Package ‘ZOIP’

December 19, 2017

Title ZOIP Distribution, ZOIP Regression, ZOIP Mixed Regression

Description The ZOIP distribution (Zeros Ones Inflated Proportional) is a proportional data distribution inflated with zeros and/or ones, this distribution is defined on the most known proportional data distributions, the beta and simplex distribution, Jørgensen and Barndorff-Nielsen (1991) <doi:10.1016/0047-259X(91)90008-P>, also allows it to have different parameterizations of the beta distribution, Ferrari and Cribari-Neto (2004) <doi:10.1080/0266476042000214501>, Rigby and Stasinopoulos (2005) <doi:10.18637/jss.v023.i07>. The ZOIP distribution has four parameters, two of which correspond to the proportion of zeros and ones, and the other two correspond to the distribution of the proportional data of your choice. The 'ZOIP' package allows adjustments of regression models for fixed and mixed effects for proportional data inflated with zeros and/or ones.

Version 0.1

Depends R (>= 3.2.0)

License GPL (>= 2)

Encoding UTF-8

LazyData true

RoxygenNote 6.0.1

Imports rmutil, boot, numDeriv, GHQp, stats

Suggests knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation no

Author Juan Camilo Diaz Zapata [aut, cre]

Maintainer Juan Camilo Diaz Zapata <jucdiazza@unal.edu.co>

Repository CRAN

Date/Publication 2017-12-19 09:29:05 UTC

R topics documented:

coef.ZOIP *coef.ZOIP*

Description

Extract ZOIP model coefficients.

Usage

```
## S3 method for class 'ZOIP'
coef(object, ...)
```

Arguments

object	An object of class ZOIP.
...	other arguments.

Examples

```
#Test 1---------------------------------------------------
library(ZOIP)
library(boot)
n<-1000
x1<-stats::runif(n)
x2<-stats::runif(n)
b1<-0.3
b2<--0.5
b3<-0.9
sigma_i<-boot::inv.logit(b1+b2*x1+b3*x2)
c1<-0.2
c2<--1
c3<-0.1
mu_i<-boot::inv.logit(c1+c2*x1)
d1<-0.07
p0_i<-rep(d1,length(n))
e1<-0.02
e2<--4
```

```
p1_i<-boot::inv.logit(e1+e2*x2)
param<-cbind(mu_i,sigma_i,p0_i,p1_i)
system.time(y_i<-apply(param,1,function(x){rZOIP(1,mu=x[1],sigma=x[2],
                                            p0=x[3],p1=x[4],family='R-S')}))
data<-as.data.frame(cbind(y_i,x1,x2))
formula.mu=y_i~x1
formula.sigma=~x1+x2
formula.p0=~1
formula.p1=~x1+x2
link=c('logit','logit','identity','logit')
family='R-S'
mod<-RM.ZOIP(formula.mu=formula.mu,formula.sigma=formula.sigma,
         formula.p0=formula.p0,formula.p1=formula.p1,data=data,link=link,family=family)
coef(mod)
```

dZOIP *ZOIP Distribution*

Description

The dZOIP function defines the probability density function of the ZOIP distribution.

Usage

```
dZOIP(x, mu = 0.5, sigma = 0.1, p0 = 0.08333333, p1 = 0.08333333,
  family = "R-S", log = FALSE)
```

Arguments

x	quantiles vector.
mu	vector of location parameters.
sigma	vector of scale parameters.
p0	parameter of proportion of zeros.
p1	Parameter of proportion of ones.
family	choice of the parameterization or distribution, family = ’R-S’ parameterization beta distribution Rigby and Stasinopoulos, ’F-C’ distribution Beta parametrization Ferrari and Cribari-Neto, ’Original’ Beta distribution classic parameterization, ’Simplex’ simplex distribution.
log	logical; if TRUE, the probabilities of p will be given as log (p).

Details

x has ZOIP distribution with shape parameters "μ", scale "σ", proportion of zeros "$p0$" and proportion of ones "$p1$ ", has density: $p0$ if $x = 0$, $p1$ if $x = 1$, $(1 - p0 - p1)f(x; \mu, \sigma)$ yes $0 < x < 1$.

where $p0 \geq 0$ represents the probability that $x = 0, p1 \geq 0$ represents the probability that $x = 1, 0 \leq p0 + p1 \leq 1$ and $f(x; \mu, \sigma)$ represents some of the functions of probability density for proportional data, such as the beta distribution with its different parameterizations and the simplex distribution.

