Package 'rregm'

November 1, 2026

Type Package

Title Reparameterized Regression Models

Version 1.5

Maintainer Diego Gallardo <dgallardo@ubiobio.cl>

Description Provides estimation and data generation tools for several new regression models, including the gamma, beta, inverse gamma, beta prime, log-normal and log-logistic distributions. These models can be parameterized based on the mean, median, mode, geometric mean and harmonic mean, except for the log-logistic model which is based on alternative parametrizations.
For details, see Bourguignon and Gallardo (2025a) <doi:10.1016/j.chemolab.2025.105382> and Bourguignon and Gallardo (2025b) <doi:10.1111/stan.70007>.
The package also implements higher-order likelihood inference through Skovgaard-adjusted likelihood ratio statistics and predictive shrinkage estimators reparameterized beta regression models.

Depends R (>= 4.0.0), stats

Imports extraDistr, pracma, gamlss, gamlss.dist, invgamma, skewMLRM

License GPL (>= 2)

Encoding UTF-8

NeedsCompilation no

Author Diego Gallardo [aut, cre],
Marcelo Bourguignon [aut],
Marcia Brandao [aut],
Tiago Magalhaes [ctb],
Rafael Izbicki [ctb],
Yolanda Gomez [ctb]

Repository CRAN

Date/Publication 2026-11-01 22:30:02 UTC

Contents

RBE 2
RBE.skov 5

RBP 7
RGA 11
RIG 14
RLL 18
RLN 22
tools.rregm 25

Index **27**

RBE *Tools for a reparameterized beta regression model*

Description

A set of functions related to the reparameterized beta regression model based on different measures of central tendency: mean, median, mode, geometric mean or harmonic mean.

Usage

```
BEAM(mu.link = "logit", sigma.link = "log")
BEGM(mu.link = "logit", sigma.link = "log")
BEHM(mu.link = "logit", sigma.link = "log")
BEMD(mu.link = "logit", sigma.link = "log")
BEMO(mu.link = "logit", sigma.link = "log")
dBEAM(x, mu = 0.5, sigma = 1, log = FALSE)
dBEGM(x, mu = 0.5, sigma = 1, log = FALSE)
dBEHM(x, mu = 0.5, sigma = 1, log = FALSE)
dBEMD(x, mu = 0.5, sigma = 1, log = FALSE)
dBEMO(x, mu = 0.5, sigma = 1, log = FALSE)
dRBE(x, mu=0.5, sigma=1, param="AM", log=FALSE)
fit.RBE(formula = formula(data), sigma.formula=~1, data, param="AM")
pBEAM(q, mu = 0.5, sigma = 1, lower.tail = TRUE, log.p = FALSE)
pBEGM(q, mu = 0.5, sigma = 1, lower.tail = TRUE, log.p = FALSE)
pBEHM(q, mu = 0.5, sigma = 1, lower.tail = TRUE, log.p = FALSE)
pBEMD(q, mu = 0.5, sigma = 1, lower.tail = TRUE, log.p = FALSE)
pBEMO(q, mu = 0.5, sigma = 1, lower.tail = TRUE, log.p = FALSE)
pRBE(q, mu=0.5, sigma=1, param="AM", lower.tail = TRUE, log.p = FALSE)
qBEAM(p, mu = 0.5, sigma = 1, lower.tail = TRUE, log.p = FALSE)
qBEGM(p, mu = 0.5, sigma = 1, lower.tail = TRUE, log.p = FALSE)
qBEHM(p, mu = 0.5, sigma = 1, lower.tail = TRUE, log.p = FALSE)
qBEMD(p, mu = 0.5, sigma = 1, lower.tail = TRUE, log.p = FALSE)
qBEMO(p, mu = 0.5, sigma = 1, lower.tail = TRUE, log.p = FALSE)
qRBE(p, mu=0.5, sigma=1, param="AM", lower.tail = TRUE, log.p = FALSE)
rBEAM(n, mu = 0.5, sigma = 1)
rBEGM(n, mu = 0.5, sigma = 1)
rBEHM(n, mu = 0.5, sigma = 1)
rBEMD(n, mu = 0.5, sigma = 1)
rBEMO(n, mu = 0.5, sigma = 1)
rRBE(n, mu=0.5, sigma=1, param="AM")
```

Arguments

mu.link	the mu link function with default logit
sigma.link	the sigma link function with default log
mu, sigma	vector of parameter values
formula	an object of class "*formula*" (or one that can be coerced to that class): a symbolic description of the model to be fitted. The details of model specification are given under 'Details'.
data	an optional data frame, list or environment (or object coercible by *as.data.frame* to a data frame) containing the variables in the model. If not found in data, the variables are taken from environment(formula), typically the environment from which lm is called.
sigma.formula	a formula object for fitting a model to the sigma parameter, as in the formula above, e.g. sigma.formula=~x1+x2.
param	parameterization used for the model. "AM" for mean, "MD" for median, "MO" for mode, "GM" for geometric mean, and "HM" for harmonic mean.
x, q	vector of quantiles
p	vector of probabilities
n	number of observations. If length$(n) > 1$, the length is taken to be the number required.
log, log.p	logical; if TRUE, probabilities p are given as log(p).
lower.tail	logical; if TRUE, probabilities are $P(X \leq x)$ otherwise, $P(X > x)$.

