Package 'CompDist'

October 12, 2022

Type Package

Title Multisection Composite Distributions

Version 1.0

Author Martin Wiegand and Saralees Nadarajah

Date 2017-08-16

Maintainer Saralees Nadarajah <Saralees.Nadarajah@manchester.ac.uk>

Depends R (>= 3.3.0), numDeriv

Imports fExtremes, actuar, VGAM, rmutil, PearsonDS

Description Computes density function, cumulative distribution function, quantile function and random numbers for a multisection composite distribution specified by the user. Also fits the user specified distribution to a given data set. More details of the package can be found in the following paper submitted to the R journal Wiegand M and Nadarajah S (2017) CompDist: Multisection composite distributions.

License GPL (>= 2)

NeedsCompilation no

Repository CRAN

Date/Publication 2017-08-17 11:38:03 UTC

R topics documented:

dcomp *dcomp*

Description

Returns a density function of a user specified multisection composite distribution

Usage

```
dcomp(xx,dists,par,borders,par.pos,buffer)
```

Arguments

xx	Evaluation locations
dists	A vector of strings stating the desired partial distributions in order
par	A list of parameters, a vector of parameters for each partial distribution, with the first two being the interval limits and the second argument being the weights to be used
borders	Optional: If the distribution has to have continuous and differentiable catentation points, the user specifies a list for each of area following the first, containing a range for a parameter of the following partial distribution to lie within.
par.pos	Optional: If 'borders' is non empty, e.g a smooth function is desired, here the vector of parameter positions that need to be changed is specified. Default value is 1, meaning the first parameter for each partial distribution is amended
buffer	Optional: A two dimensional vector, containing the values for upper and lower buffer from the respective catenation points during optimization

Value

An object of the same length as xx, giving the density values

Author(s)

Martin Wiegand, Saralees Nadarajah

Examples

```
par<-list()
distvec<-c("lnorm","gamma")
par[[1]]<-c(0,1,Inf)
par[[2]]<-c(1)
par[[3]]<-c(0,1)
par[[4]]<-c(1,1)

x<-seq(0,3,0.01)
# non-continuous case
y1<-dcomp(x,distvec,par)
```

```
# continuous case
y2<-dcomp(x,distvec,par,borders=list(c(0.00001,10)),buffer=c(10e-5,0))

par(mfrow=c(1,2),oma=rep(0,4))
xrange<-range(x)
yrange<-range(y1,y2)
plot(x,y1,type="l",xlab="x",ylab="Density function",xlim=xrange,ylim=yrange)
abline(v=1)
plot(x,y2,type="l",xlab="x",ylab="Density function",xlim=xrange,ylim=yrange)
abline(v=1)
```

`par.fit` *par.fit*

Description

Returns the parameters fitted to a random sample along with a number of error measures, such as the log likelihood, AIC, BIC, AICc, CAIC and HQC.

Usage

```
par.fit(data,dists,par,borders,par.pos,optit,buffer,cont)
```

Arguments

`data`	Data set to be fitted to the distribution
`dists`	A vector of strings stating the desired partial distributions in order
`par`	A list of parameters, a vector of parameters for each partial distribution, with the first two being the interval limits and the second argument being he weights to be used
`borders`	Optional: If the distribution has to have continuous and differentiable catentation points, the user specifies a list for each of area following the first, containing a range for a parameter of the following partial distribution to lie within.
`par.pos`	Optional: If 'borders' is non empty, e.g a smooth function is desired, here the vector of parameter positions that need to be changed is specified. Default value is 1, meaning the first parameter for each partial distribution is amended
`optit`	Number of iteration loops over the parameter optimisation
`buffer`	Optional: A two dimensional vector, containing the values for upper and lower buffer from the respective catenation points during optimization
`cont`	Logical value for smooth catenation points. Default FALSE.

Value

Gives parameter estimates and values of the log likelihood, AIC, BIC, AICc, CAIC and HQC.

Author(s)

Martin Wiegand, Saralees Nadarajah

Examples

```
# Generate random data


par<-list()

distvec<-c("lnorm","gamma")

par[[1]]<-c(0,1,Inf)

par[[2]]<-c(1)

par[[3]]<-c(0,1)

par[[4]]<-c(1,1)


n<-1000

# non-continuous case

r1<-rcomp(n,distvec,par)

# continuous case

r2<-rcomp(n,distvec,par,borders=list(c(0.00001,10)),buffer=c(10e-5,0))


# Initial Guess


par<-list()

distvec<-c("lnorm","gamma")

par[[1]]<-c(0,1,Inf)

par[[2]]<-c(1)

par[[3]]<-c(0,0.5)

par[[4]]<-c(0.5,1)


# Fitting
```

```
# non-continuous case

estimate1<-par.fit(r1,distvec,par,optit=1)

# continuous case

estimate2<-par.fit(r2,distvec,par,borders=list(c(0.00001,10)),optit=1,buffer=c(10e-5,0),cont=TRUE)


x<-seq(0,30,0.01)

# non-continuous case

y1<-dcomp(x,distvec,estimate1$Parameter)

# continuous case

y2<-dcomp(x,distvec,estimate2$Parameter,borders=list(c(0.00001,10)),buffer=c(10e-5,0))


par(mfrow=c(1,2),oma=rep(0,4))

hist(r1,probability=TRUE,breaks=40,main="",xlab="Data",ylab="Fitted density")

lines(x,y1,col="red")

hist(r2,probability=TRUE,breaks=40,main="",xlab="Data",ylab="Fitted density")

lines(x,y2,col="red")


estimate1

estimate2
```