When family =' R-S 'uses the beta distribution with beta parameterization Rigby and Stasinopoulos (2005) which has a beta distribution function. μ is the parameter of mean and shape, plus σ is the dispersion parameter of the distribution. family =' F-C 'distribution Beta parametrization Ferrari and Cribari-Neto (2004), where $\sigma = \phi$, ϕ is a precision parameter. family =' Original 'beta distribution original parametrization where $\mu = a$, a parameter of form 1; $\sigma = b$, b parameter of form 2. family =' Simplex 'simplex distribution. proposed by Barndorff-Nielsen and Jørgensen (1991)

Examples

```
library(ZOIP)
dZOIP(x=0.5, mu = 0.2, sigma = 0.5, p0 = 0.2, p1 =0.2,family='R-S',log = FALSE)
dZOIP(x=0.5, mu = 0.2, sigma = 3, p0 = 0.2, p1 = 0.2,family='F-C',log = FALSE)
dZOIP(x=0.5, mu = 0.6, sigma = 2.4, p0 = 0.2, p1 = 0.2,family='Original',log = FALSE)
dZOIP(x=0.5, mu = 0.2, sigma = 3, p0 = 0, p1 = 0,family='Simplex',log = FALSE)
dZOIP(x=0.5, mu = 0.2, sigma = 0.5, p0 = 0.2, p1 =0,family='R-S',log = FALSE)
dZOIP(x=0.5, mu = 0.2, sigma = 0.5, p0 = 0, p1 =0.2,family='R-S',log = FALSE)
dZOIP(x=0.5, mu = 0.2, sigma = 0.5, p0 = 0, p1 =0,family='R-S',log = FALSE)
```

`print.ZOIP` *print.ZOIP*

Description

print a ZOIP model.

Usage

```
## S3 method for class 'ZOIP'
print(x, ...)
```

Arguments

`x`	An object of class `ZOIP`.
`...`	other arguments.

Examples

```
#Test 1--------------------------------------------------
library(ZOIP)
library(boot)
n<-1000
x1<-stats::runif(n)
x2<-stats::runif(n)

b1<-0.3
b2<--0.5
b3<-0.9
sigma_i<-boot::inv.logit(b1+b2*x1+b3*x2)

c1<-0.2
c2<--1
c3<-0.1
mu_i<-boot::inv.logit(c1+c2*x1)

d1<-0.07
p0_i<-rep(d1,length(n))

e1<-0.02
e2<--4
p1_i<-boot::inv.logit(e1+e2*x2)

param<-cbind(mu_i,sigma_i,p0_i,p1_i)

system.time(y_i<-apply(param,1,function(x){rZOIP(1,mu=x[1],sigma=x[2],
                                                 p0=x[3],p1=x[4],family='R-S')}))
data<-as.data.frame(cbind(y_i,x1,x2))

formula.mu=y_i~x1
formula.sigma=~x1+x2
formula.p0=~1
formula.p1=~x1+x2
link=c('logit','logit','identity','logit')
family='R-S'
mod<-RM.ZOIP(formula.mu=formula.mu,formula.sigma=formula.sigma,
          formula.p0=formula.p0,formula.p1=formula.p1,data=data,link=link,family=family)
mod
```

print.ZOIPM *print.ZOIPM*

Description

print a ZOIP model mixed.

Usage

```
## S3 method for class 'ZOIPM'
print(x, ...)
```

Arguments

`x`	An object of class `ZOIPM`.
`...`	other arguments.

