Package ‘gamlss.countKinf’

November 14, 2018

Type Package
Title Generating and Fitting K-Inflated ‘discrete gamlss.family’ Distributions
Version 3.5.1
Author Saeed Mohammadpour <s.mohammadpour1111@gmail.com>, Mikis Stasinopoulos <d.stasinopoulos@londonmet.ac.uk>
Maintainer Saeed Mohammadpour <s.mohammadpour1111@gmail.com>
Depends R (>= 2.2.1), gamlss.dist, gamlss (>= 5.0-0), stats
Description This is an add on package to 'GAMLSS'. The main purpose of this package is generating and fitting inflated distributions at any desired point (0, 1, 2, ...). The function gen.Kinf() generates K-inflated version of an existing discrete 'GAMLSS' family distribution.
License GPL-2 | GPL-3
LazyData True
URL http://www.gamlss.org/
NeedsCompilation no
Repository CRAN
Date/Publication 2018-11-14 10:30:03 UTC

R topics documented:

  gamlss.countKinf-package ........................................ 2
gen.Kinf ................................................................. 3
KIBNB ................................................................. 4
KIDEL ................................................................. 7
KIDPO ................................................................. 10
KIGEOM ............................................................... 13
KIGEOMo .............................................................. 15
KIGPO ................................................................. 18
KILG ................................................................. 21
KINBF ................................................................. 24
Generating and Fitting K-Inflated 'discrete gamlss.family' Distributions

Description

The main purpose of this package is to allow the user of the GAMLSS models to fit K-inflated discrete distributions.

Details

Package: gamlss.countKinf
Type: Package
Version: 3.5.1
Date: 2018-11-2

The user can generates K-inflated distributions from discrete gamlss.family for fitting gamlss model.

Author(s)

Saeed Mohammadpour <<s.mohammadpour1111@gamlil.com>>, Mikis Stasinopoulos <<d.stasinopoulos@londonmet.ac.uk>>

Maintainer: Saeed Mohammadpour <<s.mohammadpour1111@gamlil.com>>

References


**Examples**

```r
# generating one inflated distribution from SICHEL model
gen.Kinf(family=SICHEL, kinf=1)
```

```r
# generating two inflated distribution from Delaporte model
gen.Kinf(family=DEL, kinf=1)
```

---

**Description**

The `gen.Kinf()` function allows the user to generate d, p, q, and r K-inflated distribution functions plus an extra K-inflated from `gamlss.family` function for fitting a K-inflated distribution with `gamlss`.

**Usage**

```r
gen.Kinf(family = "NO", kinf=1)
```

**Arguments**

- `family` a `gamlss.family` object, which is used to define the distribution for generating K-inflated model. The distribution families supported by `gamlss()` can be found in `gamlss.family`.
- `kinf` define inflated point in generating K-inflated distribution from discrete `gamlss.family`

**Value**

The functions `gen.Kinf` return d, p, q, and r K-inflated distribution functions and K-inflated distribution from discrete `gamlss.family`

**Author(s)**

Saeed Mohammadpour <<s.mohammadpour1111@gamlil.com>>, Mikis Stasinopoulos <<d.stasinopoulos@londonmet.ac.uk>>

**References**


Examples

# generate one inflated Negative Binomial distribution
gen.Kinflated(family ="NBI", kinf=1)

# generate one inflated Delaporte distribution
gen.Kinflated(family ="DEL", kinf=1)

# generate one inflated Sichel distribution
gen.Kinflated(family ="SICHEL", kinf=1)

KINFB

K-inflated Beta Negative Binomial distributions for fitting a GAMLSS model

Description

The function KINFB defines the K-inflated Beta Negative Binomial distribution, a four parameter distribution, for a gamlss.family object to be used in GAMLSS fitting using the function gamlss(). The functions dkKINFB, pkKINFB, qKINFB and rKINFB define the density, distribution function, quantile function and random generation for the K-inflated Beta Negative Binomial, KINFB(), distribution.

Usage

KINFB(mu.link = "log", sigma.link = "log", nu.link = "log", tau.link = "logit", kinf="K")

dKINFB(x, mu = 1, sigma = 1, nu = 1, tau = 0.1, kinf=0, log = FALSE)

pKINFB(q, mu = 1, sigma = 1, nu = 1, tau = 0.1, kinf=0, lower.tail = TRUE, log.p = FALSE)

qKINFB(p, mu = 1, sigma = 1, nu = 1, tau = 0.1, kinf=0, lower.tail = TRUE, log.p = FALSE, max.value = 10000)

rKINFB(n, mu = 1, sigma = 1, nu = 1, tau = 0.1, kinf=0, max.value = 10000)
Arguments

- **mu.link**: Defines the mu.link, with "log" link as the default for the mu parameter.
- **sigma.link**: Defines the sigma.link, with "log" link as the default for the sigma parameter.
- **nu.link**: Defines the nu.link, with "log" link as the default for the nu parameter.
- **tau.link**: Defines the tau.link, with "logit" link as the default for the tau parameter.
- **x**: vector of (non-negative integer) quantiles.
- **mu**: vector of positive means.
- **sigma**: vector of positive dispersion parameter.
- **nu**: vector of nu.
- **tau**: vector of inflated point probability.
- **p**: vector of probabilities.
- **q**: vector of quantiles.
- **n**: number of random values to return.
- **kinf**: defines inflated point in generating K-inflated distribution.
- **log,log.p**: logical; if TRUE, probabilities p are given as log(p).
- **lower.tail**: logical; if TRUE (default), probabilities are P[X <= x], otherwise, P[X > x].
- **max.value**: a constant, set to the default value of 10000 for how far the algorithm should look for q.

Details

The definition for the K-inflated Beta Negative Binomial distribution.

Value

The functions KIBNB return a gamlss.family object which can be used to fit K-inflated Beta Negative Binomial distribution in the gamlss() function.

Author(s)

Saeed Mohammadpour <<s.mohammadpour1111@gamlil.com>>, Mikis Stasinopoulos <<d.stasinopoulos@londonmet.ac.uk>>

References


**See Also**

`gamlss.family`, `KINBNB`

**Examples**

```
# ---------------------------------------------------------------------------
KINBNB() # gives information about the default links for the Beta Negative Binomial distribution
# ---------------------------------------------------------------------------

# generate zero inflated Beta Negative Binomial distribution
gen.Kinf(family=BNB, kinf=0)

# generate random sample from zero inflated Beta Negative Binomial distribution
x<-rinf0BNB(1000, mu=1, sigma=.5, nu=.2, tau=.2)

# fit the zero inflated Beta Negative Binomial distribution using gamlss
data<-data.frame(x=x)
  ## Not run:
gamlss(x~1, family=inf0BNB, data=data)
histDist(x, family=inf0BNB)
  ## End(Not run)
# ---------------------------------------------------------------------------

# generated one inflated Beta Negative Binomial distribution
gen.Kinf(family=BNB, kinf=1)

# generate random sample from one inflated Beta Negative Binomial distribution
x<-rinf1BNB(1000, mu=1, sigma=.5, nu=.2, tau=.2)

# fit the one inflated Beta Negative Binomial distribution using gamlss
data<-data.frame(x=x)
  ## Not run:
gamlss(x~1, family=inf1BNB, data=data)
histDist(x, family=inf1BNB)
  ## End(Not run)
# ---------------------------------------------------------------------------

mu=4; sigma=.5; nu=.2; tau=.2;
par(mgp=c(2,1,0),mar=c(4,4,4,1)+0.1)

# plot the pdf using plot
```

plot(function(x) dinfBNB(x, mu=mu, sigma=sigma, nu=nu, tau=tau), from=0, to=20, n=20+1, type="h", xlab="x", ylab="f(x)", cex.lab=1.5)
#---------------------------------------------------------------

# plot the cdf using plot
cdf <- stepfun(0:19, c(0,pinfBNB(0:19, mu=mu, sigma=sigma, nu=nu, tau=tau)), f=0)
plot(cdf, xlab="x", ylab="F(x)", verticals=FALSE, cex.points=.8, pch=16, main="", cex.lab=1.5)
#---------------------------------------------------------------

# plot the qdf using plot
invcdf <- stepfun(seq(0.01,.99,length=19), qinfBNB(seq(0.1,.99,length=20), mu, sigma), f=0)
plot(invcdf, ylab=expression(x[p]==F^{-1}(p)), do.points=FALSE, verticals=TRUE, cex.points=.8, pch=16, main="", cex.lab=1.5, xlab="p")
#---------------------------------------------------------------

# generate random sample
Ni <- rinfBNB(1000, mu=mu, sigma=sigma, nu=nu, tau=tau)
hist(Ni,breaks=seq(min(Ni)-0.5,max(Ni)+0.5,by=1),col="lightgray", main="",cex.lab=2)
barplot(table(Ni))
#---------------------------------------------------------------

KIDEL  
K-inflated Delaporte distributions for fitting a GAMLSS model

Description

The function KIDEL defines the K-inflated Delaporte distribution, a four parameter distribution, for a gamlss.family object to be used in GAMLSS fitting using the function gamlss(). The functions dKIDEL, pKIDEL, qKIDEL and rKIDEL define the density, distribution function, quantile function and random generation for the K-inflated Delaporte, KIDEL(), distribution.

