameters using the method indicated by use.q
small	small positive number showing range of quantiles to plot
...	Additional parameters passed to plot()

**Value**

If `print.fit` is TRUE, return the optimal parameters

**Author(s)**

Robin K. S. Hankin

**See Also**

[least.squares](#), [maximum.likelihood](#)

**Examples**

```
fit.davies.q(rchisq(100,1))
fit.davies.p(exp(rnorm(100)))

data(x00m700p4)
fit.davies.q(x00m700p4)
```

---

Gld

*The Generalized Lambda Distribution*


---

**Description**

Density, distribution function, quantile function and random generation for the Generalized Lambda Distribution

**Usage**

```
dgld(x, params)
dglp(x, params)
pgld(q, params)
qgld(p, params)
rgld(n, params)
```

**Arguments**

x, q	vector of quantiles
p	vector of probabilities
n	In function <code>rgld()</code> , the number of observations. If <code>length(n) &gt; 1</code> , the length is taken to be the number required
params	vector of parameters: <code>params[1]==lambda1</code> et seq



**Details**

The Generalized Lambda distribution has quantile function

$$f(x) = \lambda_1 + (p^{\lambda_3} - (1-p)^{\lambda_4})/\lambda_2$$

**Value**

Function `dgld()` gives the density, `dgld.p()` gives the density in terms of the quantile, `pgld()` gives the distribution function, `qgld()` gives the quantile function, and `rgld()` generates random deviates.

**References**

- M. J. Wichura 1988. “Algorithm AS 241: The Percentage Points of the Normal Distribution”. *Applied Statistics*, **37**, 477–484.
- A. Ozturk and R. F. Dale 1985. “Least squares estimation of the parameters of the generalized lambda distribution”. *Technometrics* 27(1):84

**See Also**

[Davies, expected.gld](#)

**Examples**

```
params <- c(4.114,0.1333,0.0193,0.1588) #taken straight from some paper
gld.rv <- rgld(100,params)

hist(gld.rv)
fit.davies.q(gld.rv) #remember the Davies distn has 3 DF and the GLD 4...
```

---

least.squares

*Finds the optimal Davies distribution for a dataset*

---

**Description**

Finds the best-fit Davies distribution using either the least-squares criterion (`least.squares()`) or maximum likelihood (`maximum.likelihood()`)

**Usage**

```
least.squares(data, do.print = FALSE, start.v = NULL)
maximum.likelihood(data, do.print = FALSE, start.v = NULL)
```

**Arguments**

<code>data</code>	dataset to be fitted
<code>do.print</code>	Boolean with TRUE meaning print a GFM
<code>start.v</code>	A suitable starting vector of parameters $c(C, \lambda_1, \lambda_2)$ , with default NULL meaning to use <code>start()</code>

**Details**

Uses `optim()` to find the best-fit Davies distribution to a set of data.

**Value**

Returns the parameters  $C, \lambda_1, \lambda_2$  of the best-fit Davies distribution to the dataset `data`

**Note**

BUGS: can be screwed with bad value for `start.v`. Function `maximum.likelihood()` can be very slow. It might be possible to improve this by using some sort of hot-start for `optim()`.

**Author(s)**

Robin K. S. Hankin

**See Also**

[davies.start](#), [optim](#), [objective](#), [likelihood](#)

**Examples**

```
p <- c(10 , 0.1 , 0.1)
d <- rdavies(50,p)
system.time(print(maximum.likelihood(d)))
#observe how long this takes.
#The time is taken in repeated calls
#to pdavies(), which uses uniroot().

system.time(print(least.squares(d)))
#Much faster.
```

---

likelihood

*likelihood for the Davies distribution*

---

**Description**

Likelihood of observing data, on the hypothesis of their coming from a Davies distribution of parameters `params`.

Function `neg.log.likelihood()` gives minus the loglikelihood

**Usage**

```
likelihood(params, data)
```

**Arguments**

params	Parameters of the Davies distribution
data	dataset for which the likelihood is computed

**Author(s)**

Robin K. S. Hankin

**See Also**

[Davies](#)

**Examples**

```
p1 <- c(10, 0.1, 0.1)
p2 <- c(10, 0.4, 0.1)
d <- rdavies(100,p1)
likelihood(p1,d)
likelihood(p2,d) #should be smaller.
neg.log.likelihood(p1,rstupid(100)) #should be large negative.
```

---

objective

*The objective function for fitting the Davies distribution*

---

**Description**

The “distance” of a dataset from a particular Davies distribution

**Usage**

```
objective(params, dataset)
objective.approx(params, dataset)
```

**Arguments**

params	A three-member vector holding $C$ , $\lambda_1$ and $\lambda_2$
dataset	The dataset to be considered

**Details**

Used by the `fit.davies.p()` and `fit.davies.q()` functions

**Value**

`objective` returns the “distance” of a dataset from a particular Davies distribution as measured by the sums of the squares of the differences between observed (`dataset`) and expected (`expected.value()`) values.

`objective.approx()` uses `expected.approx()` rather than `expected()` to calculate expectations, as per equation 6.

**Author(s)**

Robin K. S. Hankin

**See Also**

[fit.davies.p](#), [fit.davies.q](#)

**Examples**

```
params <- c(10, 0.1, 0.1)
x <- rdavies(100, params)
objective(params, x)
objective.approx(params, x)

objective(least.squares(x), x)
objective(davies.start(x), x)
```

---

ozturk

*Parameters used in a paper by Ozturk*

---

**Description**

A four-element vector giving the parameters used by Ozturk.

**Usage**

```
data(x00m700p4)
```

**References**

A. Ozturk and R. F. Dale 1985. “Least squares estimation of the parameters of the generalized lambda distribution”. *Technometrics* 27(1):84; see discussion under `expected.gld.Rd`.