Examples

```
library(ZOIP)

N<-2
ni<-10
set.seed(12345)
Ciudad <- rep(1:N, each=ni)
Total_mora<-rexp(N*ni,rate=1)
set.seed(12345)
b0i <- rep(rnorm(n=N,sd=0.5), each=ni)
set.seed(12345)
b1i <- rep(rnorm(n=N,sd=0.4), each=ni)

neta <- (-1.13+b0i)+0.33*Total_mora
neta2<-(0.33+b1i)+0.14*Total_mora

mu <- 1 / (1 + exp(-neta))
sigma <- 1 / (1 + exp(-neta2))

p0 <- 0.05
p1 <- 0.05

mu[mu==1] <- 0.999
mu[mu==0] <- 0.001

sigma[sigma==1] <- 0.999
sigma[sigma==0] <- 0.001
family<-'R-S'
set.seed(12345)
Y <- rZOIP(n=length(mu), mu = mu, sigma = sigma ,p0=p0,p1=p1,family=family)

data_sim<-data.frame(Y,Total_mora,Ciudad)

n.points <- 3
pruning <- TRUE

formula.mu=Y~Total_mora
formula.sigma=~Total_mora
formula.p0=~1
formula.p1=~1
formula.random= ~ 1 | Ciudad
link=c('logit','logit','identity','identity')
optimizer<-'nlminb'

mod<-RMM.ZOIP(formula.mu=formula.mu,formula.sigma=formula.sigma,formula.p0=formula.p0,
           formula.p1=formula.p1,data=data_sim,formula.random=formula.random,link=link,
             family=family,optimizer=optimizer,n.points=n.points,pruning=pruning)
mod
```

pZOIP *ZOIP Distribution*

Description

The pZOIP function defines the cumulative distribution function of the ZOIP distribution.

Usage

```
pZOIP(q, mu = 0.5, sigma = 0.1, p0 = 0.08333333, p1 = 0.08333333,
  family = "R-S", lower.tail = TRUE, log.p = FALSE)
```

Arguments

q	quantiles vector.
mu	vector of location parameters.
sigma	vector of scale parameters.
p0	parameter of proportion of zeros.
p1	Parameter of proportion of ones.
family	choice of the parameterization or distribution, family = 'R-S' parameterization beta distribution Rigby and Stasinopoulos, 'F-C' distribution Beta parametrization Ferrari and Cribari-Neto, 'Original' Beta distribution classic parameterization, 'Simplex' simplex distribution.
lower.tail	logical; if TRUE (default), probabilities will be P [X <= x], otherwise, P [X> x].
log.p	logical; if TRUE, the probabilities of p will be given as log (p).

Details

x has ZOIP distribution with shape parameters "μ", scale "σ", proportion of zeros "$p0$" and proportion of ones "$p1$ ", has density: $p0$ if $x = 0$, $p1$ if $x = 1$, $(1 - p0 - p1)f(x; \mu, \sigma)$ yes $0 < x < 1$.

where $p0 \geq 0$ represents the probability that $x = 0, p1 \geq 0$ represents the probability that $x = 1, 0 \leq p0 + p1 \leq 1$ and $f(x; \mu, \sigma)$ represents some of the functions of probability density for proportional data, such as the beta distribution with its different parameterizations and the simplex distribution.

When family =' R-S 'uses the beta distribution with beta parameterization Rigby and Stasinopoulos (2005) which has a beta distribution function. μ is the parameter of mean and shape, plus σ is the dispersion parameter of the distribution. family =' F-C 'distribution Beta parametrization Ferrari and Cribari-Neto (2004), where $\sigma = \phi$, ϕ is a precision parameter. family =' Original 'beta distribution original parametrization where $\mu = a$, a parameter of form 1; $\sigma = b$, b parameter of form 2. family =' Simplex 'simplex distribution. proposed by Barndorff-Nielsen and Jørgensen (1991)