Details

The parameterization for the reparameterized beta distribution is given by

$$f(x;\mu,\sigma) = \frac{x^{\mu\,\sigma+\tau_1-1}(1-x)^{(1-\mu)\sigma+\tau_2-\tau_1-1}}{B(\mu\,\sigma+\tau_1,(1-\mu)\sigma+\tau_2-\tau_1)}, \quad 0<x<1,$$

where $0<\mu<1$, $\sigma>0$ and τ_1 and τ_2 are constant. The following cases are highlighted:

- param="AM": $\tau_1=\tau_2=0$ and μ represents the mean of the distribution.

- param="GM": $\tau_1=\tau_2=1/2$ and μ represents the geometric mean of the distribution.

- param="HM": $\tau_1=\tau_2=1$ and μ represents the harmonic mean of the distribution.

- param="MO": $\tau_1=1$ and $\tau_2=2$ and μ represents the mode of the distribution.

- param="MD": $\tau_1=1/2$ and $\tau_2=0$ and μ represents the median of the distribution.

Suppose the central tendency and the concentration parameter of Y_i satisfies the following functional relations

$$\text{logit}(\mu_i) = \mathbf{x}_i^\top \boldsymbol{\xi} \quad \text{and} \quad \log(\sigma_i) = \eta_{2i} = \mathbf{z}_i^\top \boldsymbol{\nu},$$

where $\text{logit}(u) = \log(u/(1-u))$ is the logit function, $\boldsymbol{\xi} = (\xi_1,\ldots,\xi_p)^\top$ and $\boldsymbol{\nu} = (\nu_1,\ldots,\nu_q)^\top$ are vectors of unknown regression coefficients which are assumed to be functionally independent, $\boldsymbol{\xi} \in \mathbb{R}^p$ and $\boldsymbol{\nu} \in \mathbb{R}^q$, with $p+q<n$, and $\mathbf{x}_i = (x_{i1},\ldots,x_{ip})^\top$ and $\mathbf{z}_i = (z_{i1},\ldots,z_{iq})^\top$ are observations on p and q known regressors, for $i=1,\ldots,n$. Furthermore, we assume that the covariate matrices $\mathbf{X} = (\mathbf{x}_1,\ldots,\mathbf{x}_n)^\top$ and $\mathbf{Z} = (\mathbf{z}_1,\ldots,\mathbf{z}_n)^\top$ have rank p and q, respectively.

For this model, the Pearson's residuals are given by

$$r_i = \frac{y_i - m_i}{s_i}, \quad i = 1, \ldots, n,$$

where

$$m_i = \frac{\mu_i\sigma_i + \tau_1}{\sigma_i + \tau_2} \quad \text{and} \quad s_i = \sqrt{\frac{(\mu_i\sigma_i + \tau_1)((1-\mu_i)\sigma_i + \tau_2 - \tau_1)}{(\sigma_i + \tau_2)^2(\sigma_i + \tau_2 + 1)}}.$$

whereas the modified Pearson's residuals are given by

$$r_i^* = \frac{\text{logit}(y_i) - m_i^*}{s_i^*}, \quad i = 1, \ldots, n,$$

where

$$m_i^* = \psi(\mu_i\sigma_i+\tau_1)-\psi((1-\mu_i)\sigma_i+\tau_2-\tau_1) \quad \text{and} \quad s_i^* = \sqrt{\psi'(\mu_i\sigma_i + \tau_1) + \psi'((1-\mu_i)\sigma_i + \tau_2 - \tau_1)},$$

with $\psi(\cdot)$ and $\psi'(\cdot)$ denoting the digamma and trigamma functions, respectively. Finally, the quantile residuals are given by

$$r_i^q = \Phi^{-1}\left(I_{y_i}(\mu_i\sigma_i + \tau_1, (1-\mu_i)\sigma_i + \tau_2 - \tau_1)\right), \quad i = 1, \ldots, n,$$

where $\Phi^{-1}(\cdot)$ denotes the inverse of the cumulative distribution function for the standard normal model and $I_y(\alpha, \beta) = B_x(\alpha, \beta)/B(\alpha, \beta)$ is the incomplete beta function ratio, $B_x(\alpha, \beta) = \int_0^x \omega^{\alpha-1}(1-\omega)^{\beta-1}\text{d}\omega$ is the incomplete beta function, $B(\alpha, \beta) = \Gamma(\alpha)\Gamma(\beta)/\Gamma(\alpha+\beta)$ is the beta function and $\Gamma(\alpha) = \int_0^\infty \omega^{\alpha-1}\text{e}^{-\omega}\text{d}\omega$ is the gamma function. dRBE gives the density, pRBE gives the distribution function, qRBE gives the quantile function, and rRBE generates random deviates from the beta distribution with the specified parameterization. In addition, dBEXX, pBEXX, qBEXX and rBEXX also provide the equivalent functions for a specified parameterization for XX: AM (mean), GM (geometric mean), HM (harmonic mean), MD (median) and MO (mode). For instance, dBEAM gives the density for the beta model parameterized in the mean, pBEGM gives the distribution function for the beta model parameterized in the geometric mean and so on. Finally, the functions BEAM, BEGM, BEHM, BEMD and BEMO also provide a framework to fit models with *gamlss*.

Value

an object of class "rregm" is returned. The object returned for these functions is a list containing the following components:

`estimate`	A matrix containing the estimates and standard errors.
`logLik`	the log-likelihood function evaluated at the corresponding estimators.
`AIC`	the Akaike information criterion.
`BIC`	the Bayesian information criterion.
`tau1`, `tau2`	values for tau1 and tau2, depending on the considered parameterization.
`pearson.res`	Pearson's residuals.
`mod.pearson.res`	modified Pearson's residuals.

quant.res	quantile residuals.
convergence	logical. If convergence was attained.
dist	BE (the beta distribution).
param	The specified parameterization.
mu.x	design matrix for mu.
sigma.x	design matrix for sigma.