---

pcomp *pcomp*

Description

Returns a cumulative distribution function of a user specified multisection composite distribution

Usage

```
pcomp(xx,dists,par,borders,par.pos,buffer)
```

Arguments

xx	Evaluation locations
dists	A vector of strings stating the desired partial distributions in order
par	A list of parameters, a vector of parameters for each partial distribution, with the first two being the interval limits and the second argument being he weights to be used
borders	Optional: If the distribution has to have continuous and differentiable catentation points, the user specifies a list for each of area following the first, containing a range for a parameter of the following partial distribution to lie within.
par.pos	Optional: If 'borders' is non empty, e.g a smooth function is desired, here the vector of parameter positions that need to be changed is specified. Default value is 1, meaning the first parameter for each partial distribution is amended
buffer	Optional: A two dimensional vector, containing the values for upper and lower buffer from the respective catenation points during optimization

Value

An object of the same length as xx, giving the cumulative distribution function values

Author(s)

Martin Wiegand, Saralees Nadarajah

Examples

```
par<-list()

distvec<-c("lnorm","gamma")

par[[1]]<-c(0,1,Inf)

par[[2]]<-c(1)

par[[3]]<-c(0,1)

par[[4]]<-c(1,1)


x<-seq(0,3,0.01)

# non-continuous case

y1<-pcomp(x,distvec,par)

# continuous case

y2<-pcomp(x,distvec,par,borders=list(c(0.00001,10)),buffer=c(10e-5,0))
```

```
par(mfrow=c(1,2),oma=rep(0,4))

xrange<-range(x)

yrange<-range(y1,y2)

plot(x,y1,type="l",xlab="x",ylab="Distribution function",xlim=xrange,ylim=yrange)

abline(v=1,lty=2)

plot(x,y2,type="l",xlab="x",ylab="Distribution function",xlim=xrange,ylim=yrange)

abline(v=1,lty=2)
```

---

qcomp *qcomp*

Description

Returns a quantile function to the specifications of a user specified multisection composite distribution

Usage

```
qcomp(xx,dists,par,borders,par.pos,buffer)
```

Arguments

xx	Desired quantiles between 0 and 1
dists	A vector of strings stating the desired partial distributions in order
par	A list of parameters, a vector of parameters for each partial distribution, with the first two being the interval limits and the second argument being he weights to be used
borders	Optional: If the distribution has to have continuous and differentiable catentation points, the user specifies a list for each of area following the first, containing a range for a parameter of the following partial distribution to lie within.
par.pos	Optional: If 'borders' is non empty, e.g a smooth function is desired, here the vector of parameter positions that need to be changed is specified. Default value is 1, meaning the first parameter for each partial distribution is amended
buffer	Optional: A two dimensional vector, containing the values for upper and lower buffer from the respective catenation points during optimization

Value

An object of the same length as xx, giving the quantile values

Author(s)

Martin Wiegand, Saralees Nadarajah

Examples

```
par<-list()

distvec<-c("lnorm","gamma")

par[[1]]<-c(0,1,Inf)

par[[2]]<-c(1)

par[[3]]<-c(0,1)

par[[4]]<-c(1,1)


x<-seq(0.01,0.99,0.01)

# non-continuous case

y1<-qcomp(x,distvec,par)

# continuous case

y2<-qcomp(x,distvec,par,borders=list(c(0.00001,10)),buffer=c(10e-5,0))


par(mfrow=c(1,2),oma=rep(0,4))

xrange<-range(x)

yrange<-range(y1,y2)

plot(x,y1,type="l",xlab="x",ylab="Quantile function",xlim=xrange,ylim=yrange)

abline(h=1,lty=2)

plot(x,y2,type="l",xlab="x",ylab="Quantile function",xlim=xrange,ylim=yrange)

abline(h=1,lty=2)
```

---

rcomp *rcomp*

Description

Returns a random sample of size n of a user specified multisection composite distribution

Usage

```
rcomp(nn,dists,par,borders,par.pos,buffer)
```

Arguments

nn	Desired random sample size
dists	A vector of strings stating the desired partial distributions in order
par	A list of parameters, a vector of parameters for each partial distribution, with the first two being the interval limits and the second argument being he weights to be used
borders	Optional: If the distribution has to have continuous and differentiable catentation points, the user specifies a list for each of area following the first, containing a range for a parameter of the following partial distribution to lie within.
par.pos	Optional: If 'borders' is non empty, e.g a smooth function is desired, here the vector of parameter positions that need to be changed is specified. Default value is 1, meaning the first parameter for each partial distribution is amended
buffer	Optional: A two dimensional vector, containing the values for upper and lower buffer from the respective catenation points during optimization

Value

An object of length nn, giving the random numbers

Author(s)

Martin Wiegand, Saralees Nadarajah

Examples

```
par<-list()

distvec<-c("lnorm","gamma")

par[[1]]<-c(0,1,Inf)

par[[2]]<-c(1)

par[[3]]<-c(0,1)

par[[4]]<-c(1,1)


n<-1000

# non-continuous case

y1<-rcomp(n,distvec,par)

# continuous case
```

```
y2<-rcomp(n,distvec,par,borders=list(c(0.00001,10)),buffer=c(10e-5,0))


par(mfrow=c(1,2),oma=rep(0,4))

hist(y1,nclass=10,xlab="x",ylab="Frequency",main="")

hist(y2,nclass=10,xlab="x",ylab="Frequency",main="")
```

Index