Examples

```
library(ZOIP)
pZOIP(q=0.5, mu = 0.2, sigma = 0.5, p0 = 0.2, p1 = 0.2,family='R-S',log = FALSE)
pZOIP(q=0.5, mu = 0.2, sigma = 3, p0 = 0.2, p1 = 0.2,family='F-C',log = FALSE)
pZOIP(q=0.5, mu = 0.6, sigma = 2.4, p0 = 0.2, p1 = 0.2,family='Original',log = FALSE)
pZOIP(q=0.5, mu = 0.2, sigma = 3, p0 = 0.2, p1 = 0.2,family='Simplex',log = FALSE)

pZOIP(q=0.5, mu = 0.2, sigma = 0.5, p0 = 0, p1 = 0.2,family='R-S',log = FALSE)
pZOIP(q=0.5, mu = 0.2, sigma = 3, p0 = 0, p1 = 0.2,family='F-C',log = FALSE)
pZOIP(q=0.5, mu = 0.6, sigma = 2.4, p0 = 0, p1 = 0.2,family='Original',log = FALSE)
pZOIP(q=0.5, mu = 0.2, sigma = 3, p0 = 0, p1 = 0.2,family='Simplex',log = FALSE)

pZOIP(q=0.5, mu = 0.2, sigma = 0.5, p0 = 0.2, p1 = 0,family='R-S',log = FALSE)
pZOIP(q=0.5, mu = 0.2, sigma = 3, p0 = 0.2, p1 = 0,family='F-C',log = FALSE)
pZOIP(q=0.5, mu = 0.6, sigma = 2.4, p0 = 0.2, p1 = 0,family='Original',log = FALSE)
pZOIP(q=0.5, mu = 0.2, sigma = 3, p0 = 0.2, p1 = 0,family='Simplex',log = FALSE)

pZOIP(q=0.5, mu = 0.2, sigma = 0.5, p0 = 0, p1 = 0,family='R-S',log = FALSE)
pZOIP(q=0.5, mu = 0.2, sigma = 3, p0 = 0, p1 = 0,family='F-C',log = FALSE)
pZOIP(q=0.5, mu = 0.6, sigma = 2.4, p0 = 0, p1 = 0,family='Original',log = FALSE)
pZOIP(q=0.5, mu = 0.2, sigma = 3, p0 = 0, p1 = 0,family='Simplex',log = FALSE)
```

qZOIP *ZOIP Distribution*

Description

The qZOIP function defines the quantile function of the ZOIP distribution.

Usage

```
qZOIP(p, mu = 0.5, sigma = 0.1, p0 = 0.08333333, p1 = 0.08333333,
  family = "R-S", lower.tail = TRUE, log.p = FALSE)
```

Arguments

p	vector of probabilities.
mu	vector of location parameters.
sigma	vector of scale parameters.
p0	parameter of proportion of zeros.

`p1` Parameter of proportion of ones.

`family` choice of the parameterization or distribution, family = 'R-S' parameterization beta distribution Rigby and Stasinopoulos, 'F-C' distribution Beta parametrization Ferrari and Cribari-Neto, 'Original' Beta distribution classic parameterization, 'Simplex' simplex distribution.

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

`log.p` logical; if TRUE, the probabilities of p will be given as log (p).

Details

x has ZOIP distribution with shape parameters "μ", scale "σ", proportion of zeros "$p0$" and proportion of ones "$p1$ ", has density: $p0$ if $x = 0$, $p1$ if $x = 1$, $(1 - p0 - p1)f(x; \mu, \sigma)$ yes $0 < x < 1$.

where $p0 \geq 0$ represents the probability that $x = 0, p1 \geq 0$ represents the probability that $x = 1, 0 \leq p0 + p1 \leq 1$ and $f(x; \mu, \sigma)$ represents some of the functions of probability density for proportional data, such as the beta distribution with its different parameterizations and the simplex distribution.

When family =' R-S 'uses the beta distribution with beta parameterization Rigby and Stasinopoulos (2005) which has a beta distribution function. μ is the parameter of mean and shape, plus σ is the dispersion parameter of the distribution. family =' F-C 'distribution Beta parametrization Ferrari and Cribari-Neto (2004), where $\sigma = \phi$, ϕ is a precision parameter. family =' Original 'beta distribution original parametrization where $\mu = a$, a parameter of form 1; $\sigma = b$, b parameter of form 2. family =' Simplex 'simplex distribution. proposed by Barndorff-Nielsen and Jørgensen (1991)