Author(s)

Diego Gallardo and Marcelo Bourguignon.

References

Bourguignon, M., Gallardo, D.I. (2025) A general and unified parameterization of the beta distribution: A flexible and robust beta regression model. Statistica Neerlandica, 79(2), e70007.

See Also

fit.RBE, fit.RGA, fit.RBP, fit.RIG, fit.RLL

Examples

```
set.seed(2100)
n=100; x1=rnorm(max(n)) ##drawing covariates, the same for mu and sigma
mu=plogis(0.5-0.4*x1); sigma=exp(-0.1+0.05*x1)
y=rRBE(n, mu, sigma, param="MD") ## model parameterized in the median
data=list(y=y, x1=x1)
aux.RBE=fit.RBE(y~x1, sigma.formula=~x1, data=data, param="MD")
summary(aux.RBE)
qqnorm(res(aux.RBE, type="mod.pearson"))
#The beta model parameterized in the median also can be fitted using gamlss
#gamlss(y~x1, sigma.formula=~x1, data=data, family=BEMD)
```

RBE.skov *Skovgaard's adjustment and predictive measures for a reparameterized beta regression model*

Description

A set of functions related to Skovgaard's adjustment for the reparameterized beta regression model based on different measures of central tendency: mean, median, mode, geometric mean or harmonic mean.

Usage

```
RBE.skovgaard(formula, data, reduced.fo=~1, phi.test=NULL, param="AM")
RBE.predictive(formula, data, reduced.fo=~1, phi.test=NULL, train=0.7, param="AM")
```

Arguments

formula	an object of class "*formula*" (or one that can be coerced to that class): a symbolic description of the model to be fitted. The details of model specification are given under 'Details'.
data	an optional data frame, list or environment (or object coercible by *as.data.frame* to a data frame) containing the variables in the model. If not found in data, the variables are taken from environment(formula), typically the environment from which lm is called.
reduced.fo	a formula object for fitting the reduced model for μ, reduced.fo=~1 by default.
param	parameterization used for the model. "AM" for mean, "MD" for median, "MO" for mode, "GM" for geometric mean, and "HM" for harmonic mean.
phi.test	value to include for ϕ in the null hypothesis. phi.test=NULL represents no hypothesis for ϕ.
train	train proportion of observations used for model training. (0.7 by default)

Details

The function `RBE.skovgaard()` performs likelihood ratio inference in reparameterized beta (RB) regression models by computing the standard likelihood ratio statistic together with two adjusted versions based on Skovgaard's higher-order asymptotic approximation. These adjusted statistics are designed to improve the finite-sample performance of likelihood ratio tests by reducing the size distortions commonly observed when the sample size is small or moderate.

The user specifies a full model through `formula` and a reduced model through `reduced.fo`. The function then fits both models and returns the standard likelihood ratio statistic (`SLR`) as well as the two adjusted statistics (`SLR1` and `SLR2`), together with their corresponding chi-squared p-values.

The function `RBE.predictive()` implements likelihood-based shrinkage estimators for RB regression models. The data are randomly divided into training and test sets according to the proportion specified in `train`. Model fitting is performed using the training sample and predictive performance is evaluated on the test sample.

Five competing estimators are considered:

- `MLE`: maximum likelihood estimator;
- `US`: classical uniform shrinkage estimator;
- `US1`: shrinkage estimator based on the first Skovgaard-adjusted likelihood ratio statistic;
- `US2`: shrinkage estimator based on the second Skovgaard-adjusted likelihood ratio statistic;
- `GAIC`: variable selection based on the generalized Akaike information criterion.

Predictive performance is summarized through the mean squared error (MSE), mean absolute error (MAE), and logarithmic score (LS). Lower values of MSE and MAE indicate better predictive accuracy, whereas larger values of LS indicate better probabilistic predictions.

Both functions support all currently available parameterizations of the RB distribution implemented in **rregm**, namely the arithmetic mean (`"AM"`), geometric mean (`"GM"`), harmonic mean (`"HM"`), mode (`"MO"`), and median (`"MD"`) parameterizations.

The methods implemented in these functions are based on higher-order likelihood theory and likelihood-based shrinkage methodology, providing tools for both improved finite-sample inference and enhanced predictive performance in RB regression models.

Value

Both functions return an object of class "LRskov".

For RBE.skovgaard(), the returned object contains likelihood ratio statistics and associated p-values.

For RBE.predictive(), the returned object contains out-of-sample prediction measures (MSE, MAE, and LS) for the competing estimators, together with information on the training/test split and fitted model specification.

Author(s)

Diego Gallardo, Tiago Magalhaes and Rafael Izbicki.

References

Bourguignon, M., Gallardo, D.I. (2025) A general and unified parameterization of the beta distribution: A flexible and robust beta regression model. Statistica Neerlandica, 79(2), e70007.

Examples

```
set.seed(123)
n <- 100
x1 <- runif(n)
x2 <- runif(n)
x3 <- runif(n)
mu <- plogis(-1 + 2*x1 - x2)
y <- rBEGM(n, mu = mu, sigma = 15)
dat <- data.frame(y, x1, x2, x3)
## Adjusted LR test
RBE.skovgaard(y ~ x1 + x2 + x3, data = dat, reduced.fo = ~ x1 + x2, param = "GM")

## Predictive shrinkage
set.seed(123)
RBE.predictive(y ~ x1 + x2 + x3, data = dat, train = 0.8, param = "GM")
```

RBP *Tools for a reparameterized beta prime regression model*

Description

A set of functions related to the reparameterized beta prime regression model based on different measures of central tendency: mean, median, mode, geometric mean or harmonic mean.