Usage

KIDEL(mu.link = "log", sigma.link = "log", nu.link = "logit", tau.link = "logit", kinf="K")

dKIDEL(x, mu = 1, sigma = 1, nu = 0.5, tau = 0.1, kinf=0, log = FALSE)
pKIDEL(q, mu = 1, sigma = 1, nu = 0.5, tau = 0.1, kinf=0, lower.tail = TRUE, log.p = FALSE)
qKIDEL(p, mu = 1, sigma = 1, nu = 0.5, tau = 0.1, kinf=0, lower.tail = TRUE, log.p = FALSE, max.value = 10000)
rKIDEL(n, mu = 1, sigma = 1, nu = 0.5, tau = 0.1, kinf=0, max.value = 10000)
Arguments

- `mu.link` defines the `mu.link`, with "log" link as the default for the `mu` parameter
- `sigma.link` defines the `sigma.link`, with "log" link as the default for the `sigma` parameter
- `nu.link` defines the `nu.link`, with "logit" link as the default for the `nu` parameter
- `tau.link` defines the `tau.link`, with "logit" link as the default for the `tau` parameter

- `x` vector of (non-negative integer) quantiles
- `mu` vector of positive means
- `sigma` vector of positive dispersion parameter
- `nu` vector of `nu`
- `tau` vector of inflated point probability
- `p` vector of probabilities
- `q` vector of quantiles
- `n` number of random values to return
- `kinf` defines inflated point in generating K-inflated distribution
- `log.log.p` logical; if TRUE, probabilities `p` are given as `log(p)`
- `lower.tail` logical; if TRUE (default), probabilities are `P[X <= x]`, otherwise, `P[X > x]`
- `max.value` a constant, set to the default value of 10000 for how far the algorithm should look for `q`

Details

The definition for the K-inflated Delaporte distribution.

Value

The functions `KIDEL` return a `gamlss.family` object which can be used to fit K-inflated Delaporte distribution in the `gamlss()` function.

Author(s)

Saeed Mohammadpour <<s.mohammadpour1111@gamlil.com>>, Mikis Stasinopoulos <<d.stasinopoulos@londonmet.ac.uk>>

References


See Also
gamlss.family, KIDEL

Examples

# gives information about the default links for the Delaporte distribution
KIDEL()
# generate zero inflated Delaporte distribution
gen.Kinf(family=DEL, kinf=0)
# generate random sample from zero inflated Delaporte distribution
x<-rinf0DEL(1000, mu=1, sigma=.5, nu=.2, tau=.2)

# fit the zero inflated Delaporte distribution using gamlss
data<-data.frame(x=x)
## Not run:
gamlss(x~1, family=infl0DEL, data=data)
histDist(x, family=infl0DEL)
## End(Not run)

# generated one inflated Delaporte distribution
gen.Kinf(family=DEL, kinf=1)

# generate random sample from one inflated Delaporte distribution
x<-rinf1DEL(1000, mu=1, sigma=.5, nu=.2, tau=.2)

# fit the one inflated Delaporte distribution using gamlss
data<-data.frame(x=x)
## Not run:
gamlss(x~1, family=infl1DEL, data=data)
histDist(x, family=infl1DEL)
## End(Not run)

mu=4; sigma=.5; nu=.2; tau=.2;
par(mgp=c(2,1,0),mar=c(4,4,4,1)+0.1)

#plot the pdf using plot
plot(function(x) dinfl1DEL(x, mu=mu, sigma=sigma, nu=nu, tau=tau), from=0, to=20, n=20+1, type="h", xlab="x", ylab="f(x)", cex.lab=1.5)
#---\n
#plot the cdf using plot
cdf <- stepfun(c(0, pInfDEL(0:19, mu=mu, sigma=sigma, nu=nu, tau=tau)), f=0)
plot(cdf, xlab="x", ylab="F(x)", verticals=FALSE, cex.points=.8, pch=16, main="", cex.lab=1.5)

#plot the qdf using plot
qcdf <- stepfun(seq(0.01, .99, length=19), qInfDEL(seq(0.1, .99, length=20), mu, sigma), f=0)
plot(qcdf, ylab=expression(paste("x","="), f~"(-1)"(p)), do.points=FALSE, verticals=TRUE,
cex.points=.8, pch=16, main="", cex.lab=1.5, xlab="p")

# generate random sample
Ni <- rInfDEL(1000, mu=mu, sigma=sigma, nu=nu, tau=tau)
hist(Ni, breaks=seq(min(Ni)-0.5, max(Ni)+0.5, by=1), col="lightgray", main="", cex.lab=2)
barplot(table(Ni))

### KIDPO

**K-inflated Double Poisson distributions for fitting a GAMLSS model**

**Description**

The function KIDPO defines the K-inflated Double Poisson distribution, a three parameter distribution, for a gamlss.family object to be used in GAMLSS fitting using the function gamlss(). The functions dKIDPO, pKIDPO, qKIDPO and rKIDPO define the density, distribution function, quantile function and random generation for the K-inflated Double Poisson, KIDPO(), distribution.

**Usage**

KIDPO(mu.link = "log", sigma.link = "log", nu.link = "logit", kinf="K")
dKIDPO(x, mu = 1, sigma = 1, nu = 0.3, kinf=0 ,log = FALSE)
pKIDPO(q, mu = 1, sigma = 1, nu = 0.3, kinf=0, lower.tail = TRUE, log.p = FALSE)
qKIDPO(p, mu = 1, sigma = 1, nu = 0.3, kinf=0, lower.tail = TRUE, log.p = FALSE)
rKIDPO(n, mu = 1, sigma = 1, nu = 0.3, kinf=0)

**Arguments**

- mu.link: Defines the mu.link, with "log" link as the default for the mu parameter
- sigma.link: Defines the sigma.link, with "log" link as the default for the sigma parameter
nu.link | Defines the nu.link, with "logit" link as the default for the nu parameter
---|---
x | vector of (non-negative integer) quantiles
mu | vector of positive means
sigma | vector of positive dispersion parameter
nu | vector of inflated point probability
p | vector of probabilities
q | vector of quantiles
n | number of random values to return
kinf | defines inflated point in generating K-inflated distribution
log, log.p | logical; if TRUE, probabilities p are given as log(p)
lower.tail | logical; if TRUE (default), probabilities are P[X <= x], otherwise, P[X > x]

**Details**

The definition for the K-inflated Double Poisson distribution.

**Value**

The functions KIDPO return a gamlss.family object which can be used to fit K-inflated Double Poisson distribution in the gamlss() function.

**Author(s)**

Saeed Mohammadpour <<s.mohammadpour1111@gamlil.com>>, Mikis Stasinopoulos <<d.stasinopoulos@londonmet.ac.uk>>

**References**


**See Also**

*gamlss.family, KIDPO*
Examples

# gives information about the default links for the Double Poisson distribution
KIDPO()
#---------------------------------------------------------------

# generate zero inflated Double Poisson distribution
gen.Kinf(family=DPO, kinf=0)

# generate random sample from zero inflated Double Poisson distribution
x<-rinfDPO(1000, mu=1, sigma=.5, nu=.2)

# fit the zero inflated Double Poisson distribution using gamlss
data<-data.frame(x=x)
## Not run:
gamlss(x~1, family=infdpo, data=data)
histDist(x, family=infdpo)
## End(Not run)
#---------------------------------------------------------------

# generated one inflated Double Poisson distribution
gen.Kinf(family=DPO, kinf=1)

# generate random sample from one inflated Double Poisson distribution
x<-rinf1DPO(1000, mu=1, sigma=.5, nu=.2)

# fit the one inflated Double Poisson distribution using gamlss
data<-data.frame(x=x)
## Not run:
gamlss(x~1, family=inf1dpo, data=data)
histDist(x, family=inf1DPO)
## End(Not run)
#---------------------------------------------------------------

mu=4; sigma=.5; nu=.2;
par(mgp=c(2,1,0),mar=c(4,4,4,1)+.1)

# plot the pdf using plot
plot(function(x) dinf1DPO(x, mu=mu, sigma=sigma, nu=nu), from=0, to=20, n=20+1, type="h",xlab="x",ylab="f(x)",cex.lab=1.5)
#---------------------------------------------------------------

# plot the cdf using plot
cdf <- stepfun(0:19, c(0,inf1DPO(0:19, mu=mu, sigma=sigma, nu=nu)), f=0)
plot(cdf, xlab="x", ylab="F(x)", verticals=FALSE, cex.points=.8, pch=16, main="",cex.lab=1.5)
#---------------------------------------------------------------

# plot the qdf using plot
invcdf <- stepfun(seq(0.01,.99,length=19), qinf1DPO(seq(0.1,.99,length=20),mu, sigma), f=0)
plot(invcdf, ylab=expression(x[p]==F^(-1)(p)), do.points=FALSE,verticals=TRUE, cex.points=.8, pch=16, main="",cex.lab=1.5, xlab="p")
# generate random sample
Ni <- rinf1DP0(1000, mu=mu, sigma=sigma, nu=nu)
hist(Ni,breaks=seq(min(Ni)-0.5,max(Ni)+0.5,by=1),col="lightgray", main="",cex.lab=2)
barplot(table(Ni))

---

**KIGEOM**

*K*-inflated Geometric distributions for fitting a GAMLSS model

**Description**

The function KIGEOM defines the *K*-inflated Geometric distribution, a two parameter distribution, for a `gamlss.family` object to be used in GAMLSS fitting using the function `gamlss()`. The functions `dkIGeom`, `pkIGeom`, `qkIGeom` and `rkIGeom` define the density, distribution function, quantile function and random generation for the *K*-inflated Geometric, KIGEOM, distribution.