Examples

```
library(ZOIP)
qZOIP(p=0.7, mu = 0.2, sigma = 0.5, p0 = 0.2, p1 = 0.2,family='R-S',log = FALSE)
qZOIP(p=0.7, mu = 0.2, sigma = 3, p0 = 0.2, p1 = 0.2,family='F-C',log = FALSE)
qZOIP(p=0.7, mu = 0.6, sigma = 2.4, p0 = 0.2, p1 = 0.2,family='Original',log = FALSE)
qZOIP(p=0.7, mu = 0.2, sigma = 3, p0 = 0.2, p1 = 0.2,family='Simplex',log = FALSE)

qZOIP(p=0.7, mu = 0.2, sigma = 0.5, p0 = 0.2, p1 = 0,family='R-S',log = FALSE)
qZOIP(p=0.7, mu = 0.2, sigma = 3, p0 = 0.2, p1 = 0,family='F-C',log = FALSE)
qZOIP(p=0.7, mu = 0.6, sigma = 2.4, p0 = 0.2, p1 = 0,family='Original',log = FALSE)
qZOIP(p=0.7, mu = 0.2, sigma = 3, p0 = 0.2, p1 = 0,family='Simplex',log = FALSE)

qZOIP(p=0.7, mu = 0.2, sigma = 0.5, p0 = 0, p1 = 0.2,family='R-S',log = FALSE)
qZOIP(p=0.7, mu = 0.2, sigma = 3, p0 = 0, p1 = 0.2,family='F-C',log = FALSE)
qZOIP(p=0.7, mu = 0.6, sigma = 2.4, p0 = 0, p1 = 0.2,family='Original',log = FALSE)
qZOIP(p=0.7, mu = 0.2, sigma = 3, p0 = 0, p1 = 0.2,family='Simplex',log = FALSE)

qZOIP(p=0.7, mu = 0.2, sigma = 0.5, p0 = 0, p1 = 0,family='R-S',log = FALSE)
qZOIP(p=0.7, mu = 0.2, sigma = 3, p0 = 0, p1 = 0,family='F-C',log = FALSE)
qZOIP(p=0.7, mu = 0.6, sigma = 2.4, p0 = 0, p1 = 0,family='Original',log = FALSE)
qZOIP(p=0.7, mu = 0.2, sigma = 3, p0 = 0, p1 = 0,family='Simplex',log = FALSE)
```

RM.ZOIP *ZOIP regression with fixed efects*

Description

The RM.ZOIP function adjusts a ZOIP regression model via maximum likelihood. the model may or may not include covariables in any of its parameters, just as it can be a bilaterally inflated model, unilaterally or without parameters of inflation

Usage

```
RM.ZOIP(formula.mu, formula.sigma = ~1, formula.p0 = ~1, formula.p1 = ~1,
  data, link = c("identity", "identity", "identity", "identity"),
  family = "R-S", optimizer = "nlminb")
```

Arguments

`formula.mu` Formula that defines the regression function for mu, p.e and ~ x1 + x2, it is necessary to define the response variable.

`formula.sigma` Formula that defines the regression function for the sigma parameter, a possible value is ~ x1, by default ~ 1.

`formula.p0` Formula that defines the regression function for p0, a possible value is ~ x1, by default ~ 1.

`formula.p1` Formula that defines the regression function for p1, a possible value is ~ x1, by default ~ 1.

`data` It is the data set in data.frame format where it must contain the names of the columns as they are in the formulas.

`link` It is a vector with the appropriate link functions for each parameter to be estimated according to the options chosen in the family and formula parameters. If the regression model does not have covariables, the identity option should be used as a link function, regardless of the value chosen in the family, possible options are logit, log, default link = c ('identity', 'identity', 'identity', 'identity').

`family` choice of the parameterization or distribution, family = 'R-S' parameterization beta distribution Rigby and Stasinopoulos, 'F-C' distribution Beta parametrization Ferrari and Cribari-Neto, 'Original' Beta distribution classic parameterization, 'Simplex' simplex distribution.

`optimizer` Choice of the optimizer, used to find the convergence of the maximum likelihood. you can choose the value of 'nlminb' or 'optim', by default 'nlminb'.

