Package ‘CopulaRegression’

February 19, 2015

Type Package
Title Bivariate Copula Based Regression Models
Version 0.1-5
Depends R (>= 2.11.0), MASS, VineCopula
Date 2014-09-04
Author Nicole Kraemer, Daniel Silvestrini
Maintainer Nicole Kraemer <kraemer_r_packages@yahoo.de>
Description This R-package presents a bivariate, copula-based model for the joint distribution of a pair of continuous and discrete random variables. The two marginal random variables are modeled via generalized linear models, and their joint distribution (represented by a parametric copula family) is estimated using maximum-likelihood techniques.
License GPL (>= 2.0)
NeedsCompilation no
Repository CRAN
Date/Publication 2014-09-04 13:43:24

R topics documented:

CopulaRegression-package .................................................. 2
copreg ................................................................. 3
density_conditional ..................................................... 5
density_joint .......................................................... 6
dgam ................................................................. 8
dpolicy_loss .......................................................... 9
dztz ................................................................. 10
D_u ................................................................. 11
epolicy_loss .......................................................... 12
loglik_joint .......................................................... 13
mle_joint .......................................................... 15
mle_marginal ......................................................... 16
Description

This R-package presents a bivariate, copula-based model for the joint distribution of a pair of continuous and discrete random variables. The two marginal random variables are modeled via generalized linear models, and their joint distribution (represented by a parametric copula family) is estimated using maximum-likelihood techniques.

Details

Package: CopulaRegression
Type: Package
Version: 0.1-5
Date: 2014-09-04
License: GPL >=2

Author(s)

Nicole Kraemer <kraemer_r_packages@yahoo.de>

References

Description

This function fits a joint, bivariate regression model for a Gamma generalized linear model and a (zero-truncated) Poisson generalized linear model.

Usage

copreg(x, y, R, S, family = 1, exposure, sd.error = FALSE, joint = TRUE, zt = TRUE)

Arguments

x  n observations of the Gamma variable
y  n observations of the (zero-truncated) Poisson variable
R  n x p design matrix for the Gamma model
S  n x q design matrix for the (zero-truncated) Poisson model
family  an integer defining the bivariate copula family: 1 = Gauss, 3 = Clayton, 4 = Gumbel, 5 = Frank
exposure  exposure time for the (zero-truncated) Poisson model, all entries of the vector have to be > 0. Default is a constant vector of 1.
SD.error  logical. Should the standard errors of the regression coefficients be returned? Default is FALSE.
joint  logical. Should the two generalized linear models be estimated jointly? Default is TRUE.
zt  logical. If zt = TRUE, we fit a marginal zero-truncated Poisson regression model. Otherwise, we fit a marginal Poisson regression model. Default is TRUE.

Details

We consider positive continuous random variables \( X_i \) and positive or non-negative count variables \( Y_i \). We model \( X_i \) in terms of a covariate vector \( r_i \) and \( Y_i \) in terms of a covariate vector \( s_i \). The marginal regression models are specified via

\[
X_i \sim \text{Gamma}(\mu_i, \delta)
\]

with \( \ln(\mu_i) = r_i^T \alpha \) for the continuous variable. For the count variable, if \( zt = \text{TRUE} \), we use a zero-truncated Poisson model,

\[
Y_i \sim \text{ZTP}(\lambda_i)
\]

with \( \ln(\lambda_i) = \ln(e_i) + s_i^T \beta \). Otherwise, we use a Poisson model. \( e_i \) denotes the exposure time.

Further, we assume that the dependency of \( X_i \) and \( Y_i \) is modeled in terms of a copula family with parameter \( \theta \).
Value

This is an object of class `copreg`

- `alpha`: estimated coefficients for X, including the intercept
- `beta`: estimated coefficients for Y, including the intercept
- `sd.alpha`: estimated standard deviation of `alpha` (if `sd.error=TRUE`)
- `sd.beta`: estimated standard deviation of `beta` (if `sd.error=TRUE`)
- `delta`: estimated dispersion parameter
- `theta`: estimated copula parameter if `joint=TRUE`, 0 otherwise (in combination with `family=1`, 0 corresponds to the independence assumption)
- `family`: copula family as provided in the function call if `joint=TRUE`, 1 otherwise (in combination with `theta=0`, 1 corresponds to the independence assumption)
- `ll`: loglikelihood of the estimated model, evaluated at each observation
- `loglik`: overall loglikelihood, i.e. sum of `ll`
- `alpha0`: estimated coefficients for X under independence, including the intercept
- `beta0`: estimated coefficients for Y under independence, including the intercept
- `sd.alpha0`: estimated standard deviation (if `sd.error=TRUE`)
- `sd.beta0`: estimated standard deviation (if `sd.error=TRUE`)
- `delta0`: estimated dispersion parameter under independence
- `theta0`: 0 (in combination with `family0=1`, 0 corresponds to the independence assumption)
- `family0`: 1 (in combination with `theta0=0`, 1 corresponds to the independence assumption)
- `ll0`: loglikelihood of the estimated model under independence, evaluated at each observation
- `loglik0`: overall loglikelihood, under independence, i.e. sum of `ll0`
- `zt`: The value of `zt`
- `tau_IFM`: estimated Kendall’s τ based on the marginal models, using inference from margins
- `theta_ifm`: estimated copula parameter, estimated via inference from margins
- `npar`: the number of estimated parameters in the model