**Usage**

```r
KIGEOM(mu.link = "log", sigma.link = "logit", kinf="K")
dKIGEOM(x, mu = 1, sigma = 0.1, kinf=0, log = FALSE)
pKIGEOM(q, mu = 1, sigma = 0.1, kinf=0, lower.tail = TRUE, log.p = FALSE)
qKIGEOM(p, mu = 1, sigma = 0.1, kinf=0, lower.tail = TRUE, log.p = FALSE)
rKIGEOM(n, mu = 1, sigma = 0.1, kinf=0)
```

**Arguments**

- `mu.link` Defines the `mu.link`, with "log" link as the default for the mu parameter
- `sigma.link` Defines the `sigma.link`, with "logit" link as the default for the sigma parameter
- `x` vector of (non-negative integer) quantiles
- `mu` vector of positive means
- `sigma` vector of inflated point probability
- `p` vector of probabilities
- `q` vector of quantiles
- `n` number of random values to return
- `kinf` defines inflated point in generating *K*-inflated distribution
- `log,log.p` logical; if `TRUE`, probabilities `p` are given as `log(p)`
- `lower.tail` logical; if `TRUE` (default), probabilities are `P[X <= x]`, otherwise, `P[X > x]`
Details

The definition for the K-inflated Geometric distribution.

Value

The functions \texttt{KIGEOM} return a \texttt{gamlss} family object which can be used to fit K-inflated Geometric distribution in the \texttt{gamlss()} function.

Author(s)

Saeed Mohammadpour \texttt{<s.mohammadpour1111@gamlil.com>}, Mikis Stasinopoulos \texttt{<d.stasinopoulos@londonmet.ac.uk>}

References


Rigby, R. A. and Stasinopoulos D. M. (2010) The gamlss.family distributions, (distributed with this package or see \url{http://www.gamlss.org/})


See Also

\texttt{gamlss.family}, \texttt{KIGEOM}

Examples

```r
# gives information about the default links for the Geometric distribution
KIGEOM()

# generate zero inflated Geometric distribution
gen.Kinf(family=GEOM, kinf=0)

# generate random sample from zero inflated Geometric distribution
x<-rinf@GEOM(1000,mu=1, sigma=.2)
```
# fit the zero inflated Geometric distribution using gamlss
data<-data.frame(x=x)
## Not run:
gamlss(x~1, family=inf0GEOM, data=data)
histDist(x, family=inf0GEOM)
## End(Not run)

# generated one inflated Geometric distribution
gen.Kinfl(family=GEOM, kinfl=1)

# generate random sample from one inflated Geometric distribution
x<-rinf1GEOM(1000, mu=1, sigma=.2)

# fit the one inflated Geometric distribution using gamlss
data<-data.frame(x=x)
## Not run:
gamlss(x~1, family=inf1GEOM, data=data)
histDist(x, family=inf1GEOM)
## End(Not run)

mu=1; sigma=.2;
par(mgp=c(2.1,0),mar=c(4,4,4,1)+0.1)

#plot the pdf using plot
plot(function(x) dinf1GEOM(x, mu=mu, sigma=sigma), from=0, to=20, n=20+1,
type="h",xlab="x",ylab="f(x) ",cex.lab=1.5)

#plot the cdf using plot
cdf <- stepfun(0:19, c(0,pinf1GEOM(0:19, mu=mu, sigma=sigma)), f = 0)
plot(cdf, xlab="x", ylab="F(x)", verticals=FALSE,cex.points=.8, pch=16, main="",cex.lab=1.5)

#plot the qdf using plot
invcdf <- stepfun(seq(0.01,.99,length=19),qinf1GEOM(seq(0.1,.99,length=20),mu, sigma), f = 0)
plot(invcdf, ylab=expression(x[p]=F^{-1}(p)), do.points=FALSE,verticals=TRUE,
cex.points=.8, pch=16, main="",cex.lab=1.5, xlab="p")

# generate random sample
Ni <- rinf1GEOM(1000, mu=mu, sigma=sigma)
hist(Ni,breaks=seq(min(Ni)-0.5,max(Ni)+0.5,by=1),col="lightgray", main="",cex.lab=2)
barplot(table(Ni))

---

**KIGEO**

-K-inflated Geometric original distributions for fitting a GAMLSS model-
**Description**

The function `KIGEOMo` defines the K-inflated Geometric original distribution, a two parameter distribution, for a `gamlss.family` object to be used in GAMLSS fitting using the function `gamlss()`. The functions `dkIGEOMo`, `pkIGEOMo`, `qKIGEOMo` and `rKIGEOMo` define the density, distribution function, quantile function and random generation for the K-inflated Geometric original, `KIGEOMo()`, distribution.

**Usage**

```r
KIGEOMo(mu.link = "logit", sigma.link = "logit", kinf="K")
dKIGEOMo(x, mu = .1, sigma = 0.1, kinf=0, log = FALSE)
pKIGEOMo(q, mu = .1, sigma = 0.1, kinf=0, lower.tail = TRUE, log.p = FALSE)
qKIGEOMo(p, mu = 1, sigma = 0.1, kinf=0, lower.tail = TRUE, log.p = FALSE)
rKIGEOMo(n, mu = 1, sigma = 0.1, kinf=0)
```

**Arguments**

- `mu.link`: Defines the `mu.link`, with "logit" link as the default for the `mu` parameter
- `sigma.link`: Defines the `sigma.link`, with "logit" link as the default for the `sigma` parameter
- `x`: vector of (non-negative integer) quantiles
- `mu`: vector of positive means
- `sigma`: vector of inflated point probability
- `p`: vector of probabilities
- `q`: vector of quantiles
- `n`: number of random values to return
- `kinf`: defines inflated point in generating K-inflated distribution
- `log`, `log.p`: logical; if TRUE, probabilities p are given as log(p)
- `lower.tail`: logical; if TRUE (default), probabilities are P[X <= x], otherwise, P[X > x]

**Details**

The definition for the K-inflated Geometric original distribution.

**Value**

The functions `KIGEOMo` return a `gamlss.family` object which can be used to fit K-inflated Geometric original distribution in the `gamlss()` function.

**Author(s)**

Saeed Mohammadpour <<s.mohammadpour111@gamlil.com>>, Mikis Stasinopoulos <<d.stasinopoulos@londonmet.ac.uk>>
References


See Also

`gamlss.family`, `KIGEOMo`

Examples

```r
# gives information about the default links for the Geometric original distribution
KIGEOMo()

# generate zero inflated Geometric original distribution
gen.Kinf(family=GEOMo, kinf=0)

# generate random sample from zero inflated Geometric original distribution
x<-rinf@GEOMo(1000, mu=.5, sigma=.2)

# fit the zero inflated Geometric original distribution using gamlss
data<-data.frame(x=x)
# Not run:
gamlss(x~1, family=inf@GEOMo, data=data)
histDist(x, family=inf@GEOMo)
# End(Not run)

data

# generated one inflated Geometric original distribution
gen.Kinf(family=GEOMo, kinf=1)

# generate random sample from one inflated Geometric original distribution
x<-rinf1GEOMo(1000, mu=.5, sigma=.2)
```
# fit the one inflated Geometric original distribution using gamlss

data<-.data.frame(x=x)
## Not run:
gamlss(x~1, family=infl0GEoMo, data=data)
histDist(x, family=infl0GEoMo)
## End(Not run)

mu=.3; sigma=.2;
par(mgp=c(2,1,0),mar=c(4,4,4,1)+0.1)
#plot the pdf using plot
plot(function(x) dinf0GEoMo(x, mu=mu, sigma=sigma), from=0, to=20, n=20+1,
     type="h",xlab="x",ylab="f(x)",cex.lab=1.5)

#plot the cdf using plot
cdf <- stepfun(0:19, c(0,pinf0GEoMo(0:19, mu=mu, sigma=sigma)), f=0)
plot(cdf, xlab="x", ylab="F(x)", verticals=FALSE, cex.points=.8, pch=16, main="",cex.lab=1.5)

#plot the qdf using plot
invcdf <- stepfun(seq(-0.1,.99,length=19), qinf0GEoMo(seq(-0.1,.99,length=20),mu, sigma), f=0)
plot(invcdf, ylab=expression(x[p]=-F^-1(p)), do.points=FALSE,verticals=TRUE,
     cex.points=.8, pch=16, main="",cex.lab=1.5, xlab="p")

# generate random sample
Ni <- rinf0GEoMo(1000, mu=mu, sigma=sigma)
hist(Ni,breaks=seq(min(Ni)-0.5,max(Ni)+0.5,by=1),col="lightgray", main="",cex.lab=2)
barplot(table(Ni))

---

**KIGPO**  
*K-inflated Generalised Poisson distributions for fitting a GAMLSS model*

**Description**

The function KIGPO defines the K-inflated Generalised Poisson distribution, a three parameter distribution, for a gamlss.family object to be used in GAMLSS fitting using the function gamlss(). The functions dKIGPO, pKIGPO, qKIGPO and rKIGPO define the density, distribution function, quantile function and random generation for the K-inflated Generalised Poisson, KIGPO().