Examples

```
#Test 1--------------------------------------------------
library(ZOIP)
library(boot)
library(numDeriv)
n<-1000
x1<-stats::runif(n)
x2<-stats::runif(n)

b1<-0.3
b2<--0.5
b3<-0.9
sigma_i<-exp(b1+b2*x1+b3*x2)

c1<-0.2
c2<--1
c3<-0.1
mu_i<-exp(c1+c2*x1)

d1<-0.07
p0_i<-rep(d1,length(n))

e1<-0.02
e2<--4
p1_i<-boot::inv.logit(e1+e2*x2)

param<-cbind(mu_i,sigma_i,p0_i,p1_i)

system.time(y_i<-apply(param,1,function(x){rZOIP(1,mu=x[1],sigma=x[2],
                                               p0=x[3],p1=x[4],family='Original')}))
data<-as.data.frame(cbind(y_i,x1,x2))

formula.mu=y_i~x1
formula.sigma=~x1+x2
formula.p0=~1
formula.p1=~x2
link=c('log','log','identity','logit')
family='Original'
mod<-RM.ZOIP(formula.mu=formula.mu,formula.sigma=formula.sigma,formula.p0=formula.p0,
             formula.p1=formula.p1,data=data,link=link,family=family)
mod
summary(mod)
```

RMM.ZOIP *ZOIP regression with mixed efects*

Description

function RMM.ZOIP fits a mixed ZOIP regression model with random intercepts normals in the mean and dispersion parameter, the estimation is done via maximum likelihood and the gauss-hermite adaptive quadrangle with or without pruning. the model may or may not include effects fixed in any of its parameters, just as it can be a bilaterally inflated model, unilaterally or without parameters inflated.

Usage

```
RMM.ZOIP(formula.mu, formula.sigma = ~1, formula.p0 = ~1, formula.p1 = ~1,
  data, formula.random, link = c("identity", "identity", "identity",
  "identity"), family = "R-S", optimizer = "nlminb", n.points = 11,
  pruning = TRUE)
```

Arguments

`formula.mu` Formula that defines the regression function for mu, p.e and ~ x1 + x2, it is necessary to define the response variable.

`formula.sigma` Formula that defines the regression function for the sigma parameter, a possible value is ~ x1, by default ~ 1.

`formula.p0` Formula that defines the regression function for p0, a possible value is ~ x1, by default ~ 1.

`formula.p1` Formula that defines the regression function for p1, a possible value is ~ x1, by default ~ 1.

`data` It is the data set in data.frame format where it must contain the names of the columns as they are in the formulas.

`formula.random` Formula that defines the mixed effect within the model, it should be only the random intercept that will be taken into account in the parameter of the mean and the dispersion, the admissible structure is the following formula.random = ~ 1 | G1, where G1 is the variable that indicates the groups or subjects in the model, should always be defined.

`link` It is a vector with the appropriate link functions for each parameter to be estimated according to the options chosen in the family and formula parameters. If the regression model does not have covariables, the identity option should be used as a link function, regardless of the value chosen in the family, possible options are logit, log, default link = c (’identity’, ’identity’, ’identity’, ’identity’).

`family` choice of the parameterization or distribution, family = ’R-S’ parameterization beta distribution Rigby and Stasinopoulos, ’F-C’ distribution Beta parametrization Ferrari and Cribari-Neto, ’Original’ Beta distribution classic parameterization, ’Simplex’ simplex distribution.

`optimizer` Choice of the optimizer, used to find the convergence of the maximum likelihood. you can choose the value of ’nlminb’ or ’optim’, by default ’nlminb’.

`n.points` Number of points to use in the approximation of the likelihood function by means of quadrature Gauss-Hermite adaptive multidimensional, by default is 11, it is recommended not to give a very large value to this parameter, because it will significantly affect the times of convergence of the model.

`pruning` It is a Boolean value that indicates if pruning is used or not, for the quadrature of multidimensional Adaptive Gauss-Hermite, by default it is TRUE.