Author(s)

Nicole Kraemer

References

density_conditional

See Also

mle_marginal, mle_joint, simulate_regression_data

Examples

```r
n<-200 # number of examples
R<-cbind(rep(1,n),rnorm(n)) # design matrices with intercept
alpha<-beta<-c(1,-1) # regression coefficients
exposure<-rep(1,n) # constant exposure
delta<-0.5 # dispersion parameter
tau<-0.3 # Kendall's tau
family=3 # Clayton copula
# simulate data
my.data<-simulate_regression_data(n, alpha, beta, R, S, delta, tau, family, TRUE, exposure)
x<-my.data[,1]
y<-my.data[,2]

# joint model without standard errors
my.model<-copreg(x, y, R, S, family, exposure, FALSE, TRUE)
```

density_conditional  Conditional density of \(Y\) given \(X\)

Description

Conditional density of a (zero-truncated) Poisson variable \(Y\) given \(X=x\) for a Gamma-distributed variable \(X\).

Usage

density_conditional(y, x, mu, delta, lambda, theta, family, zt)

Arguments

- **y**: vector at which the conditional density is evaluated
- **x**: conditioning value of the Gamma distributed variable
- **mu**: expectation of the Gamma distribution
- **delta**: dispersion parameter of the Gamma distribution
- **lambda**: parameter of the zero-truncated Poisson distribution
- **theta**: copula parameter
- **family**: an integer defining the bivariate copula family: 1 = Gauss, 3 = Clayton, 4=Gumbel, 5=Frank
- **zt**: logical. If zt=TRUE, we use a zero-truncated Poisson variable. Otherwise, we use a Poisson variable. Default is TRUE.
Details

For a Gamma distributed variable X and a (zero truncated) Poisson variable Y with joint density function \( f_{XY}(x, y) \), this function evaluates

\[
P(Y = y | X = x) = \frac{f_{XY}(x, y)}{f_X(x)}.
\]

The joint density function is determined by a copula family with copula parameter theta.

Value

vector of length \( \text{length}(y) \)

Author(s)

Nicole Kraemer, Daniel Silvestrini

References


See Also

density_joint

Examples

```r
out <- density_conditional(y=0:10, x=3, mu=1, delta=1, lambda=2, theta=0.5, family=1)
names(out) = 0:10
barplot(out)
```

---

**density_joint**

Joint density of X and Y

Description

Density of a Gamma distributed variable X and a (zero-truncated) Poisson variable Y if their joint distribution is defined via a copula

Usage

density_joint(x, y, mu, delta, lambda, theta, family, zt)
Arguments

- x: vector at which the density is evaluated
- y: vector at which the density is evaluated
- mu: expectation of the Gamma distribution
- delta: dispersion parameter of the Gamma distribution
- lambda: parameter of the zero-truncated Poisson distribution
- theta: copula parameter
- family: an integer defining the bivariate copula family: 1 = Gauss, 3 = Clayton, 4 = Gumbel, 5 = Frank
- zt: logical. If zt=TRUE, we use a zero-truncated Poisson variable. Otherwise, we use a Poisson variable. Default is TRUE.

Details

For a Gamma distributed variable X and a (zero truncated) Possion variable Y, their joint density function is given by

\[ f_{XY}(x, y) = f_X(x) \left( D_u(F_Y(y), F_X(x)|\theta) - D_u(F_Y(y-1), F_X(x)|\theta) \right) . \]

Here \( D_u \) is the h-function of a copula family \( \text{family} \) with copula parameter \( \theta \).

Note

We allow two options: If \( \mu \) and \( \lambda \) are vectors of the same length as \( x \) and \( y \), we evaluate the density for the corresponding parameter values. Otherwise, \( \lambda \) and \( \nu \) have to be numbers, and the parameters are the same for all entries of \( x \) and \( y \).

Author(s)

Nicole Kraemer

References


See Also

density_joint, D_u

Examples

```r
out <- density_joint(2, 3, mu=1, delta=1, lambda=4, theta=0.5, family=1)
```
dgam

Density of a Gamma variable

Description

Density of a Gamma-distributed variable in mean parametrization

Usage

dgam(y, mu, delta)

Arguments

y vector of length n
mu mean parameter; either a number or a vector of length n
delta dispersion parameter; a number

Value

density, evaluated at y, this is a vector of length n

Author(s)

Nicole Kraemer

See Also

pgam

Examples

y<-0:10
mu<-5
delta<-1
out<-dgam(y, mu, delta)
# out equals
out1<-dgamma(y, shape=1/delta, rate=1/(delta*mu))
dpolicy_loss

Description
Density of the policy loss

Usage

dpolicy_loss(l, mu, delta, lambda, theta, family, y.max = 300, zt=TRUE)

Arguments

  l       vector at which the density is evaluated
  mu      expectation of the Gamma distribution
  delta   dispersion parameter of the Gamma distribution
  lambda  parameter of the (zero-truncated) Poisson distribution
  theta   copula parameter
  family  an integer defining the bivariate copula family: 1 = Gauss, 3 = Clayton, 4=Gumbel, 5=Frank
  y.max   upper value of the finite sum that we use to approximate the infinite sum, see below for details
  zt      logical. If zt=TRUE, we use a zero-truncated Poisson variable. Otherwise, we use a Poisson variable. Default is TRUE.