**Usage**

```
KIGPO(mu.link = "log", sigma.link = "log", nu.link = "logit", kinf="K")
dKIGPO(x, mu = 1, sigma = 1, nu = 0.3, kinf=0 ,log = FALSE)
```
pKIGPO(q, mu = 1, sigma = 1, nu = 0.3, kinf=0, lower.tail = TRUE, log.p = FALSE)

qKIGPO(p, mu = 1, sigma = 1, nu = 0.3, kinf=0, lower.tail = TRUE, log.p = FALSE)

rKIGPO(n, mu = 1, sigma = 1, nu = 0.3, kinf=0)

Arguments

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>mu</td>
<td>Defines the mu.link, with &quot;log&quot; link as the default for the mu parameter</td>
</tr>
<tr>
<td>sigma</td>
<td>Defines the sigma.link, with &quot;log&quot; link as the default for the sigma parameter</td>
</tr>
<tr>
<td>nu</td>
<td>Defines the nu.link, with &quot;logit&quot; link as the default for the nu parameter</td>
</tr>
<tr>
<td>x</td>
<td>vector of (non-negative integer) quantiles</td>
</tr>
<tr>
<td>mu</td>
<td>vector of positive means</td>
</tr>
<tr>
<td>sigma</td>
<td>vector of positive dispersion parameter</td>
</tr>
<tr>
<td>nu</td>
<td>vector of inflated point probability</td>
</tr>
<tr>
<td>p</td>
<td>vector of probabilities</td>
</tr>
<tr>
<td>q</td>
<td>vector of quantiles</td>
</tr>
<tr>
<td>n</td>
<td>number of random values to return</td>
</tr>
<tr>
<td>kinf</td>
<td>defines inflated point in generating K-inflated distribution</td>
</tr>
<tr>
<td>log.p</td>
<td>logical; if TRUE, probabilities p are given as log(p)</td>
</tr>
<tr>
<td>lower.tail</td>
<td>logical; if TRUE (default), probabilities are P[X &lt;= x], otherwise, P[X &gt; x]</td>
</tr>
</tbody>
</table>

Details

The definition for the K-inflated Generalised Poisson distribution.

Value

The functions KIGPO return a gamlss.family object which can be used to fit K-inflated Generalised Poisson distribution in the gamlss() function.

Author(s)

Saeed Mohammadpour <<s.mohammadpour1111@gamlil.com>>, Mikis Stasinopoulos <<d.stasinopoulos@londonmet.ac.uk>>

References


See Also

`gamlss.family`, `KIGPO`

Examples

```r
# gives information about the default links for the Generalised Poisson distribution
KIGPO()

# generate zero inflated Generalised Poisson distribution
gen.Kinf(family=GPO, kinf=0)

# generate random sample from zero inflated Generalised Poisson distribution
x<-rinf0GPO(1000, mu=1, sigma=.5, nu=.2)

# fit the zero inflated Generalised Poisson distribution using gamlss
data<-data.frame(x=x)
  gamlss(x~1, family=inflGPO, data=data)
  histDist(x, family=inflGPO)

# generated one inflated Generalised Poisson distribution
gen.Kinf(family=GPO, kinf=1)

# generate random sample from one inflated Generalised Poisson distribution
x<-rinflGPO(1000, mu=1, sigma=.5, nu=.2)

# fit the one inflated Generalised Poisson distribution using gamlss
data<-data.frame(x=x)
  gamlss(x~1, family=inflGPO, data=data)
  histDist(x, family=inflGPO)
```

The function KILG defines the K-inflated Logarithmic distribution, a two parameter distribution, for a `gamlss.family` object to be used in GAMLSS fitting using the function `gamlss()`. The functions `dKILG`, `pKILG`, `qKILG` and `rKILG` define the density, distribution function, quantile function and random generation for the K-inflated Logarithmic, KILG(), distribution.

Usage

```r
KILG(mu.link = "logit", sigma.link = "logit", kinf="K")
dKILG(x, mu = .1, sigma = 0.1, kinf=0, log = FALSE)
pKILG(q, mu = .1, sigma = 0.1, kinf=0, lower.tail = TRUE, log.p = FALSE)
qKILG(p, mu = 1, sigma = 0.1, kinf=0, lower.tail = TRUE, log.p = FALSE)
rKILG(n, mu = 1, sigma = 0.1, kinf=0)
```
Arguments

mu.link
Defines the mu.link, with "logit" link as the default for the mu parameter

sigma.link
Defines the sigma.link, with "logit" link as the default for the sigma parameter

x
vector of (non-negative integer) quantiles

mu
vector of positive means

sigma
vector of inflated point probability

p
vector of probabilities

q
vector of quantiles

n
number of random values to return

kinf
defines inflated point in generating K-inflated distribution

log, log.p
logical; if TRUE, probabilities p are given as log(p)

lower.tail
logical; if TRUE (default), probabilities are P[X <= x], otherwise, P[X > x]

Details

The definition for the K-inflated Logarithmic distribution.

Value

The functions KILG return a gamlss.family object which can be used to fit K-inflated Logarithmic distribution in the gamlss() function.

Author(s)

Saeed Mohammadpour <<s.mohammadpour1111@gamlil.com>>, Mikis Stasinopoulos <<d.stasinopoulos@londonmet.ac.uk>>

References


See Also

\texttt{gamlss.family.KILG}

Examples

# gives information about the default links for the Logarithmic distribution
KILG()

# generate zero inflated Logarithmic distribution
\texttt{gen.Kinf(family=LG, kinf=0)}

# generate random sample from zero inflated Logarithmic distribution
\texttt{x<-rinf0LG(1000, mu=1, sigma=.2)}

# fit the zero inflated Logarithmic distribution using \texttt{gamlss}
data<-data.frame(x=x)
# Not run:
gamlss(x~1, family=inf0LG, data=data)
\texttt{histDist(x, family=inf0LG)}
# End(Not run)

# generated one inflated Logarithmic distribution
gen.Kinf(family=LG, kinf=1)

# generate random sample from one inflated Logarithmic distribution
\texttt{x<-rinf1LG(1000, mu=1, sigma=.2)}

# fit the one inflated Logarithmic distribution using \texttt{gamlss}
data<-data.frame(x=x)
# Not run:
gamlss(x~1, family=inf1LG, data=data)
\texttt{histDist(x, family=inf1LG)}
# End(Not run)

\texttt{mu=.5; sigma=.2;}
par(mgp=c(2,1,0),mar=c(4,4,4,1)+0.1)

#plot the pdf using plot
\texttt{plot(function(x) dinf1LG(x, mu=mu, sigma=sigma), from=1, to=20, n=20+1, type="h",xlab="x",ylab="f(x)",cex.lab=1.5)}
#--

#plot the cdf using plot
\texttt{cdf <- stepfun(1:19, c(0,pinf1LG(1:19, mu=mu, sigma=sigma)), f = 0)}
\texttt{plot(cdf, xlab="x", ylab="F(x)", verticals=FALSE, cex.points=.8, pch=16, main="",cex.lab=1.5)}
#--
KINBF

K-inflated Negative Binomial Family distributions for fitting a GAMLSS model

Description
The function KINBF defines the K-inflated Negative Binomial Family distribution, a four parameter distribution, for a gamlss.family object to be used in GAMLSS fitting using the function gamlss(). The functions dKINBF, pKINBF, qKINBF and rKINBF define the density, distribution function, quantile function and random generation for the K-inflated Negative Binomial Family, KINBF(). distribution.

Usage

KINBF(mu.link = "log", sigma.link = "log", nu.link = "log", 
  tau.link = "logit", kinf="K")

dKINBF(x, mu = 1, sigma = 1, nu = 2, tau = 0.1, kinf=0, log = FALSE)

pKINBF(q, mu = 1, sigma = 1, nu = 2, tau = 0.1, kinf=0, lower.tail = TRUE, 
  log.p = FALSE)

qKINBF(p, mu = 1, sigma = 1, nu = 2, tau = 0.1, kinf=0, lower.tail = TRUE, 
  log.p = FALSE)

rKINBF(n, mu = 1, sigma = 1, nu = 2, kinf=0, tau = 0.1)

Arguments

mu.link       Defines the mu.link, with "log" link as the default for the mu parameter
sigma.link    Defines the sigma.link, with "log" link as the default for the sigma parameter
nu.link       Defines the nu.link, with "log" link as the default for the nu parameter
tau.link      Defines the tau.link, with "logit" link as the default for the tau parameter
x             vector of (non-negative integer) quantiles
mu      vector of positive means
sigma   vector of positive dispersion parameter
nu      vector of nu
tau     vector of inflated point probability
p       vector of probabilities
q       vector of quantiles
n       number of random values to return
kinf    defines inflated point in generating K-inflated distribution
log.log.p logical; if TRUE, probabilities p are given as log(p)
lower.tail logical; if TRUE (default), probabilities are P[X <= x], otherwise, P[X > x]

Details
The definition for the K-inflated Negative Binomial Family distribution.