Examples

```
library(ZOIP)
```

```
N<-2
ni<-10
set.seed(12345)
Ciudad <- rep(1:N, each=ni)
Total_mora<-rexp(N*ni,rate=1)
set.seed(12345)
b0i <- rep(rnorm(n=N,sd=0.5), each=ni)
set.seed(12345)
b1i <- rep(rnorm(n=N,sd=0.4), each=ni)

neta <- (-1.13+b0i)+0.33*Total_mora
neta2<-(0.33+b1i)+0.14*Total_mora

mu <- 1 / (1 + exp(-neta))
sigma <- 1 / (1 + exp(-neta2))

p0 <- 0.05
p1 <- 0.05

mu[mu==1] <- 0.999
mu[mu==0] <- 0.001

sigma[sigma==1] <- 0.999
sigma[sigma==0] <- 0.001
family<-'R-S'
set.seed(12345)
Y <- rZOIP(n=length(mu), mu = mu, sigma = sigma ,p0=p0,p1=p1,family=family)

data_sim<-data.frame(Y,Total_mora,Ciudad)

n.points <- 3
pruning <- TRUE

formula.mu=Y~Total_mora
formula.sigma=~Total_mora
formula.p0=~1
formula.p1=~1
formula.random= ~ 1 | Ciudad
link=c('logit','logit','identity','identity')
optimizer<-'nlminb'

mod<-RMM.ZOIP(formula.mu=formula.mu,formula.sigma=formula.sigma,formula.p0=formula.p0,
            formula.p1=formula.p1,data=data_sim,formula.random=formula.random,link=link,
               family=family,optimizer=optimizer,n.points=n.points,pruning=pruning)
mod
```

rZOIP *ZOIP Distribution*

Description

The rZOIP function defines the random number generating function for the ZOIP distribution.

Usage

```
rZOIP(n, mu = 0.5, sigma = 0.1, p0 = 0.08333333, p1 = 0.08333333,
  family = "R-S")
```

Arguments

n	number of observations. If length (n)> 1, the length is taken to be the number required.
mu	vector of location parameters.
sigma	vector of scale parameters.
p0	parameter of proportion of zeros.
p1	Parameter of proportion of ones.
family	choice of the parameterization or distribution, family = 'R-S' parameterization beta distribution Rigby and Stasinopoulos, 'F-C' distribution Beta parametrization Ferrari and Cribari-Neto, 'Original' Beta distribution classic parameterization, 'Simplex' simplex distribution.

Details

x has ZOIP distribution with shape parameters "μ", scale "σ", proportion of zeros "$p0$" and proportion of ones "$p1$ ", has density: $p0$ if $x = 0$, $p1$ if $x = 1$, $(1 - p0 - p1)f(x; \mu, \sigma)$ yes $0 < x < 1$.

where $p0 \geq 0$ represents the probability that $x = 0, p1 \geq 0$ represents the probability that $x = 1, 0 \leq p0 + p1 \leq 1$ and $f(x; \mu, \sigma)$ represents some of the functions of probability density for proportional data, such as the beta distribution with its different parameterizations and the simplex distribution.

When family =' R-S 'uses the beta distribution with beta parameterization Rigby and Stasinopoulos (2005) which has a beta distribution function. μ is the parameter of mean and shape, plus σ is the dispersion parameter of the distribution. family =' F-C 'distribution Beta parametrization Ferrari and Cribari-Neto (2004), where $\sigma = \phi$, ϕ is a precision parameter. family =' Original 'beta distribution original parametrization where $\mu = a$, a parameter of form 1; $\sigma = b$, b parameter of form 2. family =' Simplex 'simplex distribution. proposed by Barndorff-Nielsen and Jørgensen (1991)

Examples

```
library(ZOIP)
a1<-rZOIP(n=1000, mu = 0.2, sigma = 0.5, p0 = 0.2, p1 = 0.2,family='R-S')
a2<-rZOIP(n=1000, mu = 0.2, sigma = 3, p0 = 0.2, p1 = 0.2,family='F-C')
a3<-rZOIP(n=1000, mu = 0.6, sigma = 2.4, p0 = 0.2, p1 = 0.2,family='Original')
system.time(a4<-rZOIP(n=10, mu = 0.2, sigma = 3, p0 = 0.2, p1 = 0.2,family='Simplex'))

plot(density(a1))
plot(density(a2))
plot(density(a3))
plot(density(a4))

a1<-rZOIP(n=1000, mu = 0.2, sigma = 0.5, p0 = 0.2, p1 = 0,family='R-S')
a2<-rZOIP(n=1000, mu = 0.2, sigma = 3, p0 = 0.2, p1 = 0,family='F-C')
a3<-rZOIP(n=1000, mu = 0.6, sigma = 2.4, p0 = 0.2, p1 = 0,family='Original')
system.time(a4<-rZOIP(n=10, mu = 0.2, sigma = 3, p0 = 0.2, p1 = 0,family='Simplex'))

plot(density(a1))
plot(density(a2))
plot(density(a3))
plot(density(a4))

a1<-rZOIP(n=1000, mu = 0.2, sigma = 0.5, p0 = 0, p1 = 0.2,family='R-S')
a2<-rZOIP(n=1000, mu = 0.2, sigma = 3, p0 = 0, p1 = 0.2,family='F-C')
a3<-rZOIP(n=1000, mu = 0.6, sigma = 2.4, p0 = 0, p1 = 0.2,family='Original')
system.time(a4<-rZOIP(n=10, mu = 0.2, sigma = 3, p0 = 0, p1 = 0.2,family='Simplex'))

plot(density(a1))
plot(density(a2))
plot(density(a3))
plot(density(a4))
```