Usage

```
BPAM(mu.link = "log", sigma.link = "log")
BPGM(mu.link = "log", sigma.link = "log")
BPHM(mu.link = "log", sigma.link = "log")
BPMD(mu.link = "log", sigma.link = "log")
BPMO(mu.link = "log", sigma.link = "log")
dBPAM(x, mu = 1, sigma = 1.5, log = FALSE)
dBPGM(x, mu = 1, sigma = 1.5, log = FALSE)
dBPHM(x, mu = 1, sigma = 1.5, log = FALSE)
dBPMD(x, mu = 1, sigma = 1.5, log = FALSE)
dBPMO(x, mu = 1, sigma = 1.5, log = FALSE)
dRBP(x, mu=1, sigma=1.5, param="AM", log=FALSE)
fit.RBP(formula = formula(data), sigma.formula=~1, data, param="AM")
pBPAM(q, mu = 1, sigma = 1.5, lower.tail = TRUE, log.p = FALSE)
pBPGM(q, mu = 1, sigma = 1.5, lower.tail = TRUE, log.p = FALSE)
pBPHM(q, mu = 1, sigma = 1.5, lower.tail = TRUE, log.p = FALSE)
pBPMD(q, mu = 1, sigma = 1.5, lower.tail = TRUE, log.p = FALSE)
pBPMO(q, mu = 1, sigma = 1.5, lower.tail = TRUE, log.p = FALSE)
pRBP(q, mu = 1, sigma=1.5, param="AM", lower.tail = TRUE, log.p = FALSE)
qBPAM(p, mu = 1, sigma = 1.5, lower.tail = TRUE, log.p = FALSE)
qBPGM(p, mu = 1, sigma = 1.5, lower.tail = TRUE, log.p = FALSE)
qBPHM(p, mu = 1, sigma = 1.5, lower.tail = TRUE, log.p = FALSE)
qBPMD(p, mu = 1, sigma = 1.5, lower.tail = TRUE, log.p = FALSE)
qBPMO(p, mu = 1, sigma = 1.5, lower.tail = TRUE, log.p = FALSE)
qRBP(p, mu = 1, sigma=1.5, param="AM", lower.tail = TRUE, log.p = FALSE)
rBPAM(n, mu = 1, sigma = 1.5)
rBPGM(n, mu = 1, sigma = 1.5)
rBPHM(n, mu = 1, sigma = 1.5)
rBPMD(n, mu = 1, sigma = 1.5)
rBPMO(n, mu = 1, sigma = 1.5)
rRBP(n, mu = 1, sigma=1.5, param="AM")
```

Arguments

mu.link	the mu link function with default log
sigma.link	the sigma link function with default log
mu, sigma	vector of parameter values
formula	an object of class "*formula*" (or one that can be coerced to that class): a symbolic description of the model to be fitted. The details of model specification are given under 'Details'.
data	an optional data frame, list or environment (or object coercible by *as.data.frame* to a data frame) containing the variables in the model. If not found in data, the variables are taken from environment(formula), typically the environment from which lm is called.
sigma.formula	a formula object for fitting a model to the sigma parameter, as in the formula above, e.g. sigma.formula=~x1+x2.

`param`	parameterization used for the model. "AM" for mean, "MD" for median, "MO" for mode, "GM" for geometric mean, and "HM" for harmonic mean.
`x`, `q`	vector of quantiles
`p`	vector of probabilities
`n`	number of observations. If length$(n) > 1$, the length is taken to be the number required.
`log`, `log.p`	logical; if TRUE, probabilities p are given as log(p).
`lower.tail`	logical; if TRUE, probabilities are $P(X \leq x)$ otherwise, $P(X > x)$.

Details

The parameterization for the reparameterized beta prime distribution is given by

$$f(x;\mu,\sigma) = \frac{x^{\mu\,\sigma+\tau_1-1}(1+x)^{-(\sigma(1+\mu)+\tau_1-\tau_2+1)}}{B(\mu\,\sigma+\tau_1,\sigma-\tau_2+1)}, \quad x>0,$$

where $\delta = \delta(\sigma) = (\sqrt{\sigma(\sigma+4)}+\sigma)/2$, $0<\mu<1$, $\sigma>0$ and τ is a constant. The following cases are highlighted:

- param="AM": $\tau = 0$ and μ represents the mean of the distribution.

- param="GM": $\tau = 1/2$ and μ represents the geometric mean of the distribution.

- param="MD": $\tau = 1/3$ and μ represents the median of the distribution.

- param="MO" or ="HM": $\tau = 1$ and μ represents the mode or the harmonic mean of the distribution.

Suppose the central tendency and the concentration parameter of Y_i satisfies the following functional relations

$$\log(\mu_i) = \mathbf{x}_i^\top \boldsymbol{\xi} \quad \text{and} \quad \log(\sigma_i) = \eta_{2i} = \mathbf{z}_i^\top \boldsymbol{\nu},$$

where $\boldsymbol{\xi} = (\xi_1,\ldots,\xi_p)^\top$ and $\boldsymbol{\nu} = (\nu_1,\ldots,\nu_q)^\top$ are vectors of unknown regression coefficients which are assumed to be functionally independent, $\boldsymbol{\xi} \in \mathbb{R}^p$ and $\boldsymbol{\nu} \in \mathbb{R}^q$, with $p+q<n$, and $\mathbf{x}_i = (x_{i1},\ldots,x_{ip})^\top$ and $\mathbf{z}_i = (z_{i1},\ldots,z_{iq})^\top$ are observations on p and q known regressors, for $i=1,\ldots,n$. Furthermore, we assume that the covariate matrices $\mathbf{X} = (\mathbf{x}_1,\ldots,\mathbf{x}_n)^\top$ and $\mathbf{Z} = (\mathbf{z}_1,\ldots,\mathbf{z}_n)^\top$ have rank p and q, respectively.