Value
The functions KINBF return a gamlss.family object which can be used to fit K-inflated Negative Binomial Family distribution in the gamlss() function.

Author(s)
Saeed Mohammadpour <<s.mohammadpour1111@gamlil.com>>, Mikis Stasinopoulos <<d.stasinopoulos@londonmet.ac.uk>>

References

See Also
gamlss.family, KINBF
Examples

# gives information about the default links for the Negative Binomial Family distribution
KinBF()

# generate zero inflated Negative Binomial Family distribution
gen.Kinf(family=NB, kinf=0)

# generate random sample from zero inflated Negative Binomial Family distribution
x<-rinf0NB(1000, mu=1, sigma=.5, nu=-.2, tau=.2)

# fit the zero inflated Negative Binomial Family distribution using gamlss
data<-data.frame(x=x)
## Not run:
gamlss(x~1, family=inf0NB, data=data)
histDist(x, family=inf0NB)
## End(Not run)

# generated one inflated Negative Binomial Family distribution
gen.Kinf(family=NB, kinf=1)

# generate random sample from one inflated Negative Binomial Family distribution
x<-rinf1NB(1000, mu=1, sigma=.5, nu=-.2, tau=.2)

# fit the one inflated Negative Binomial Family distribution using gamlss
data<-data.frame(x=x)
## Not run:
gamlss(x~1, family=inf1NB, data=data)
histDist(x, family=inf1NB)
## End(Not run)

mu=4; sigma=.5; nu=.2; tau=.2;
par(mgp=c(2,1,0),mar=c(4,4,4,1)+0.1)

# plot the pdf using plot
plot(function(x) dinf1NB(x, mu=mu, sigma=sigma, nu=nu, tau=tau), from=0, to=20, n=20+1, type="h", xlab="x", ylab="f(x)", cex.lab=1.5)

# plot the cdf using plot
cdf <- stepfun(0:19, c(0,pinf1NB(0:19, mu=mu, sigma=sigma, nu=nu, tau=tau)), f = 0)
plot(cdf, xlab="x", ylab="F(x)", verticals=FALSE, cex.points=.8, pch=16, main=",cex.lab=1.5)

# plot the qdf using plot
invcdf <- stepfun(seq(0.01, .99,length=19), qinf1NB(seq(0.1,.99,length=20),mu, sigma), f = 0)
plot(invcdf, ylab=expression(x[p]=F^(-1)(p)), do.points=FALSE,verticals=TRUE, cex.points=.8, pch=16, main="",cex.lab=1.5, xlab="p")
KINBI

---

# generate random sample
N1 <- rinfNBF(1000, mu=mu, sigma=sigma, nu=nu, tau=tau)
hist(N1,breaks=seq(min(N1)-0.5,max(N1)+0.5,by=1),col="lightgray", main="",cex.lab=2)
barplot(table(N1))

---

KINBI  

K-inflated Negative Binomial distributions for fitting a GAMLSS model

Description

The function KINBI defines the K-inflated Negative Binomial distribution, a three parameter distribution, for a gamlss.family object to be used in GAMLSS fitting using the function gamlss(). The functions dKINBI, pKINBI, qKINBI and rKINBI define the density, distribution function, quantile function and random generation for the K-inflated Negative Binomial.KINBI() distribution.

Usage

KINBI(mu.link = "log", sigma.link = "log", nu.link = "logit", kinf="K")
dKINBI(x, mu = 1, sigma = 1, nu = 0.3, kinf=0 ,log = FALSE)
pKINBI(q, mu = 1, sigma = 1, nu = 0.3, kinf=0, lower.tail = TRUE, log.p = FALSE)
qKINBI(p, mu = 1, sigma = 1, nu = 0.3, kinf=0, lower.tail = TRUE, log.p = FALSE)
rKINBI(n, mu = 1, sigma = 1, nu = 0.3, kinf=0)

Arguments

mu.link  
Defines the mu.link, with "log" link as the default for the mu parameter
sigma.link  
Defines the sigma.link, with "log" link as the default for the sigma parameter
nu.link  
Defines the nu.link, with "logit" link as the default for the nu parameter
x  
vector of (non-negative integer) quantiles
mu  
vector of positive means
sigma  
vector of positive dispersion parameter
nu  
vector of inflated point probability
p  
vector of probabilities
q  
vector of quantiles
n  
number of random values to return
kInf defines inflated point in generating K-inflated distribution

log, log.p logical; if TRUE, probabilities p are given as log(p)

lower.tail logical; if TRUE (default), probabilities are P[X <= x], otherwise, P[X > x]

Details

The definition for the K-inflated Negative Binomial distribution.

Value

The functions KINBI return a gamlss.family object which can be used to fit K-inflated Negative Binomial distribution in the gamlss() function.

Author(s)

Saeed Mohammadpour <s.mohammadpour1111@gamlil.com>, Mikis Stasinopoulos <d.stasinopoulos@londonmet.ac.uk>

References


See Also

gamlss.family,KINBI

Examples

```
# gives information about the default links for the Negative Binomial distribution
KINBI()
```

# generate zero inflated Negative Binomial distribution
gen.Kinf(family=NBI, kinf=0)

# generate random sample from zero inflated Negative Binomial distribution
x<-rinf0NBI(1000, mu=1, sigma=.5, nu=.2)

# fit the zero inflated Negative Binomial distribution using gamlss
data<-data.frame(x=x)
## Not run:
gamlss(x~1, family=inflNBI, data=data)
histDist(x, family=inflNBI)
## End(Not run)
#-------------------------------------------------------------------------------------

# generated one inflated Negative Binomial distribution
gen.Kinf(family=NBI, kinf=1)

# generate random sample from one inflated Negative Binomial distribution
x<-rinf1NBI(1000, mu=1, sigma=.5, nu=.2)

# fit the one inflated Negative Binomial distribution using gamlss
data<-data.frame(x=x)
## Not run:
gamlss(x~1, family=inflNBI, data=data)
histDist(x, family=inflNBI)
## End(Not run)
#-------------------------------------------------------------------------------------

mu=4; sigma=.5; nu=.2;
par(mgp=c(2,1,0), mar=c(4,4,4,1)+0.1)

#plot the pdf using plot
plot(function(x) dinf1NBI(x, mu=mu, sigma=sigma, nu=nu), from=0, to=20, n=20+1, type="h", xlab="x", ylab="f(x)", cex.lab=1.5)
#-------------------------------------------------------------------------------------

#plot the cdf using plot
cdf<-stepfun(0:19, c(0,pinf1NBI(0:19, mu=mu, sigma=sigma, nu=nu)), f = 0)
plot(cdf, xlab="x", ylab="F(x)", verticals=FALSE, cex.points=.8, pch=16, main="", cex.lab=1.5)
#-------------------------------------------------------------------------------------

#plot the qdf using plot
invcdf <- stepfun(seq(0.01,.99,length=19), qinf1NBI(seq(0.1,.99,length=20),mu,sigma), f = 0)
plot(invcdf, ylab=expression(x[p]=F^(-1)(p)), do.points=FALSE, verticals=TRUE, cex.points=.8, pch=16, main="", cex.lab=1.5, xlab="p")
#-------------------------------------------------------------------------------------

# generate random sample
Ni <- rinf1NBI(1000, mu=mu, sigma=sigma, nu=nu)
hist(Ni,breaks=seq(min(Ni)/.5,max(Ni)+.5,by=1), col="lightgray", main="", cex.lab=2)
barplot(table(Ni))
#-------------------------------------------------------------------------------------
KINBII  

K-inflated Negative Binomial type II distributions for fitting a GAMLSS model

Description

The function KINBII defines the K-inflated Negative Binomial type II distribution, a three parameter distribution, for a gamlss.family object to be used in GAMLSS fitting using the function gamlss(). The functions dKINBII, pKINBII, qKINBII and rKINBII define the density, distribution function, quantile function and random generation for the K-inflated Negative Binomial type II, KINBII(), distribution.

Usage

KINBII(mu.link = "log", sigma.link = "log", nu.link = "logit", kinf="K")
dKINBII(x, mu = 1, sigma = 1, nu = 0.3, kinf=0, log = FALSE)
pKINBII(q, mu = 1, sigma = 1, nu = 0.3, kinf=0, lower.tail = TRUE, log.p = FALSE)
qKINBII(p, mu = 1, sigma = 1, nu = 0.3, kinf=0, lower.tail = TRUE, log.p = FALSE)
rKINBII(n, mu = 1, sigma = 1, nu = 0.3, kinf=0)

Arguments

mu.link Defines the mu.link, with "log" link as the default for the mu parameter
sigma.link Defines the sigma.link, with "log" link as the default for the sigma parameter
nu.link Defines the nu.link, with "logit" link as the default for the nu parameter
x vector of (non-negative integer) quantiles
mu vector of positive means
sigma vector of positive dispersion parameter
nu vector of inflated point probability
p vector of probabilities
q vector of quantiles
n number of random values to return
kinf defines inflated point in generating K-inflated distribution
log, log.p logical; if TRUE, probabilities p are given as log(p)
lower.tail logical; if TRUE (default), probabilities are P[X <= x], otherwise, P[X > x]
Details

The definition for the K-inflated Negative Binomial type II distribution.