Details
For a Gamma distributed variable X and a (zero truncated) Possion variable Y, the policy loss is defined as \( L = X \cdot Y \). Its density is an infinite sum of weighted Gamma densities. The parameter \( y.max \) is the upper value of the finite sum that approximates the infinite sum.

Value
density, evaluated at the vector l

Note
lambda and mu can be scalars, or vectors of the same length as l

Author(s)
Nicole Kraemer

References
See Also

epolicy_loss, qpolicy_loss

Examples

# example taken from the paper
library(VineCopula)
mu<-1000
delta<-0.09
lambda<-2.5
family<-1
theta<-BiCopTau2Par(tau=0.5,family=family)
l<-seq(1,7000,length=100)
out<-dpolicy_loss(l,mu,delta,lambda,theta,family)
plot(l,out,type="l",lwd=3,xlab="loss",ylab="density")

---

dztp

Density of a zero truncated Poisson variable

Description

Probability mass function of a zero truncated Poisson variable

Usage

dztp(y, lambda)

Arguments

y vector of length n
lambda either a number or a vector of length n

Value

density, evaluated at y, this is a vector of length n

Author(s)

Nicole Kraemer

See Also

pztp
Examples

```r
y<-0:10
lambda1<-5 # fixed lambda
out1<-dztp(y,lambda1)
lambda2<-2:12 # variable lambda
out2<-dztp(y,lambda2)
```

---

**D_u**

*H-function of the copula*

---

**Description**

H-function of the copula

**Usage**

```r
D_u(u, v, theta, family)
```

**Arguments**

- **u**: vector of [0,1] values
- **v**: vector of [0,1] values
- **theta**: copula parameter
- **family**: an integer defining the bivariate copula family: 1 = Gauss, 3 = Clayton, 4=Gumbel, 5=Frank

**Details**

For a bivariate copula family \( C(u, v|\theta) \), the h-function is defined as

\[
D_u(u, v) = \frac{\partial}{\partial u} C(u, v|\theta)
\]

**Value**

value of the h-function

**Note**

This is the function BiCopHfunc from the VineCopula package.

**Author(s)**

Nicole Kraemer

**See Also**

density_joint
epolicy_loss

Examples

\begin{verbatim}
  u<-v<-0.5
ttheta<-0.2
family=1
out<D_u(u,v,theta,family)
\end{verbatim}

Description

Expectation and variance of the policy loss

Usage

\begin{verbatim}
  epolicy_loss(mu, delta, lambda, theta, family, y.max = 300, zt=TRUE, compute.var=FALSE)
\end{verbatim}

Arguments

- **mu**: expectation of the Gamma distribution, can be a vector
- **delta**: dispersion parameter of the Gamma distribution
- **lambda**: parameter of the (zero truncated) Poisson distribution, can be a vector of the same length as mu
- **theta**: copula parameter
- **family**: an integer defining the bivariate copula family: 1 = Gauss, 3 = Clayton, 4=Gumbel, 5=Frank
- **y.max**: upper value of the finite sum that we use to approximate the infinite sum in the density, see below for details
- **zt**: logical. If zt=TRUE, we use a zero-truncated Poisson variable. Otherwise, we use a Poisson variable. Default is TRUE.
- **compute.var**: logical. If compute.var=TRUE, we also compute the variance of the policy loss. Default is FALSE.

Details

For a Gamma distributed variable X and a (zero truncated) Possion variable Y, the policy loss is defined as \( L = X \cdot Y \). Its density is an infinite sum of weighted Gamma densities. The parameter y.max is the upper value of the finite sum that approximates the infinite sum.

Value

- **mean**: expectation of the policy loss
- **var**: variance of the policy loss
loglik_joint

Author(s)

Nicole Kraemer

References


See Also
dpolicy_loss

Examples

library(VineCopula)
mu<-1000
delta<-0.09
lambda<-2.5
family<-3
theta<-BiCopTau2Par(tau=0.5,family=family)
out<-epolicy_loss(mu,delta,lambda,theta,family)

loglik_joint

Loglikelihood of the joint regression model

Description

Loglikelihood of the joint regression model

Usage

loglik_joint(alpha,beta,theta, delta, x, y, R, S, family, exposure, negative,zt)

Arguments

alpha The regression coefficients for the Gamma regression
beta The regression coefficients for the (zero-truncated) Poisson regression
theta The copula parameter
delta dispersion parameter of the Gamma distribution
x n observations of the Gamma variable
y n observations of the zero-truncated Poisson variable
R n x p design matrix for the Gamma model
S n x q design matrix for the zero-truncated Poisson model
family an integer defining the bivariate copula family: 1 = Gauss, 3 = Clayton, 4=Gumbel, 5=Frank
exposure time for the zero-truncated Poisson model, all entries of the vector have to be > 0. Default is a constant vector of 1.

negative

boolean, if TRUE the negative of the loglikelihood is returned. Default is TRUE.

zt

logical. If zt=TRUE, we use a zero-truncated Poisson variable. Otherwise, we use a Poisson variable. Default is TRUE.