For this model, the Pearson's residuals are given by

$$r_i = \frac{y_i - m_i}{s_i}, \quad i=1,\ldots,n,$$

where

$$m_i = \frac{\mu_i\sigma_i+\tau_1}{\sigma_i-\tau_2+1} \quad \text{and} \quad s_i = \frac{(\mu_i\sigma_i+\tau_1)((1+\mu_i)\sigma_i+\tau_1-\tau_2+1)}{(\sigma_i+\tau-1)}(\sigma_i-\tau_2)^{-1/2}.$$

Note that the Pearson's residuals are well defined as long as $\sigma_i-\tau_2>1$, $\forall i=1,\ldots,n$. On the other hand, the modified Pearson's residuals are given by

$$r_i^* = \frac{\text{logit}(y_i) - m_i^*}{s_i^*}, \quad i=1,\ldots,n,$$

where

$$m_i^* = \psi(\mu_i\sigma_i+\tau_1)-\psi((1+\mu_i)\sigma_i+\tau_1-\tau_2+1) \quad \text{and} \quad s_i^* = \sqrt{\psi'(\mu_i\sigma_i+\tau_1)-\psi'((1+\mu_i)\sigma_i+\tau_1-\tau_2+1)},$$

with $\psi(\cdot)$ and $\psi'(\cdot)$ denoting the digamma and trigamma functions, respectively. Finally, the quantile residuals are given by

$$r_i^q = \Phi^{-1}\left(I_{\frac{y_i}{1+y_i}}(\mu_i\sigma_i+\tau_1,(1+\mu_i)\sigma_i+\tau_1-\tau_2+1)\right), \quad i=1,\ldots,n,$$

where $\Phi^{-1}(\cdot)$ denotes the inverse of the cumulative distribution function for the standard normal model and $I_y(\alpha,\beta) = B_x(\alpha,\beta)/B(\alpha,\beta)$ is the incomplete beta function ratio, $B_x(\alpha,\beta) = \int_0^x \omega^{\alpha-1}(1-\omega)^{\beta-1}\mathrm{d}\omega$ is the incomplete beta function, $B(\alpha,\beta) = \Gamma(\alpha)\Gamma(\beta)/\Gamma(\alpha+\beta)$ is the beta function and $\Gamma(\alpha) = \int_0^\infty \omega^{\alpha-1}\mathrm{e}^{-\omega}\mathrm{d}\omega$ is the gamma function. dRBP gives the density, pRBP gives the distribution function, qRBP gives the quantile function, and rRBP generates random deviates from the beta prime distribution with the specified parameterization.

Value

an object of class "rregm" is returned. The object returned for these functions is a list containing the following components:

`estimate`	A matrix containing the estimates and standard errors.
`logLik`	the log-likelihood function evaluated at the corresponding estimators.
`AIC`	the Akaike information criterion.
`BIC`	the Bayesian information criterion.
`tau1`, `tau2`	values for tau1 and tau2, depending on the considered parameterization.
`pearson.res`	Pearson's residuals.
`mod.pearson.res`	modified Pearson's residuals.
`quant.res`	quantile residuals.
`convergence`	logical. If convergence was attained.
`dist`	BP (the beta prime distribution).
`param`	The specified parameterization.
`mu.x`	design matrix for mu.
`sigma.x`	design matrix for sigma.

Author(s)

Diego Gallardo and Marcelo Bourguignon.

References

Bourguignon, M., Gallardo, D.I. (2025) A general and unified parameterization of the beta distribution: A flexible and robust beta regression model. Statistica Neerlandica, 79(2), e70007.

See Also

fit.RBE, fit.RGA, fit.RIG, fit.RLN, fit.RLL

Examples

```
set.seed(2100)
n=100; x1=rnorm(max(n)) ##drawing covariates, the same for mu and sigma
mu=exp(0.5-0.4*x1); sigma=exp(-0.1+0.05*x1)
y=rRBP(n, mu, sigma, param="MD") ## model parameterized in the median
data=list(y=y, x1=x1)
aux.RBP=fit.RBP(y~x1, sigma.formula=~x1, data=data, param="MD")
summary(aux.RBP)
qqnorm(res(aux.RBP, type="mod.pearson"))
#The beta prime model parameterized in the median also can be fitted using gamlss
#gamlss(y~x1, sigma.formula=~x1, data=data, family=BPMD)
```

RGA *Tools for a reparameterized gamma regression model*

Description

A set of functions related to the reparameterized gamma regression model based on different measures of central tendency: mean, median, mode, geometric mean or harmonic mean.