summary.ZOIP *summary.ZOIP*

Description

Summarize a ZOIP model.

Usage

```
## S3 method for class 'ZOIP'
summary(object, ...)
```

Arguments

object	An object of class ZOIP.
...	other arguments.

Examples

```
#Test 1--------------------------------------------------
library(ZOIP)
library(boot)
n<-1000
x1<-stats::runif(n)
x2<-stats::runif(n)
```

```
b1<-0.3
b2<--0.5
b3<-0.9
sigma_i<-boot::inv.logit(b1+b2*x1+b3*x2)

c1<-0.2
c2<--1
c3<-0.1
mu_i<-boot::inv.logit(c1+c2*x1)

d1<-0.07
p0_i<-rep(d1,length(n))

e1<-0.02
e2<--4
p1_i<-boot::inv.logit(e1+e2*x2)

param<-cbind(mu_i,sigma_i,p0_i,p1_i)

system.time(y_i<-apply(param,1,function(x){rZOIP(1,mu=x[1],sigma=x[2],
                                             p0=x[3],p1=x[4],family='R-S')}))
data<-as.data.frame(cbind(y_i,x1,x2))

formula.mu=y_i~x1
formula.sigma=~x1+x2
formula.p0=~1
formula.p1=~x1+x2
link=c('logit','logit','identity','logit')
family='R-S'
mod<-RM.ZOIP(formula.mu=formula.mu,formula.sigma=formula.sigma,formula.p0=formula.p0,
            formula.p1=formula.p1,data=data,link=link,family=family)
summary(mod)
```

summary.ZOIPM *summary.ZOIPM*

Description

Summarize a ZOIP model mixed.

Usage

```
## S3 method for class 'ZOIPM'
summary(object, ...)
```

Arguments

object	An object of class ZOIPM.
...	other arguments.

Examples

```
library(ZOIP)

N<-2
ni<-10
set.seed(12345)
Ciudad <- rep(1:N, each=ni)
Total_mora<-rexp(N*ni,rate=1)
set.seed(12345)
b0i <- rep(rnorm(n=N,sd=0.5), each=ni)
set.seed(12345)
b1i <- rep(rnorm(n=N,sd=0.4), each=ni)

neta <- (-1.13+b0i)+0.33*Total_mora
neta2<-(0.33+b1i)+0.14*Total_mora

mu <- 1 / (1 + exp(-neta))
sigma <- 1 / (1 + exp(-neta2))

p0 <- 0.05
p1 <- 0.05

mu[mu==1] <- 0.999
mu[mu==0] <- 0.001

sigma[sigma==1] <- 0.999
sigma[sigma==0] <- 0.001
family<-'R-S'
set.seed(12345)
Y <- rZOIP(n=length(mu), mu = mu, sigma = sigma ,p0=p0,p1=p1,family=family)

data_sim<-data.frame(Y,Total_mora,Ciudad)

n.points <- 3
pruning <- TRUE

formula.mu=Y~Total_mora
formula.sigma=~Total_mora
formula.p0=~1
formula.p1=~1
formula.random= ~ 1 | Ciudad
link=c('logit','logit','identity','identity')
optimizer<-'nlminb'

mod<-RMM.ZOIP(formula.mu=formula.mu,formula.sigma=formula.sigma,formula.p0=formula.p0,
           formula.p1=formula.p1,data=data_sim,formula.random=formula.random,link=link,
             family=family,optimizer=optimizer,n.points=n.points,pruning=pruning)
summary(mod)
```

Index