Value

The functions KINBII return a gamlss.family object which can be used to fit K-inflated Negative Binomial type II distribution in the gamlss() function.

Author(s)

Saeed Mohammadpour <<s.mohammadpour1111@gamlil.com>>, Mikis Stasinopoulos <<d.stasinopoulos@londonmet.ac.uk>>

References


See Also

gamlss.family,KINBII

Examples

#---------------------------------------------------------------
# gives default links for the Negative Binomial distribution type II
# KINBII()
#---------------------------------------------------------------

# generate zero inflated Negative Binomial type II distribution
gen.Kinf(family=NBII, kinf=0)

# generate random sample from zero inflated Negative Binomial type II distribution
x<-rinf0NBII(1000, mu=1, sigma=.5, nu=.2)
# fit the zero inflated Negative Binomial type II distribution using gamlss
# Not run:
gamlss(x~1, family=inf0NBII, data=data)
histDist(x, family=inf0NBII)
# End(Not run)

# generated one inflated Negative Binomial type II distribution
gen.Kinf(family=NBII, kinf=1)

# generate random sample from one inflated Negative Binomial type II distribution
x<-rinf1NBII(1000, mu=1, sigma=.5, nu=.2)

# fit the one inflated Negative Binomial type II distribution using gamlss
# Not run:
gamlss(x~1, family=inf1NBII, data=data)
histDist(x, family=inf1NBII)
# End(Not run)

mu=4; sigma=.5; nu=.2; tau=.2;
par(mgp=c(2,1,0),mar=c(4,4,4,1)+0.1)

#plot the pdf using plot
plot(function(x) dinf1NBII(x, mu=mu, sigma=sigma, nu=nu), from=0, to=20, n=20+1, type="h",xlab="x",ylab="f(x)",cex.lab=1.5)

#plot the cdf using plot
cdf <- stepfun(0:19, c(0,pinf1NBII(0:19, mu=mu, sigma=sigma, nu=nu)), f = 0)
plot(cdf, xlab="x", ylab="F(x)", verticals=FALSE, cex.points=.8, pch=16, main="",cex.lab=1.5)

#plot the qdf using plot
invcdf <- stepfun(seq(0.01,.99,length=19), qinf1NBII(seq(0.01,.99,length=20),mu, sigma), f = 0)
plot(invcdf, ylab=expression(x[p]=F^(-1)(p)), do.points=FALSE,verticals=TRUE, cex.points=.8, pch=16, main="",cex.lab=1.5, xlab="p")

# generate random sample
Ni <- rinf1NBII(1000, mu=mu, sigma=omega, nu=nu)
hist(Ni,breaks=seq(min(Ni)-0.5,max(Ni)+0.5,by=1),col="lightgray", main="",cex.lab=2)
barplot(table(Ni))

---

KIPIG

K-inflated Poisson Inverse Gaussian distributions for fitting a GAMMSS model
Description

The function KIPIG defines the K-inflated Poisson Inverse Gaussian distribution, a three parameter distribution, for a gamlss.family object to be used in GAMLSS fitting using the function gamlss(). The functions dKIPIG, pKIPIG, qKIPIG and rKIPIG define the density, distribution function, quantile function and random generation for the K-inflated Poisson Inverse Gaussian, KIPIG(), distribution.

Usage

```r
KIPIG(mu.link = "log", sigma.link = "log", nu.link = "logit", kinf="K")
dKIPIG(x, mu = 1, sigma = 1, nu = 0.3, kinf=0, log = FALSE)
pKIPIG(q, mu = 1, sigma = 1, nu = 0.3, kinf=0, lower.tail = TRUE,
      log.p = FALSE)
qKIPIG(p, mu = 1, sigma = 1, nu = 0.3, kinf=0, lower.tail = TRUE,
      log.p = FALSE, max.value = 10000)
rKIPIG(n, mu = 1, sigma = 1, nu = 0.3, kinf=0, max.value = 10000)
```

Arguments

- **mu.link**: Defines the mu.link, with "log" link as the default for the mu parameter
- **sigma.link**: Defines the sigma.link, with "log" link as the default for the sigma parameter
- **nu.link**: Defines the nu.link, with "logit" link as the default for the nu parameter
- **x**: vector of (non-negative integer) quantiles
- **mu**: vector of positive means
- **sigma**: vector of positive dispersion parameter
- **nu**: vector of inflated point probability
- **p**: vector of probabilities
- **q**: vector of quantiles
- **n**: number of random values to return
- **kinf**: defines inflated point in generating K-inflated distribution
- **log.p**: logical; if TRUE, probabilities p are given as log(p)
- **lower.tail**: logical; if TRUE (default), probabilities are P[X <= x], otherwise, P[X > x]
- **max.value**: a constant, set to the default value of 10000 for how far the algorithm should look for q

Details

The definition for the K-inflated Poisson Inverse Gaussian distribution.
Value

The functions KIPIG return a gamlss.family object which can be used to fit K-inflated Poisson Inverse Gaussian distribution in the `gamlss()` function.

Author(s)

Saeed Mohammadpour <<s.mohammadpour1111@gamlil.com>>, Mikis Stasinopoulos <<d.stasinopoulos@londonmet.ac.uk>>

References


See Also

gamlss.family, KIPIG

Examples

#---------------------------------------------------------------

# gives information about the default links for the Poisson Inverse Gaussian distribution
KIPIG()
#---------------------------------------------------------------

# generate zero inflated Poisson Inverse Gaussian distribution
gen.Kinf(family=PIG, kinf=0)

# generate random sample from zero inflated Poisson Inverse Gaussian distribution
x<-rinf0PIG(1000, mu=1, sigma=.5, nu=.2)

# fit the zero inflated Poisson Inverse Gaussian distribution using gamlss
data<-data.frame(x=x)
## Not run:
gamlss(x~1, family=inf0PIG, data=data)
histDist(x, family=inf0PIG)
KIPO

K-inflated Poisson distributions for fitting a GAMLSS model

Description

The function KIPO defines the K-inflated Poisson distribution, a two parameter distribution, for a gamlss.family object to be used in GAMLSS fitting using the function gamlss(). The functions dKIPO, pKIPO, qKIPO and rKIPO define the density, distribution function, quantile function and random generation for the K-inflated Poisson, KIPO(), distribution.
Usage

KIPO(mu.link = "log", sigma.link = "logit", kinf="K")

dKIPO(x, mu = 1, sigma = 0.1, kinf=0, log = FALSE)

pKIPO(q, mu = 1, sigma = 0.1, kinf=0, lower.tail = TRUE, log.p = FALSE)

qKIPO(p, mu = 1, sigma = 0.1, kinf=0, lower.tail = TRUE, log.p = FALSE)

rKIPO(n, mu = 1, sigma = 0.1, kinf=0)

Arguments

mu.link Defines the mu.link, with "log" link as the default for the mu parameter
sigma.link Defines the sigma.link, with "logit" link as the default for the sigma parameter
x vector of (non-negative integer) quantiles
mu vector of positive means
sigma vector of inflated point probability
p vector of probabilities
q vector of quantiles
n number of random values to return
kinf defines inflated point in generating K-inflated distribution
log, log.p logical; if TRUE, probabilities p are given as log(p)
lower.tail logical; if TRUE (default), probabilities are P[X <= x], otherwise, P[X > x]

Details

The definition for the K-inflated Poisson distribution.

Value

The functions KIPO return a gamlss.family object which can be used to fit K-inflated Poisson
distribution in the gamlss() function.

Author(s)

Saeed Mohammadpour <<s.mohammadpour1111@gamlil.com>>, Mikis Stasinopoulos <<d.stasinopoulos@londonmet.ac.uk>>

References


See Also

`gamlss.family`, `KIPO`

Examples

```r
# gives information about the default links for the Poisson distribution type II
KIPO()

# generate zero inflated Poisson distribution
gen.Kinf(family="PO", kinf=0)

# generate random sample from zero inflated Poisson distribution
x<-rinf0PO(1000, mu=1, sigma=.1)

# fit the zero inflated Poisson distribution using gamlss
data<-data.frame(x=x)
## Not run:
gamlss(x~1, family=inf0PO, data=data)
histDist(x, family=inf0PO)
## End(Not run)

# generated one inflated Poisson distribution
gen.Kinf(family="PO", kinf=1)

# generate random sample from one inflated Poisson distribution
x<-rinf1PO(1000, mu=1, sigma=.1)

# fit the one inflated Poisson distribution using gamlss
data<-data.frame(x=x)
## Not run:
gamlss(x~1, family=inf1PO, data=data)
histDist(x, family=inf1PO)
## End(Not run)
```
KISI

K-inflated sichel distributions for fitting a GAMLSS model

Description

The function KISI defines the K-inflated sichel distribution, a four parameter distribution, for a gamlss.family object to be used in GAMLSS fitting using the function gamlss(). The functions dKISI, pKISI, qKISI and rKISI define the density, distribution function, quantile function and random generation for the K-inflated sichel, KISI(), distribution.

