Package ‘TRIANGG’

February 14, 2012

Type Package

Title General discrete triangular distribution

Version 1.0

Date 2010-08-18

Author Tristan Senga Kiess\{}'{e}, Francial G. Libengu\{}'e, Silvio S. Zocchi, C\{}'{e}lestin C. Kokonendji

Maintainer Tristan Senga Kiess\{}'{e} <tristan.senga-kiesse@lcpc.fr>

Description This is a package for general discrete triangular distributions which is an extension of the symmetric discrete triangular distributions

License GPL-2

LazyLoad yes

Repository CRAN

Date/Publication 2010-08-22 07:57:44

R topics documented:

`dtrgg` *General discrete triangular distributions*

Description

The function plots general discrete triangular distributions

Usage

```
dtrgg(m,a1,a2,h1,h2,y)
```

Arguments

`m`	The mode \$m\$ is an integer
`a1, a2`	The left arm \$a_1\$ and right arm \$a_2\$ are non-negative integers
`h1, h2`	The left order \$h_1\$ and right order \$h_2\$ are positive real numbers
`y`	The vector of entire observations

Details

The general discrete triangular distribution has the probability mass function

\$ Pr(Y=y) = (1/D)[1-{(m-y)/(a_1+1)}^h_1]1_{m-a_1,...,m-2,m-1}(y)\$

\$+ (1/D)[1-{(y-m)/(a_2+1)}^h_2]1_{m,m+1,...,m+a_2}(y) \$

with the normalizing constant \$D=(a_1+a_2+1) - (a_1+1)^-h_1sum_k=1^a_1k^h_1 - (a_2+1)^-h_2sum_k=1^a_2k^h_2\$.

Value

The function returns the probability mass function in [0,1] of the corresponding \$y\$ value.

Author(s)

Tristan Senga Kiess\'e, Francial G. Libengu\'e, Silvio S. Zocchi, C\'elestin C. Kokonendji

References

Kokonendji, C.C. and Zocchi, S.S. (2010). Extensions of discrete triangular distributions and boundary bias in kernel estimation for discrete functions. Statistics and Probability Letters, 80, 1655–1662.

Kokonendji, C.C., Senga Kiess\'e, T. and Zocchi, S.S. (2007). Discrete triangular distributions and non-parametric estimation for probability mass function. Journal of Nonparametric Statistics, 19, 241–254.

See Also

'dtrg' for symmetric discrete triangular distributions

Examples

```
#The following example illustrates different shapes of general
#discrete triangular distributions :
y=-3:15
ym2_10=y
m=2
h1=1/2
h2=1/3
a1=4
a2=12
Tm2=dtrgg(m,a1,a2,h1,h2,y)

m=10
h1=1/2
h2=1/3
a1=12
a2=4
Tm10=dtrgg(m,a1,a2,h1,h2,y)

y=3:21
ym9_15=y
m=9
h1=1/2
h2=3
a1=5
a2=11
Tm9=dtrgg(m,a1,a2,h1,h2,y)

m=15
h1=1/2
h2=3
a1=11
a2=5
Tm15=dtrgg(m,a1,a2,h1,h2,y)

par(mfrow=c(1,2))

plot(ym2_10,Tm2,xlab="y",ylab="P(Y=y)", ylim=c(0,0.3),main="h1=1/2 and h2=1/3",pch=20,
col="grey",cex.lab=1.5,cex.axis=1.5,lwd=2)
lines(ym2_10,Tm2,lty=1,col="grey",lwd=2)
points(ym2_10,Tm10,xlab="y",ylab="P(Y=y)", main="h1=1/2 and h2=1/3",pch=20,
col="black",cex.lab=1.5,cex.axis=1.5,lwd=2)
lines(ym2_10,Tm10,lty=2,lwd=2)
op <- par(bg="white")
legend(0,0.3,c("m=2,a1=4,a2=12", "m=10,a1=12,a2=4"),
pch=c(20,20), lty=c(1,2),col=c("grey","black"),cex =1,lwd=c(2,2))


plot(ym9_15,Tm9,xlab="y",ylab="P(Y=y)",ylim=c(0,0.2), main="h1=1/2 and h2=3",pch=20,
col="grey",cex.lab=1.5,cex.axis=1.5)
lines(ym9_15,Tm9,lty=1,col="grey",lwd=2)
points(ym9_15,Tm15,xlab="y",ylab="P(Y=y)", main="h1=1/2 and h2=3",pch=20,
```

```
col="black",cex.lab=1.5,cex.axis=1.5)
lines(ym9_15,Tm15,lty=2,lwd=2)
op <- par(bg="white")
legend(5,0.2,c("m=9,a1=5,a2=11", "m=15,a1=11,a2=5"),
pch=c(20,20), lty=c(1,2),col=c("grey","black"),cex =1,lwd=c(2,2))



## The function is defined as
function(m,a1,a2,h1,h2,y){
T=rep(0,length(y));

if ((a1==0)&(a2==0))
 {
    {for (j in 1:length(y))             # Loop in j for each observation y

      {if (y[j]==m)
        T[j]= 1  # Dirac distribution at m

       else{
           T[j]=0
            }
      }
    }
  }


else if ((h1==0)&(h2==0))
 {
    {for (j in 1:length(y))

      {if (y[j]==m)
        T[j]= 1  # Dirac distribution at m

       else{
           T[j]=0
            }
      }
    }
  }



else if ((h1==Inf)&(h2==Inf))
{
   {for (j in 1:length(y))

     {if (y[j]>=(m-a1) & y[j]<=(m+a2) & y[j]==as.integer(y[j]))
           #  Support {m-a1,...,m,...m+a2}

        T[j]= 1/(a1+a2+1)
    # Discrete uniform distribution
```

```
       else{
           T[j]=0
           }
      }
    }
  }


else if ((h1==Inf)&(h2==0))
{
   {for (j in 1:length(y))

     {if (y[j]>=(m-a1) & y[j]<=m & y[j]==as.integer(y[j]))
          #  Support {m-a1,...,m-1,m}

 T[j]= 1/(a1+1)
  # Discrete uniform distribution

       else{
           T[j]=0
           }
      }
    }
  }

else if ((h1==0)&(h2==Inf))
{
   {for (j in 1:length(y))

     {if (y[j]>=m & y[j]<=(m+a2) & y[j]==as.integer(y[j]))
           #  Support {m,m+1,...,m+a2}

T[j]= 1/(a2+1)
   # Discrete uniform distribution

       else{
           T[j]=0
           }
      }
    }
  }

else
{
 u1=0;
 u2=0;


 {for (k in 1:a1)

   {
    u1=u1+k^h1
    }
```

```
  }

 {for (k in 1:a2)

   {
    u2=u2+k^h2
    }

  }


  D=(a1+a2+1)-(a1+1)^(-h1)*u1 -(a2+1)^(-h2)*u2               # Normalizing constant



  {for (j in 1:length(y))

      {if (y[j]>=(m-a1) & y[j]<=(m-1)& y[j]==as.integer(y[j]))
              #  Support {m-a1,...,m-2,m-1}

T[j]= (1 - ((m-y[j])/(a1+1))^h1)/D
            # Discrete triangular distribution a1, h1


      else if (y[j]>=m & y[j]<=(m+a2)& y[j]==as.integer(y[j]))
            #  Support {m,m+1,...m+a2}

        T[j]= (1 - ((y[j]-m)/(a2+1))^h2)/D
    # Discrete triangular distribution a2, h2


       else{
           T[j]=0
            }
      }
   }
}

return(T)
}
```

Index