Details

For a Gamma distributed variable $X$ and a (zero truncated) Possion variable $Y$, the loglikelihood is given by

$$
\ell = \sum_{i=1}^{n} \left( f_X(x_i) \left( D_u(F_Y(y_i), F_X(x_i)|\theta) - D_u(F_Y(y_i - 1), F_X(x_i)|\theta) \right) \right).
$$

Here $D_u$ is the h-function of a copula family with copula parameter $\theta$.

Value

loglikelihood

Author(s)

Nicole Kraemer, Daniel Silvestrini

References


See Also

density_joint

Examples

```r
library(VineCopula)
n<-200 # number of examples
R<-5<-cbind(rep(1,n),rnorm(n)) # design matrices with intercept
alpha<-beta<-c(1,1) # regression coefficients
exposure<-rep(1,n) # constant exposure
delta<-0.5 # dispersion parameter
tau<-0.3 # Kendall's tau
family=3 # Clayton copula
theta<-BiCopTau2Par(tau=tau,family=family)
# simulate data
my.data<-simulate_regression_data(n,alpha,beta,R,S,deltau,family=TRUE,exposure)
x<-my.data[,1]
y<-my.data[,2]
#compute loglikelihood for the true coefficients
out<-loglik_joint(alpha,beta,theta,deltau,x,y,R,S,family,exposure)
```
**Description**

Computes the maximum-likelihood estimates for the regression coefficients and the copula parameter.

**Usage**

```r
mle_joint(alpha0, beta0, theta0, delta0, x, y, R, S, family, exposure, sd.error, zt)
```

**Arguments**

- `alpha0`: The starting value of the regression coefficients for the Gamma regression.
- `beta0`: The starting value of the regression coefficients for the (zero-truncated) Poisson regression.
- `theta0`: The starting value of the copula parameter.
- `delta0`: The starting value for the dispersion parameter of the Gamma distribution.
- `x`: `n` observations of the Gamma variable.
- `y`: `n` observations of the zero-truncated Poisson variable.
- `R`: `n x p` design matrix for the Gamma model.
- `S`: `n x q` design matrix for the zero-truncated Poisson model.
- `family`: an integer defining the bivariate copula family: 1 = Gauss, 3 = Clayton, 4 = Gumbel, 5 = Frank.
- `exposure`: exposure time for the zero-truncated Poisson model, all entries of the vector have to be > 0. Default is a constant vector of 1.
- `sd.error`: logical. Should the standard errors of the regression coefficients be returned? Default is FALSE.
- `zt`: logical. If `zt` = TRUE, we use a zero-truncated Poisson variable. Otherwise, we use a Poisson variable. Default is TRUE.

**Details**

This is an internal function called by `copreg`.

**Value**

- `alpha`: estimated coefficients for X, including the intercept.
- `beta`: estimated coefficients for Y, including the intercept.
- `sd.alpha`: estimated standard deviation (if `sd.error = TRUE`).
- `sd.beta`: estimated standard deviation (if `sd.error = TRUE`).
- `sd.g.theta`: estimated standard deviation of `g(θ)` (if `sd.error = TRUE`).
### mle_marginal

#### delta
- estimated dispersion parameter

#### theta
- estimated copula parameter

#### tau
- estimated value of Kendall’s tau

#### family
- copula family

#### ll
- loglikelihood of the estimated model, evaluated at each observation

#### loglik
- overall loglikelihood, i.e. sum of ll

#### Author(s)
- Nicole Kraemer

#### References


#### See Also
- copreg, mle_marginal

#### Examples

```r
# This is an internal function called by copreg()
```

---

#### mle_marginal

**ML-estimates of the marginal models**

#### Description

We fit the Gamma and the (zero-truncated) Poisson model separately.

#### Usage

```r
mle_marginal(x, y, R, S, family, exposure, sd.error=FALSE, zt=TRUE)
```

#### Arguments

- **x**: n observations of the Gamma variable
- **y**: n observations of the (zero-truncated) Poisson variable
- **R**: n x p design matrix for the Gamma model
- **S**: n x q design matrix for the zero-truncated Poisson model
- **family**: an integer defining the bivariate copula family: 1 = Gauss, 3 = Clayton, 4 = Gumbel, 5 = Frank
**exposure**

exposure time for the zero-truncated Poisson model, all entries of the vector have to be $> 0$. Default is a constant vector of 1.

**sd.error**

logical. Should the standard errors of the regression coefficients be returned? Default is FALSE.