Usage

KISI(mu.link = "log", sigma.link = "log", nu.link = "identity",

tau.link = "logit", kinf="K")

dKISI(x, mu = 1, sigma = 1, nu = -0.5, tau = 0.1, kinf=0, log = FALSE)

pKISI(q, mu = 1, sigma = 1, nu = -0.5, tau = 0.1, kinf=0, lower.tail = TRUE,

log.p = FALSE)

qKISI(p, mu = 1, sigma = 1, nu = -0.5, tau = 0.1, kinf=0, lower.tail = TRUE,

log.p = FALSE, max.value = 10000)

rKISI(n, mu = 1, sigma = 1, nu = -0.5, tau = 0.1, kinf=0, max.value = 10000)
Arguments

- `mu.link`: Defines the \( \mu \) link, with "log" link as the default for the \( \mu \) parameter.
- `sigma.link`: Defines the \( \sigma \) link, with "log" link as the default for the \( \sigma \) parameter.
- `nu.link`: Defines the \( \nu \) link, with "identity" link as the default for the \( \nu \) parameter.
- `tau.link`: Defines the \( \tau \) link, with "logit" link as the default for the \( \tau \) parameter.
- `x`: vector of (non-negative integer) quantiles.
- `mu`: vector of positive \( \mu \).
- `sigma`: vector of positive dispersion parameter.
- `nu`: vector of \( \nu \).
- `tau`: vector of inflated point probability.
- `p`: vector of probabilities.
- `q`: vector of quantiles.
- `n`: number of random values to return.
- `kinf`: defines inflated point in generating K-inflated distribution.
- `log,log.p`: logical; if TRUE, probabilities p are given as log(p).
- `lower.tail`: logical; if TRUE (default), probabilities are \( P[X \leq x] \), otherwise, \( P[X > x] \).
- `max.value`: a constant, set to the default value of 10000 for how far the algorithm should look for q.

Details

The definition for the K-inflated sichel distribution.

Value

The functions KISI return a `gamlss.family` object which can be used to fit K-inflated sichel distribution in the `gamlss()` function.

Author(s)

Saeed Mohammadpour <<s.mohammadpour1111@gamlil.com>>, Mikis Stasinopoulos <<d.stasinopoulos@londonmet.ac.uk>>

References


See Also

`gamlss.family`, `KISICHEL`

Examples

```r
# gives information about the default links for the Sichel distribution
KISI()
#--------------------------------------------------

# generate zero inflated sichel distribution
gen.Kinf(family=SI, kinf=0)

# generate random sample from zero inflated sichel distribution
x<-rinf0SI(1000, mu=1, sigma=.5, nu=.2, tau=.2)

# fit the zero inflated sichel distribution using gamlss
data<-data.frame(x=x)
## Not run:
gamlss(x~1, family=inf0SI, data=data)
histDist(x, family=inf0SI)
## End(Not run)
#--------------------------------------------------

# generated one inflated sichel distribution
gen.Kinf(family=SI, kinf=1)

# generate random sample from one inflated sichel distribution
x<-rinf1SI(1000, mu=1, sigma=.5, nu=.2, tau=.2)

# fit the one inflated sichel distribution using gamlss
data<-data.frame(x=x)
## Not run:
gamlss(x~1, family=inf1SI, data=data)
histDist(x, family=inf1SI)
## End(Not run)
#--------------------------------------------------

mu=4; sigma=.5; nu=.2; tau=.2;
par(mgp=c(2,1,0),mar=c(4,4,4,1)+.1)

#plot the pdf using plot
plot(function(x) dinf1SI(x, mu=mu, sigma=sigma, nu=nu, tau=tau), from=0, to=20, n=20+1, type="h", xlab="x", ylab="f(x)", cex.lab=1.5)
```
KISICHEL

K-inflated sichel distributions for fitting a GAMLSS model

Description

The function KISICHEL defines the K-inflated sichel distribution, a four parameter distribution, for a gamlss.family object to be used in GAMLSS fitting using the function gamlss(). The functions dKISICHEL, pKISICHEL, qKISICHEL and rKISICHEL define the density, distribution function, quantile function and random generation for the K-inflated sichel, KISICHEL(), distribution.

Usage

KISICHEL(mu.link = "log", sigma.link = "log", nu.link = "identity", tau.link = "logit", kinf="K")

dKISICHEL(x, mu = 1, sigma = 1, nu = -0.5, tau = 0.1, kinf=0, log = FALSE)

pKISICHEL(q, mu = 1, sigma = 1, nu = -0.5, tau = 0.1, kinf=0, lower.tail = TRUE, log.p = FALSE)

qKISICHEL(p, mu = 1, sigma = 1, nu = -0.5, tau = 0.1, kinf=0, lower.tail = TRUE, log.p = FALSE, max.value = 10000)

rKISICHEL(n, mu = 1, sigma = 1, nu = -0.5, tau = 0.1, kinf = 0, max.value = 10000)
Arguments

mu.link Defines the mu.link, with "log" link as the default for the mu parameter
sigma.link Defines the sigma.link, with "log" link as the default for the sigma parameter
nu.link Defines the nu.link, with "identity" link as the default for the nu parameter
tau.link Defines the tau.link, with "logit" link as the default for the tau parameter
x vector of (non-negative integer) quantiles
mu vector of positive means
sigma vector of positive dispersion parameter
nu vector of nu
tau vector of inflated point probability
p vector of probabilities
q vector of quantiles
n number of random values to return
kinf defines inflated point in generating K-inflated distribution
log,log.p logical; if TRUE, probabilities p are given as log(p)
lower.tail logical; if TRUE (default), probabilities are P[X <= x], otherwise, P[X > x]
max.value a constant, set to the default value of 10000 for how far the algorithm should look for q

Details

The definition for the K-inflated sichel distribution.

Value

The functions KISICHEL return a gamlss.family object which can be used to fit K-inflated sichel distribution in the gamlss() function.

Author(s)

Saeed Mohammadpour <<s.mohammadpour1111@gamlil.com>>, Mikis Stasinopoulos <<d.stasinopoulos@londonmet.ac.uk>>

References


**See Also**
gamlss.family, KISICHEL

**Examples**

```r
# gives information about the default links for the Sichel distribution
KISICHEL()

# generate zero inflated sichel distribution
gen.Kinf(family=SICHEL, kinf=0)

# generate random sample from zero inflated sichel distribution
x<-rinf0SICHEL(1000, mu=1, sigma=.5, nu=.2, tau=.2)

# fit the zero inflated sichel distribution using gamlss
data<-data.frame(x=x)
## Not run:
gamlss(x~1, family=inf0SICHEL, data=data)
histDist(x, family=inf0SICHEL)
## End(Not run)

# generated one inflated sichel distribution
gen.Kinf(family=SICHEL, kinf=1)

# generate random sample from one inflated sichel distribution
x<-rinf1SICHEL(1000, mu=1, sigma=.5, nu=.2, tau=.2)

# fit the one inflated sichel distribution using gamlss
data<-data.frame(x=x)
## Not run:
gamlss(x~1, family=inf1SICHEL, data=data)
histDist(x, family=inf1SICHEL)
## End(Not run)

mu=4; sigma=.5; nu=.2; tau=.2;
par(mgp=c(2,1,0),mar=c(4,4,4,1)+0.1)

#plot the pdf using plot
plot(function(x) dinf1SICHEL(x, mu=mu, sigma=sigma, nu=nu, tau=tau), from=0, to=20, n=20+1, type="h", xlab="x", ylab="f(x)", cex.lab=1.5)
```
KIWARING

K-inflated Waring distributions for fitting a GAMLSS model

Description

The function KIWARING defines the K-inflated Waring distribution, a three parameter distribution, for a gamlss.family object to be used in GAMLSS fitting using the function gamlss(). The functions dKIWARING, pKIWARING, qKIWARING and rKIWARING define the density, distribution function, quantile function and random generation for the K-inflated Waring, KIWARING(), distribution.

Usage

KIWARING(mu.link = "log", sigma.link = "log", nu.link = "logit", kinf="K")
dKIWARING(x, mu = 1, sigma = 1, nu = 0.3, kinf=0 ,log = FALSE)
pKIWARING(q, mu = 1, sigma = 1, nu = 0.3, kinf=0, lower.tail = TRUE, log.p = FALSE)
qKIWARING(p, mu = 1, sigma = 1, nu = 0.3, kinf=0, lower.tail = TRUE, log.p = FALSE)
rKIWARING(n, mu = 1, sigma = 1, nu = 0.3, kinf=0)

Arguments

mu.link Defines the mu.link, with "log" link as the default for the mu parameter
KIWARING

sigma.link  Defines the sigma.link, with "log" link as the default for the sigma parameter
nu.link  Defines the nu.link, with "logit" link as the default for the nu parameter
x  vector of (non-negative integer) quantiles
mu  vector of positive means
sigma  vector of positive dispersion parameter
nu  vector of inflated point probability
p  vector of probabilities
q  vector of quantiles
n  number of random values to return
kinf  defines inflated point in generating K-inflated distribution
log, log.p  logical; if TRUE, probabilities p are given as log(p)
lower.tail  logical; if TRUE (default), probabilities are P[X <= x], otherwise, P[X > x]

Details

The definition for the K-inflated Waring distribution.