Usage

```
GAM(mu.link = "log", sigma.link = "log")
GGM(mu.link = "log", sigma.link = "log")
GHM(mu.link = "log", sigma.link = "log")
GMD(mu.link = "log", sigma.link = "log")
GMO(mu.link = "log", sigma.link = "log")
dGAM(x, mu = 1, sigma = 1, log = FALSE)
dGGM(x, mu = 1, sigma = 1, log = FALSE)
dGHM(x, mu = 1, sigma = 1, log = FALSE)
dGMD(x, mu = 1, sigma = 1, log = FALSE)
dGMO(x, mu = 1, sigma = 1, log = FALSE)
dRGA(x, mu=1, sigma=1, param="AM", log=FALSE)
fit.RGA(formula = formula(data), sigma.formula=~1, data, param="AM")
pGAM(q, mu = 1, sigma = 1, lower.tail = TRUE, log.p = FALSE)
pGGM(q, mu = 1, sigma = 1, lower.tail = TRUE, log.p = FALSE)
pGHM(q, mu = 1, sigma = 1, lower.tail = TRUE, log.p = FALSE)
pGMD(q, mu = 1, sigma = 1, lower.tail = TRUE, log.p = FALSE)
pGMO(q, mu = 1, sigma = 1, lower.tail = TRUE, log.p = FALSE)
pRGA(q, mu=1, sigma=1, param="AM", lower.tail = TRUE, log.p = FALSE)
qGAM(p, mu = 1, sigma = 1, lower.tail = TRUE, log.p = FALSE)
qGGM(p, mu = 1, sigma = 1, lower.tail = TRUE, log.p = FALSE)
qGHM(p, mu = 1, sigma = 1, lower.tail = TRUE, log.p = FALSE)
```

```
qGMD(p, mu = 1, sigma = 1, lower.tail = TRUE, log.p = FALSE)
qGMO(p, mu = 1, sigma = 1, lower.tail = TRUE, log.p = FALSE)
qRGA(p, mu=1, sigma=1, param="AM", lower.tail = TRUE, log.p = FALSE)
rGAM(n, mu = 1, sigma = 1)
rGGM(n, mu = 1, sigma = 1)
rGHM(n, mu = 1, sigma = 1)
rGMD(n, mu = 1, sigma = 1)
rGMO(n, mu = 1, sigma = 1)
rRGA(n, mu=1, sigma=1, param="AM")
```

Arguments

mu.link	the mu link function with default log
sigma.link	the sigma link function with default log
mu, sigma	vector of parameter values
formula	an object of class "*formula*" (or one that can be coerced to that class): a symbolic description of the model to be fitted. The details of model specification are given under 'Details'.
data	an optional data frame, list or environment (or object coercible by *as.data.frame* to a data frame) containing the variables in the model. If not found in data, the variables are taken from environment(formula), typically the environment from which lm is called.
sigma.formula	a formula object for fitting a model to the sigma parameter, as in the formula above, e.g. sigma.formula=~x1+x2.
param	parameterization used for the model. "AM" for mean, "MD" for median, "MO" for mode, "GM" for geometric mean, and "HM" for harmonic mean.
x, q	vector of quantiles
p	vector of probabilities
n	number of observations. If length$(n) > 1$, the length is taken to be the number required.
log, log.p	logical; if TRUE, probabilities p are given as log(p).
lower.tail	logical; if TRUE, probabilities are $P(X \leq x)$ otherwise, $P(X > x)$.

Details

The parameterization for the reparameterized gamma distribution is given by

$$f(x;\mu,\sigma) = \frac{(\delta/\mu)^{\delta+\tau}}{\Gamma(\delta+\tau)} x^{\delta+\tau-1} \mathrm{e}^{-\delta x/\mu}, \quad y > 0,$$

where $\delta = \delta(\sigma) = (\sqrt{\sigma(\sigma+4)} + \sigma)/2$, $\mu > 0$, $\sigma > 0$ and τ is a constant. The following cases are highlighted:

- param="AM": $\tau = 0$ and μ represents the mean of the distribution.

- param="GM": $\tau = 1/2$ and μ represents the geometric mean of the distribution.

- param="MD": $\tau = 1/3$ and μ represents the median of the distribution.

- param="MO" or ="HM": $\tau = 1$ and μ represents the mode or the harmonic mean of the distribution.

Suppose the central tendency and the concentration parameter of Y_i satisfies the following functional relations

$$\log(\mu_i) = \mathbf{x}_i^\top \boldsymbol{\xi} \quad \text{and} \quad \log(\sigma_i) = \eta_{2i} = \mathbf{z}_i^\top \boldsymbol{\nu},$$

where $\boldsymbol{\xi} = (\xi_1, \ldots, \xi_p)^\top$ and $\boldsymbol{\nu} = (\nu_1, \ldots, \nu_q)^\top$ are vectors of unknown regression coefficients which are assumed to be functionally independent, $\boldsymbol{\xi} \in \mathbb{R}^p$ and $\boldsymbol{\nu} \in \mathbb{R}^q$, with $p + q < n$, and $\mathbf{x}_i = (x_{i1}, \ldots, x_{ip})^\top$ and $\mathbf{z}_i = (z_{i1}, \ldots, z_{iq})^\top$ are observations on p and q known regressors, for $i = 1, \ldots, n$. Furthermore, we assume that the covariate matrices $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)^\top$ and $\mathbf{Z} = (\mathbf{z}_1, \ldots, \mathbf{z}_n)^\top$ have rank p and q, respectively.