**zt**

logical. If zt=TRUE, we use a zero-truncated Poisson variable. Otherwise, we use a Poisson variable. Default is TRUE.

**Details**

This is an internal function called by `copreg`.

**Value**

- **alpha**: estimated coefficients for X, including the intercept
- **beta**: estimated coefficients for Y, including the intercept
- **sd.alpha**: estimated standard deviation (if sd.error=TRUE)
- **sd.beta**: estimated standard deviation (if sd.error=TRUE)
- **delta**: estimated dispersion parameter
- **theta**: 0, in combination with family=1, this corresponds to the independence assumption
- **family**: 1, in combination with theta=0, this corresponds to the independence assumption
- **family0**: copula family as provided in the function call
- **theta.ifm**: estimated copula parameter, estimated via inference from margins
- **tau.ifm**: estimated value of Kendall’s tau, estimated via inference from margins
- **ll**: loglikelihood of the estimated model, assuming independence, evaluated at each observation
- **loglik**: overall loglikelihood, assuming independence, i.e. sum of ll
- **ll.ifm**: loglikelihood of the estimated model, using theta.ifm as the copula parameter, evaluated at each observation
- **loglik.ifm**: overall loglikelihood, using theta.ifm as the copula parameter, i.e. sum of ll.ifm

**Author(s)**

Nicole Kraemer

**References**


**See Also**

copreg, mle_joint
Examples

```r
# This is an internal function called by copreg()
```

---

**pgam**

Distribution of a Gamma variable

---

**Description**

Cumulative distribution function of a Gamma-distributed variable in mean parametrization

**Usage**

```r
pgam(y, mu, delta)
```

**Arguments**

- `y`: vector of length n
- `mu`: mean parameter; either a number or a vector of length n
- `delta`: dispersion parameter; a number

**Value**

cumulative distribution function, evaluated at y, this is a vector of length n

**Author(s)**

Nicole Kraemer

**See Also**

`pztp`, `dgam`

**Examples**

```r
y<-0:10
mu<5
delta<-1
out<-pgam(y, mu, delta)
```
Description

Cumulative distribution function of the policy loss

Usage

ppolicy_loss(l, mu, delta, lambda, theta, family, y.max = 20, zt=TRUE)

Arguments

- l: vector at which the distribution is evaluated
- mu: expectation of the Gamma distribution
- delta: dispersion parameter of the Gamma distribution
- lambda: parameter of the ZTP distribution
- theta: copula parameter
- family: an integer defining the bivariate copula family: 1 = Gauss, 3 = Clayton, 4 = Gumbel, 5 = Frank
- y.max: upper value of the finite sum that we use to approximate the infinite sum in the density, see below for more details
- zt: logical. If zt=TRUE, we use a zero-truncated Poisson variable. Otherwise, we use a Poisson variable. Default is TRUE.

Details

For a Gamma distributed variable X and a (zero truncated) Possion variable Y, the policy loss is defined as \( L = X \cdot Y \). Its density is an infinite sum of weighted Gamma densities. The parameter y.max is the upper value of the finite sum that approximates the infinite sum.

Value

distribution function, evaluated at l

Author(s)

Nicole Kraemer

References

See Also

epolicy_loss, qpolicy_loss, dpolicy_loss

Examples

```
library(VineCopula)
mu<-1000
delta<-0.09
lambda<-2.5
family<-1
theta<-BiCopTau2Par(tau=0.5,family=family)
l<-seq(1,7000,length=100)
out<-ppolicy_loss(l,mu,delta,lambda,theta,family)
plot(l,out,type="l",lwd=3,xlab="loss",ylab="density")
```

predict.copreg  Prediction of the copula regression model

Description

This function predicts the outcome of a copula regression model for new data.

Usage

```
## S3 method for class 'copreg'
predict(object,Rtest,Stest,exposure=rep(1,nrow(Stest)),independence=FALSE,...)
```

Arguments

- `object` copreg object returned from copreg
- `Rtest` design matrix of the new data for the Gamma model
- `Stest` design matrix of the new data for the zero truncated Poisson model
- `exposure` exposure time for the zero-truncated Poisson model, all entries of the vector have to be > 0. Default is a constant vector of 1.
- `independence` logical. Do we assume that the two variables are independent. Default is FALSE.
- `...` other parameters

Details

For new data that is defined by the design matrices Rtest and Stest, and the exposure time exposure, the function predicts the values x of the Gamma variable, the values y of the (zero truncated) Poisson variable, and the policy loss. If independence=TRUE, the function predicts the policy loss under the assumption that X and Y are independent.
pztp

Value

- x.ped: predicted value of x
- y.ped: predicted value of y
- l.ped: predicted value of the policy loss

Author(s)