Value

The functions KIWARING return a gamlss.family object which can be used to fit K-inflated Waring distribution in the gamlss() function.

Author(s)

Saeed Mohammadpour <<s.mohammadpour1111@gamlil.com>>, Mikis Stasinopoulos <<d.stasinopoulos@londonmet.ac.uk>>

References

See Also

gamlss.family.KIWARING

Examples

# gives information about the default links for the Waring distribution
KIWARING()

# generate zero inflated Waring distribution
gen.Kinf(family=WARING, kinf=0)

# generate random sample from zero inflated Waring distribution
x<-rinf0WARING(1000, mu=1, sigma=.5, nu=.2)

# fit the zero inflated Waring distribution using gamlss
data<-data.frame(x=x)
## Not run:
gamlss(x~1, family=inf0WARING, data=data)
histDist(x, family=inf0WARING)
## End(Not run)

# generated one inflated Waring distribution
gen.Kinf(family=WARING, kinf=1)

# generate random sample from one inflated Waring distribution
x<-rinf1WARING(1000, mu=1, sigma=.5, nu=.2)

# fit the one inflated Waring distribution using gamlss
data<-data.frame(x=x)
## Not run:
gamlss(x~1, family=inf1WARING, data=data)
histDist(x, family=inf1WARING)
## End(Not run)

mu=4; sigma=.5; nu=.2;
par(mgp=c(2,1,0),mar=c(4,4,4,1)+0.1)

#plot the pdf using plot
plot(function(x) dinf1WARING(x, mu=mu, sigma=sigma, nu=nu), from=0, to=20,
    n=20+1, type="h",xlab="x",ylab="f(x)",cex.lab=1.5)
##-----------------------------------------------

#plot the cdf using plot
ccdf <- stepfun(0:19, c(0,pinf1WARING(0:19, mu=mu, sigma=sigma, nu=nu)), f = 0)
plot(ccdf, xlab="x", ylab="F(x)", verticals=FALSE, cex.points=.8, pch=16, main="",cex.lab=1.5)
##-----------------------------------------------
KIYULE

K-inflated Yule distributions for fitting a GAMLSS model

Description

The function KIYULE defines the K-inflated Yule distribution, a two parameter distribution, for a
\texttt{gamlss.family} object to be used in GAMLSS fitting using the function \texttt{gamlss()}. The func-
tions \texttt{dKIYULE}, \texttt{pKIYULE}, \texttt{qKIYULE} and \texttt{rKIYULE} define the density, distribution function, quantile
function and random generation for the K-inflated Yule, \texttt{KIYULE()}, distribution.

Usage

\begin{verbatim}
KIYULE(mu.link = "log", sigma.link = "logit", kinf="K")
dKIYULE(x, mu = 1, sigma = 0.1, kinf=0, log = FALSE)
pKIYULE(q, mu = 1, sigma = 0.1, kinf=0, lower.tail = TRUE, log.p = FALSE)
qKIYULE(p, mu = 1, sigma = 0.1, kinf=0, lower.tail = TRUE, log.p = FALSE)
rKIYULE(n, mu = 1, sigma = 0.1, kinf=0)
\end{verbatim}

Arguments

\begin{verbatim}
mu.link \hspace{1cm} \text{Defines the mu.link, with "log" link as the default for the mu parameter}
sigma.link \hspace{1cm} \text{Defines the sigma.link, with "logit" link as the default for the sigma parameter}
x \hspace{1cm} \text{vector of (non-negative integer) quantiles}
mu \hspace{1cm} \text{vector of positive means}
sigma \hspace{1cm} \text{vector of inflated point probability}
p \hspace{1cm} \text{vector of probabilities}
q \hspace{1cm} \text{vector of quantiles}
n \hspace{1cm} \text{number of random values to return}
\end{verbatim}
kiyule

kihf defines inflated point in generating K-inflated distribution
log.log.p logical; if TRUE, probabilities p are given as log(p)
lower.tail logical; if TRUE (default), probabilities are P[X <= x], otherwise, P[X > x]

Details

The definition for the K-inflated Yule distribution.

Value

The functions KIYULE return a `gamlss.family` object which can be used to fit K-inflated Yule distribution in the `gamlss()` function.

Author(s)

Saeed Mohammadpour <<s.mohammadpour1111@gamlil.com>>, Mikis Stasinopoulos <<d.stasinopoulos@londonmet.ac.uk>>

References


See Also

`gamlss.family`, `KIYULE`

Examples

```r
# gives information about the default links for the Yule distribution type II
KIYULE()
# generate zero inflated Yule distribution
```
gen.Kinf(family=YULE, kinf=0)

# generate random sample from zero inflated Yule distribution
x<-rinf0YULE(1000, mu=1, sigma=.2)

# fit the zero inflated Yule distribution using gamlss
data<-data.frame(x=x)
## Not run:
gamlss(x~1, family=infl0YULE, data=data)
histDist(x, family=infl0YULE)
## End(Not run)
#---------------------------------------------------------------

# generated one inflated Yule distribution
gen.Kinf(family=YULE, kinf=1)

# generate random sample from one inflated Yule distribution
x<-rinf1YULE(1000, mu=1, sigma=.2)

# fit the one inflated Yule distribution using gamlss
data<-data.frame(x=x)
## Not run:
gamlss(x~1, family=infl1YULE, data=data)
histDist(x, family=infl1YULE)
## End(Not run)
#---------------------------------------------------------------

mu=1; sigma=.2;
par(mgp=c(2,1,0),mar=c(4,4,4,1)+0.1)

#plot the pdf using plot
plot(function(x) dinf1YULE(x, mu=mu, sigma=sigma), from=0, to=20, n=20+1,
    type="h",xlab="x",ylab="f(x)",cex.lab=1.5)
#---------------------------------------------------------------

#plot the cdf using plot
cdf <- stepfun(0:19, c(0,infl1YULE(0:19, mu=mu, sigma=sigma))), f = 0)
plot(cdf, xlab="x", ylab="F(x)", verticals=FALSE, cex.points=.8, pch=16, main="",cex.lab=1.5)
#---------------------------------------------------------------

#plot the qdf using plot
invcdf <- stepfun(seq(.01,.99,length=19), qinf1YULE(seq(.01,.99,length=20),mu, sigma), f = 0)
plot(invcdf, ylab=expression(x[p]=F^(-1)(p)), do.points=FALSE,verticals=TRUE,
    cex.points=.8, pch=16, main="",cex.lab=1.5, xlab="p")
#---------------------------------------------------------------

# generate random sample
Ni <- rinf1YULE(1000, mu=mu, sigma=sigma)
hist(Ni,breaks=seq(min(Ni)-0.5,max(Ni)+0.5,by=1),col="lightgray", main="",cex.lab=2)
barplot(table(Ni))
#---------------------------------------------------------------
Index

*Topic distribution
  gamlss.countKinf-package, 2
  gen.Kinf, 3
  KIBNB, 4
  KIDEL, 7
  KIDPO, 10
  KIGEO, 13
  KIGEOmo, 15
  KIGPO, 18
  KILG, 21
  KINBF, 24
  KINBI, 27
  KINBII, 30
  KIPIG, 32
  KIPO, 35
  KISI, 38
  KISICHEL, 41
  KIWARING, 44
  KIYULE, 47
*Topic package
gamlss.countKinf-package, 2
*Topic regression
gamlss.countKinf-package, 2
  dKIBNB (KIBNB), 4
  dKIDEL (KIDEL), 7
  dKIDPO (KIDPO), 10
  dKIGEO (KIGEO), 13
  dKIGEOmo (KIGEOmo), 15
  dKIGPO (KIGPO), 18
  dKILG (KILG), 21
  dKINBF (KINBF), 24
  dKINBI (KINBI), 27
  dKINBII (KINBII), 30
  dKIPIG (KIPIG), 32
  dKIPO (KIPO), 35
  dKISI (KISI), 38
  dKISICHEL (KISICHEL), 41
  dKIWARING (KIWARING), 44
  dKIYULE (KIYULE), 47

gamlss.countKinf
  (gamlss.countKinf-package), 2
  gamlss.countKinf-package, 2
  gamlss.family, 6, 9, 11, 14, 17, 20, 23, 25, 28, 31, 34, 37, 40, 43, 46, 48
  gen.Kinf, 3
  KIBNB, 4, 6
  KIDEL, 7, 9
  KIDPO, 10, 11
  KIGEO, 13, 14
  KIGEOmo, 15, 17
  KIGPO, 18, 20
  KILG, 21, 23
  KINBF, 24, 25
  KINBI, 27, 28
  KINBII, 30, 31
  KIPIG, 32, 34
  KIPO, 35, 37
  KISI, 38
  KISICHEL, 40, 41, 43
  KIWARING, 44, 46
  KIYULE, 47, 48

50