For this model, the Pearson's residuals are given by

$$r_i = \frac{y_i - m_i}{s_i}, \quad i = 1, \ldots, n,$$

where

$$m_i = \mu_i \left(1 + \frac{\tau}{k(\sigma_i, \tau)}\right) \quad \text{and} \quad s_i = \frac{\mu}{k(\sigma_i, \tau)} \sqrt{\tau + k(\sigma_i, \tau)},$$

where $k(\sigma_i, \tau) = (\sqrt{\sigma_i(\sigma_i + 4\tau)} + \sigma_i)/2$. On the other hand, the modified Pearson's residuals are given by

$$r_i^* = \frac{\log(y_i) - m_i^*}{s_i^*}, \quad i = 1, \ldots, n,$$

where

$$m_i^* = \psi(\tau + k(\sigma_i, \tau)) + \log \mu_i - \log k(\sigma_i, \tau)) \quad \text{and} \quad s_i^* = \sqrt{\psi'(\tau + k(\sigma_i, \tau))},$$

with $\psi(\cdot)$ and $\psi'(\cdot)$ denoting the digamma and trigamma functions, respectively. Finally, the quantile residuals are given by

$$r_i^q = \Phi^{-1}\left(\frac{\gamma(\tau + k(\sigma_i, \tau), k(\sigma_i, \tau) y_i/\mu_i}{\Gamma(\tau + k(\sigma_i, \tau))}\right), \quad i = 1, \ldots, n,$$

where $\Phi^{-1}(\cdot)$ denotes the inverse of the cumulative distribution function for the standard normal model and $\gamma(a, z) = \int_0^z t^{a-1} \mathrm{e}^{-t} \mathrm{d}t$ is the lower incomplete gamma function and $\Gamma(\alpha) = \int_0^\infty \omega^{\alpha-1} \mathrm{e}^{-\omega} \mathrm{d}\omega$ is the gamma function. dRGA gives the density, pRGA gives the distribution function, qRGA gives the quantile function, and rRGA generates random deviates from the gamma distribution with the specified parameterization.

Value

an object of class "rregm" is returned. The object returned for these functions is a list containing the following components:

`estimate`	A matrix containing the estimates and standard errors.
`logLik`	the log-likelihood function evaluated at the corresponding estimators.
`AIC`	the Akaike information criterion.
`BIC`	the Bayesian information criterion.
`tau1`, `tau2`	values for tau1 and tau2, depending on the considered parameterization.

pearson.res	Pearson's residuals.
mod.pearson.res	modified Pearson's residuals.
quant.res	quantile residuals.
convergence	logical. If convergence was attained.
dist	GA (the gamma distribution).
param	The specified parameterization.
mu.x	design matrix for mu.
sigma.x	design matrix for sigma.

Author(s)

Diego Gallardo and Marcelo Bourguignon.

References

Bourguignon, M., Gallardo, D.I. (2025) A general and unified class of gamma regression models. Chemometrics and Intelligent Laboratory Systems, 261, 105382.

See Also

fit.RBE, fit.RBP, fit.RIG, fit.RLN, fit.RLL

Examples

```
set.seed(2100)
n=100; x1=rnorm(max(n)) ##drawing covariates, the same for mu and sigma
mu=exp(0.5-0.4*x1); sigma=exp(-0.1+0.05*x1)
y=rRGA(n, mu, sigma, param="MD") ## model parameterized in the median
data=list(y=y, x1=x1)
aux.RGA=fit.RGA(y~x1, sigma.formula=~x1, data=data, param="MD")
summary(aux.RGA)
qqnorm(res(aux.RGA, type="mod.pearson"))
#The gamma parameterized in the median also can be fitted using gamlss
#gamlss(y~x1, sigma.formula=~x1, data=data, family=GMD)
```

RIG *Tools for a reparameterized inverse gamma regression model*

Description

A set of functions related to the reparameterized inverse gamma regression model based on different measures of central tendency: mean, median, mode, geometric mean or harmonic mean.

Usage

```
IGAM(mu.link = "log", sigma.link = "log")
IGGM(mu.link = "log", sigma.link = "log")
IGHM(mu.link = "log", sigma.link = "log")
IGMD(mu.link = "log", sigma.link = "log")
IGMO(mu.link = "log", sigma.link = "logshiftto1")
dIGAM(x, mu = 1, sigma = 1, log = FALSE)
dIGGM(x, mu = 1, sigma = 1, log = FALSE)
dIGHM(x, mu = 1, sigma = 1, log = FALSE)
dIGMD(x, mu = 1, sigma = 1, log = FALSE)
dIGMO(x, mu = 1, sigma = 1.5, log = FALSE)
dRIG(x, mu=1, sigma=1.5, param="AM", log=FALSE)
fit.RIG(formula = formula(data), sigma.formula=~1, data, param="AM")
pIGAM(q, mu = 1, sigma = 1, lower.tail = TRUE, log.p = FALSE)
pIGGM(q, mu = 1, sigma = 1, lower.tail = TRUE, log.p = FALSE)
pIGHM(q, mu = 1, sigma = 1, lower.tail = TRUE, log.p = FALSE)
pIGMD(q, mu = 1, sigma = 1, lower.tail = TRUE, log.p = FALSE)
pIGMO(q, mu = 1, sigma = 1.5, lower.tail = TRUE, log.p = FALSE)
pRIG(q, mu=1, sigma=1.5, param="AM", lower.tail = TRUE, log.p = FALSE)
qIGAM(p, mu = 1, sigma = 1, lower.tail = TRUE, log.p = FALSE)
qIGGM(p, mu = 1, sigma = 1, lower.tail = TRUE, log.p = FALSE)
qIGHM(p, mu = 1, sigma = 1, lower.tail = TRUE, log.p = FALSE)
qIGMD(p, mu = 1, sigma = 1, lower.tail = TRUE, log.p = FALSE)
qIGMO(p, mu = 1, sigma = 1.5, lower.tail = TRUE, log.p = FALSE)
qRIG(p, mu=1, sigma=1.5, param="AM", lower.tail = TRUE, log.p = FALSE)
rIGAM(n, mu = 1, sigma = 1)
rIGGM(n, mu = 1, sigma = 1)
rIGHM(n, mu = 1, sigma = 1)
rIGMD(n, mu = 1, sigma = 1)
rIGMO(n, mu = 1, sigma = 1.5)
rRIG(n, mu=1, sigma=1.5, param="AM")
```