Nicole Kraemer

References


See Also
copreg, simulate_regression_data

Examples

```r
n<-200 # number of examples
R<-S<-cbind(rep(1,n), rnorm(n)) # design matrices with intercept
alpha<-beta<-c(1,-1) # regression coefficients
exposure<-rep(1,n) # constant exposure
delta<-0.5 # dispersion parameter
tau<-0.3 # Kendall's tau
family=3 # Clayton copula

# simulate data
my.data<-simulate_regression_data(n, alpha, beta, R, S, delta, tau, family, TRUE, exposure)
x<-my.data[,1]
y<-my.data[,2]

# joint model without standard errors
my.model<-copreg(x,y,R,S,family,exposure, FALSE, TRUE)

# fitted values

## Not run: out<-predict(my.model,R,S)
```

pztp

Cumulative distribution function of a zero truncated Poisson variable

Description

Cumulative distribution function of a zero truncated Poisson variable

Usage

pztp(y, lambda)
Arguments

- `y` vector of length n
- `lambda` either a number or a vector of length n

Value

cumulative distribution function, evaluated at y, this is a vector of length n

Author(s)

Nicole Kraemer

See Also

dztp

Examples

```r
y <- 0:10
lambda1 <- 5  # fixed lambda
out1 <- qztp(y, lambda1)
lambda2 <- 2:12  # variable lambda
out2 <- qztp(y, lambda2)
```

---

**qpolicy_loss**  
Quantile of the policy loss

Description

Quantile of the policy loss

Usage

```r
qpolicy_loss(q, mu, delta, lambda, theta, family, y.max = 20, zt = TRUE)
```

Arguments

- `q` value at which the quantile function is evaluated
- `mu` expectation of the Gamma distribution
- `delta` dispersion parameter of the Gamma distribution
- `lambda` parameter of the (zero-truncated) Poisson distribution
- `theta` copula parameter
- `family` an integer defining the bivariate copula family: 1 = Gauss, 3 = Clayton, 4 = Gumbel, 5 = Frank
- `y.max` upper value of the finite sum that we use to approximate the infinite sum in the density, see below for more details
- `zt` logical. If `zt = TRUE`, we use a zero-truncated Poisson variable. Otherwise, we use a Poisson variable. Default is TRUE.
Details

For a Gamma distributed variable \(X\) and a (zero truncated) Possion variable \(Y\), the policy loss is defined as \(L = X \cdot Y\). Its density is an infinite sum of weighted Gamma densities. The parameter \(y_{\text{max}}\) is the upper value of the finite sum that approximates the infinite sum.

Value

quantile, evaluated at \(q\)

Author(s)

Nicole Kraemer

References


See Also

\(\text{ppolicy_loss, epolicy_loss, dpolicy_loss}\)

Examples

```r
library(VineCopula)
mu<-1000
delta<-0.09
lambda<-2.5
family<-1
theta<-BiCopTau2Par(tau=0.5,family=family)
# upper quartile
out<-qpolicy_loss(0.75,mu,delta,lambda,theta,family)
```

rgam

Samples from a Gamma variable

Description

Samples from a Gamma-distributed variable in mean parametrization

Usage

\(\text{rgam}(n, \mu, \delta)\)
Arguments

- **n**: number of samples
- **mu**: mean parameter; either a number or a vector of length n
- **delta**: dispersion parameter; a number

Value

- **n** samples

Author(s)

Nicole Kraemer

See Also

pgam, dgam

Examples

```r
n=20
mu<-5
delta<-1
out<-rgam(n,mu,delta)
```

Description

Simulation from the joint model

Usage

```r
simulate_joint(n, mu, delta, lambda, theta, family, max.y = 5000, eps = 1e-05,zt=TRUE)
```

Arguments

- **n**: number of samples
- **mu**: expectation of the Gamma distribution
- **delta**: dispersion parameter of the Gamma distribution
- **lambda**: parameter of the (zero-truncated) Poisson distribution
- **theta**: copula parameter
- **family**: an integer defining the bivariate copula family: 1 = Gauss, 3 = Clayton, 4=Gumbel, 5=Frank
- **max.y**: upper value for the conditional (zero truncated) Poisson variable, see below for more details
simulate_joint

eps precision, see below for more details
zt logical. If zt=TRUE, we use a zero-truncated Poisson variable. Otherwise, we use a Poisson variable. Default is TRUE.

Details

For a Gamma distributed variable X and a (zero truncated) Possion variable Y, we sample from their joint distribution that is given by the density function

\[ f_{XY}(x,y) = f_X(x) (D_u(F_Y(y), F_X(x)|\theta) - D_u(F_Y(y-1), F_X(x)|\theta)) . \]

Here \( D_u \) is the h-function of a copula famila family with copula parameter \( \theta \). First, we sample \( n \) observations \( x \) from the marginal Gamma distribution. Second, for each \( x \), we then sample an observation from the conditional distribution of \( Y \) given \( X=x \). In the second step, the conditional distribution is evaluated up to the maximum of \( \max y \) and the smallest integer > \( y, \max \) for which the conditional probability is smaller than eps.