**See Also**

[expected.gld](#)

**Examples**

```
data(ozturk)
hist(rgld(100,ozturk))
```

---

plotcf *p-value investigation*

---

**Description**

Plots sorted p-values showing which ones would have been rejected

**Usage**

```
plotcf(y, q=0.05)
```

**Arguments**

y	dataset
q	p-value of critical region

**Details**

Sorts p-values and plots the order statistic. Useful for investigating a statistical test by using it when the null hypothesis is KNOWN to be true, just to check if the probability of rejection really is alpha.

Also can be used when H0 is wrong, showing what beta is.

**Author(s)**

Robin K. S. Hankin

**Examples**

```
f.H0.T <- function(n,free=5){t.test(rt(n,df=free))$p.value}
f.H0.F <- function(n,free=5){t.test(rf(n,df1=free,df2=free))$p.value}

plotcf(sapply(rep(10,100),f.H0.T)) # should reject about 5: thus
# probability of a type I error is
# about 0.05 (as it should be; this
# is an exact test)

plotcf(sapply(rep(10,100),f.H0.F)) # should reject about 80: thus
# probability of a type II error is
# about 0.2 for this H_A.
```

---

`rstupid`*A stupid PDF*

---

**Description**

a contrived PDF that cannot be closely approximated by a Davies distribution

**Usage**

```
rstupid(n, a = 1, b = 2, c = 3, d = 4)
```

**Arguments**

<code>n</code>	Number of observations
<code>a</code>	start of first uniform bit
<code>b</code>	end of first uniform bit
<code>c</code>	start of second uniform bit
<code>d</code>	end of second uniform bit

**Details**

The stupid distribution is composed of two separate uniform distributions: one from  $a$  to  $b$ , and one from  $c$  to  $d$ . It is specifically designed to be NOT fittable to any Davies distribution.

You could probably come up with a more stupid distribution if you tried

**Author(s)**

Robin K. S. Hankin

**See Also**

[Davies](#)

**Examples**

```
stupid <- rstupid(500)
fit.davies.q(stupid)
```

---

twolines.vert	<i>Order statistic comparison</i>
---------------	-----------------------------------

---

### Description

Plots two lines and shades the bit in between them

### Usage

```
twolines.vert(p, y1, y2, ...)
```

### Arguments

p	vector of quantiles
y1	First set of ordinates
y2	Second set of ordinates
...	Extra arguments, passed to <code>segments()</code> , for the vertical lines

### Details

Plots p against y1, and p against y2, and shades the bit in between using vertical lines. This is useful for comparing two order statistics

### Author(s)

Robin K. S. Hankin

### See Also

[Davies,qqplot](#)

### Examples

```
twolines.vert(1:100,sort(rnorm(100)),sort(rnorm(100)))  
params <- c(10 , 0.1 , 0.1)  
twolines.vert(1:100 , sort(rdavies(100,params)) , sort(rdavies(100,params)))
```

---

`x00m700p4`*Peak concentration for 100 instantaneous releases*

---

**Description**

This data set gives the peak concentration for 100 independent instantaneous releases of neutral-buoyancy gas in a windtunnel

**Usage**

```
data(x00m700p4)
```

**Format**

A vector containing 100 observations

**References**

D. J. Hall and others 1991. *Repeat variability in instantaneously released heavy gas clouds—some wind tunnel model experiments*. Technical Report LR 804 (PA), Warren Spring Laboratory, Gunnels Wood Road, Stevenage, Hertfordshire SG1 2BX.

**Examples**

```
data(x00m700p4)  
fit.davies.q(x00m700p4)
```



# Index

## \*Topic **datasets**

ozturk, [12](#)  
x00m700p4, [16](#)

## \*Topic **distribution**

Davies, [2](#)  
davies.moment, [3](#)  
davies.start, [5](#)  
expected.gld, [6](#)  
fit.davies.p, [7](#)  
Gld, [8](#)  
least.squares, [9](#)  
likelihood, [10](#)  
objective, [11](#)  
plotcf, [13](#)  
rstupid, [14](#)  
twolines.vert, [15](#)

Davies, [2](#), [9](#), [11](#), [14](#), [15](#)  
davies (Davies), [2](#)  
davies.moment, [3](#)  
davies.start, [5](#), [10](#)  
ddavies (Davies), [2](#)  
dgld (Gld), [8](#)

expected.gld, [4](#), [6](#), [9](#), [12](#)  
expected.value, [4](#), [7](#)  
expected.value (davies.moment), [3](#)

fit.davies.p, [3](#), [7](#), [12](#)  
fit.davies.q, [12](#)  
fit.davies.q (fit.davies.p), [7](#)

Gld, [3](#), [7](#), [8](#)  
gld (Gld), [8](#)

kurtosis (davies.moment), [3](#)

least.squares, [3](#), [5](#), [8](#), [9](#)  
likelihood, [10](#), [10](#)

M (davies.moment), [3](#)

maximum.likelihood, [5](#), [8](#)  
maximum.likelihood (least.squares), [9](#)  
mu (davies.moment), [3](#)

neg.log.likelihood (likelihood), [10](#)

objective, [5](#), [10](#), [11](#)  
optim, [10](#)  
ozturk, [12](#)

pdavies (Davies), [2](#)  
pgld (Gld), [8](#)  
plotcf, [13](#)

qdavies (Davies), [2](#)  
qgld (Gld), [8](#)  
qqplot, [15](#)

rdavies (Davies), [2](#)  
rgld (Gld), [8](#)  
rstupid, [14](#)

skewness, [3](#)  
skewness (davies.moment), [3](#)

twolines.vert, [15](#)

variance (davies.moment), [3](#)

x00m700p4, [16](#)