Value

\( n \) samples, stored in a \( n \times 2 \) matrix

Author(s)

Nicole Kraemer

References


See Also

density_joint, simulate_regression_data, density_conditional

Examples

library(VineCopula)
n<-100 # number of observations
mu<-1000
delta<-0.09
lambda<-2.5
family<-1
theta<-BiCopTau2Par(tau=0.5,family=family)
my.data<-simulate_joint(n,mu,delta,lambda,theta,family)
**simulate_regression_data**

*Simulate regression data*

**Description**

Simulate regression data

**Usage**

`simulate_regression_data(n, alpha, beta, R, S, delta, tau, family, zt, exposure)`

**Arguments**

- `n`: number of samples
- `alpha`: coefficients for the Gamma regression
- `beta`: coefficients for the (zero-truncated) Poisson regression
- `R`: n x p design matrix for the Gamma model
- `S`: n x q design matrix for the (zero-truncated) Poisson model
- `delta`: dispersion parameter of the Gamma distribution
- `tau`: Kendalls tau
- `family`: an integer defining the bivariate copula family: 1 = Gauss, 3 = Clayton, 4=Gumbel, 5=Frank
- `exposure`: exposure time for the zero-truncated Poisson model, all entries of the vector have to be > 0. Default is a constant vector of 1.
- `zt`: logical. If `zt=True`, we use a zero-truncated Poisson variable. Otherwise, we use a Poisson variable. Default is `TRUE`.

**Details**

We consider positive continuous random variables $X_i$ and positive or non-negative count variables $Y_i$. We model $X_i$ in terms of a covariate vector $r_i$ and $Y_i$ in terms of a covariate vector $s_i$. The marginal regression models are specified via

$$X_i \sim \text{Gamma}(\mu_i, \delta)$$

with $\ln(\mu_i) = r_i^T \alpha$ for the continuous variable. For the count variable, if `zt=True`, we use a zero-truncated Poisson model,

$$Y_i \sim \text{ZTP}(\lambda_i)$$

with $\ln(\lambda_i) = \ln(e_i) + s_i^T \beta$. Otherwise, we use a Poisson model. $e_i$ denotes the exposure time.

Further, we assume that the dependency of $X_i$ and $Y_i$ is modeled in terms of a copula family with parameter $\theta$. 
theta2z

Value

n samples from the joint regression model

Author(s)

Nicole Kraemer

References


See Also

simulate_joint

Examples

n<-200 # number of examples
R<-Scbind(rep(1,n),rnorm(n)) # design matrices with intercept
alpha<-beta<-c(1,-1) # regression coefficients
exposure<-rep(1,n) # constant exposure
delta<-0.5 # dispersion parameter
tau<-0.3 # Kendall's tau
family=3 # Clayton copula
# simulate data
my.data<-simulate_regression_data(n,alpha,beta,R,S,delta,tau,family,TRUE,exposure)
x<-my.data[,1]
y<-my.data[,2]

theta2z

Transformation of the copula parameter

Description

This function transforms the copula parameter theta such that theta2z(theta) is unrestricted.

Usage

theta2z(theta, family)

Arguments

theta copula parameter
family an integer defining the bivariate copula family: 1 = Gauss, 3 = Clayton, 4 = Gumbel, 5 = Frank
Details

This function transforms the parameter of a copula such that the transformed value is unconstrained. E.g., for the Gauss copula (family=1), the parameter $\theta$ lies in the interval $]-1, 1[$, and the transformation is defined as

$$\theta \mapsto \frac{1}{2} \ln \left( \frac{1 + \theta}{1 - \theta} \right)$$

Value

transformation

Author(s)

Nicole Kraemer

See Also

z2theta

Examples

```r
#### this is an internal function ####
```
Details

Let us denote by $\ell^{(1)}$, $\ell^{(2)}$ the vectors of pointwise loglikelihoods for a model with copula family 1 and 2 respectively. Here, we assume that both models have the same degrees of freedom, i.e. the same number of parameters. We now compute the differences of the pointwise loglikelihood as

\[ m_i := \ell_i^{(1)} - \ell_i^{(2)}, \quad i = 1, \ldots, n. \]

Denote by

\[ m = \frac{1}{n} \sum_{i=1}^{n} m_i \]

the mean of the differences. The test statistic

\[ T_V := \frac{\sqrt{n} \cdot m}{\sqrt{\sum_{i=1}^{n} (m_i - m)^2}}, \]

is asymptotically normally distributed with zero mean and unit variance. Hence, we prefer copula family 1 to copula family 2 at level $\alpha$ if

\[ T_V > \Phi^{-1} \left( 1 - \frac{\alpha}{2} \right), \]

where $\Phi$ denotes the standard normal distribution function. If

\[ T_V < \Phi^{-1} \left( \frac{\alpha}{2} \right), \]

we prefer copula family 2. Otherwise, no decision among the two copula families is possible. If the models contain different numbers of estimated parameters, the test statistic is corrected using either the AIC or BIC criterion.

Value

value of the test statistic

Author(s)

Nicole Kraemer

References


See Also

copreg
Examples

```r
n<-200 # number of examples
R<-cbind(rep(1,n),rnorm(n)) # design matrices with intercept
alpha<-beta<-c(1,-1) # regression coefficients
exposure<-rep(1,n) # constant exposure
delta<-0.5 # dispersion parameter
family=3 # Clayton copula
# simulate data
my.data<-simulate_regression_data(n,alpha,beta,R,S,delta,tau,family,TRUE,exposure)
x<-my.data[,1]
y<-my.data[,2]

# joint model without standard errors
my.model.clayton<-copreg(x,y,R,S,family=3,exposure,FALSE,TRUE)
my.model.gauss<-copreg(x,y,R,S,family=1,exposure,FALSE,TRUE)
# vuongtest(my.model.clayton,my.model.gauss)
```

---

### z2theta

**Inverse of the parameter transformation**

**Description**

This is the inverse of the function `theta2z`

**Usage**

```r
z2theta(z, family)
```

**Arguments**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>z</code></td>
<td>value to be transformed</td>
</tr>
<tr>
<td><code>family</code></td>
<td>an integer defining the bivariate copula family: 1 = Gauss, 3 = Clayton, 4 = Gumbel, 5 = Frank</td>
</tr>
</tbody>
</table>

**Details**

The function `theta2z` transforms the parameter of a copula such that the transformed value in unconstrained. This function is its inverse.

**Value**

transformed value

**Author(s)**

Nicole Kraemer
**ztp.glm**

See Also

theta2z

Examples

```r
#--- this is an internal function ---
```

---

**ztp.glm**

*GLM for a zero truncated Poisson variable*

---

**Description**

Zero truncated generalized linear model.

**Usage**

```r
ztp.glm(y, S, exposure = rep(1, length(y)), sd.error=FALSE)
```

**Arguments**

- `y` vector of response values
- `S` design matrix
- `exposure` exposure time for the zero-truncated Poisson model, all entries of the vector have to be \( > 0 \). Default is a constant vector of 1.
- `sd.error` logical. Should the standard errors of the regression coefficients be returned? Default is FALSE.

**Details**

We consider positive count variables \( Y_i \). We model \( Y_i \) in terms of a covariate vector \( s_i \). The generalized linear model is specified via

\[
Y_i \sim \text{ZTP}(\lambda_i)
\]

with \( \ln(\lambda_i) = \ln(e_i) + s_i^T \beta \). Here \( e_i \) denotes the exposure time.

**Value**

- `coefficients` estimated regression coefficients
- `sd` estimated standard error, if `sd.error`=TRUE

**Author(s)**

Nicole Kraemer
See Also

mle_marginal, mle_joint

Examples

```r
n <- 200  # number of examples
R <- cbind(rep(1, n), rnorm(n))  # design matrices with intercept
alpha <- beta <- c(1, -1)  # regression coefficients
exposure <- rep(1, n)  # constant exposure
delta <- 0.5  # dispersion parameter
tau <- 0.3  # Kendall's tau
family <- 3  # Clayton copula
# simulate data
my.data <- simulate_regression_data(n, alpha, beta, R, S, delta, tau, family, TRUE, exposure)
x <- my.data[, 1]
y <- my.data[, 2]
# fit marginal ZTP-model with standard errors
my.model <- ztp.glm(y, S, exposure = exposure, TRUE)
```
Index

*Topic **datagen**
  rgam, 23
  simulate_joint, 24
  simulate_regression_data, 26

*Topic **distribution**
  D_u, 11
  density_conditional, 5
  density_joint, 6
  dgam, 8
  dpolicy_loss, 9
  dztp, 10
  epolicy_loss, 12
  loglik_joint, 13
  pgam, 18
  ppolicy_loss, 19
  pztp, 21
  qpolicy_loss, 22
  rgam, 23

*Topic **htest**
  copreg, 3
  mle_joint, 15
  mle_marginal, 16

*Topic **math**
  theta2z, 27
  z2theta, 30

*Topic **models**
  copreg, 3
  density_conditional, 5
  epolicy_loss, 12
  loglik_joint, 13
  mle_joint, 15
  mle_marginal, 16
  predict.copreg, 20
  vuongtest, 28

*Topic **multivariate**
  copreg, 3
  density_joint, 6
  loglik_joint, 13
  mle_joint, 15
  mle_marginal, 16
  ztp.glm, 31

*Topic **package**
  CopulaRegression-package, 2
  copreg, 3, 15–17, 21, 29
  CopulaRegression
    (CopulaRegression-package), 2
  CopulaRegression-package, 2
  D_u, 7, 11
  density_conditional, 5, 7, 25
  density_joint, 6, 6, 11, 14, 25
  dgam, 8, 18, 24
  dpolicy_loss, 9, 13, 20, 23
  dztp, 10, 22
  epolicy_loss, 10, 12, 20, 23
  loglik_joint, 13
  mle_joint, 5, 15, 17, 32
  mle_marginal, 5, 16, 16, 32
  pgam, 8, 18, 24
  ppolicy_loss, 19, 23
  predict (predict.copreg), 20
  predict.copreg, 20
  pztp, 10, 18, 21
  qpolicy_loss, 10, 20, 22
  rgam, 23
  simulate_joint, 24, 27
  simulate_regression_data, 5, 21, 25, 26
  theta2z, 27, 30, 31
  vuongtest, 28
  z2theta, 28, 30
  ztp.glm, 31