Arguments

mu.link	the mu link function with default log
sigma.link	the sigma link function with default log
mu, sigma	vector of parameter values
formula	an object of class "*formula*" (or one that can be coerced to that class): a symbolic description of the model to be fitted. The details of model specification are given under 'Details'.
data	an optional data frame, list or environment (or object coercible by *as.data.frame* to a data frame) containing the variables in the model. If not found in data, the variables are taken from environment(formula), typically the environment from which lm is called.
sigma.formula	a formula object for fitting a model to the sigma parameter, as in the formula above, e.g. sigma.formula=~x1+x2.

param	parameterization used for the model. "AM" for mean, "MD" for median, "MO" for mode, "GM" for geometric mean, and "HM" for harmonic mean.
x, q	vector of quantiles
p	vector of probabilities
n	number of observations. If length$(n) > 1$, the length is taken to be the number required.
log, log.p	logical; if TRUE, probabilities p are given as log(p).
lower.tail	logical; if TRUE, probabilities are $P(X \leq x)$ otherwise, $P(X > x)$.

Details

The parameterization for the reparameterized inverse gamma distribution is given by

$$f(y;\mu,\sigma) = \frac{(\sigma\mu)^{\sigma+\tau}}{\Gamma(\sigma+\tau)} y^{-\sigma-\tau-1} \mathrm{e}^{-\frac{\sigma\mu}{y}}, \quad y > 0,$$

where $0 < \mu < 1$, $\sigma > 0$ and τ is a constant. The following cases are highlighted:

- param="AM": $\tau = 1$ and μ represents the mean of the distribution.

- param="GM": $\tau = 1/2$ and μ represents the geometric mean of the distribution.

- param="HM": $\tau = 0$ and μ represents the harmonic mean of the distribution.

- param="MO": $\tau = -1$ and μ represents the mode of the distribution.

- param="MD": $\tau = 1/3$ and μ represents the median of the distribution.

Suppose the central tendency and the concentration parameter of Y_i satisfies the following functional relations

$$\log(\mu_i) = \mathbf{x}_i^\top \boldsymbol{\xi} \quad \text{and} \quad \log(\sigma_i) = \eta_{2i} = \mathbf{z}_i^\top \boldsymbol{\nu},$$

$\boldsymbol{\xi} = (\xi_1, \ldots, \xi_p)^\top$ and $\boldsymbol{\nu} = (\nu_1, \ldots, \nu_q)^\top$ are vectors of unknown regression coefficients which are assumed to be functionally independent, $\boldsymbol{\xi} \in \mathbb{R}^p$ and $\boldsymbol{\nu} \in \mathbb{R}^q$, with $p + q < n$, and $\mathbf{x}_i = (x_{i1}, \ldots, x_{ip})^\top$ and $\mathbf{z}_i = (z_{i1}, \ldots, z_{iq})^\top$ are observations on p and q known regressors, for $i = 1, \ldots, n$. Furthermore, we assume that the covariate matrices $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)^\top$ and $\mathbf{Z} = (\mathbf{z}_1, \ldots, \mathbf{z}_n)^\top$ have rank p and q, respectively.

For this model, the Pearson's residuals are given by

$$r_i = \frac{y_i - m_i}{s_i}, \quad i = 1, \ldots, n,$$

where

$$m_i = \frac{\mu_i\sigma_i}{\sigma_i + \tau_2} \quad \text{and} \quad s_i = \frac{\mu_i\sigma_i}{(\sigma_i + \tau - 1)} (\sigma_i + \tau - 2)^{-1/2}.$$

Note that the Pearson's residuals are well defined as long as $\sigma_i + \tau > 2$, $\forall i = 1, \ldots, n$. On the other hand, the modified Pearson's residuals are given by

$$r_i^* = \frac{-\log(y_i) - m_i^*}{s_i^*}, \quad i = 1, \ldots, n,$$

where

$$m_i^* = \psi(\sigma_i + \tau) - \log(\mu_i\sigma_i) \quad \text{and} \quad s_i^* = \sqrt{\psi'(\sigma_i + \tau)},$$

with $\psi(\cdot)$ and $\psi'(\cdot)$ denoting the digamma and trigamma functions, respectively. Finally, the quantile residuals are given by

$$r_i^q = \Phi^{-1}\left(\frac{\gamma(\sigma_i+\tau, \mu_i\sigma_i/y_i)}{\Gamma(\sigma_i+\tau)}\right), \quad i = 1, \ldots, n,$$

where $\Phi^{-1}(\cdot)$ denotes the inverse of the cumulative distribution function for the standard normal model and $\gamma(a, z) = \int_0^z t^{a-1}\mathrm{e}^{-t}\mathrm{d}t$ is the lower incomplete gamma function and $\Gamma(\alpha) = \int_0^\infty \omega^{\alpha-1}\mathrm{e}^{-\omega}\mathrm{d}\omega$ is the gamma function. dRIG gives the density, pRIG gives the distribution function, qRIG gives the quantile function, and rRIG generates random deviates from the inverse gamma distribution with the specified parameterization. In addition, dIGXX, pIGXX, qIGXX and rIGXX also provide the equivalent functions for a specified parameterization for XX: AM (mean), GM (geometric mean), HM (harmonic mean), MD (median) and MO (mode). For instance, dIGAM gives the density for the inverse gamma model parameterized in the mean, pIGGM gives the distribution function for the inverse gamma model parameterized in the geometric mean and so on. Finally, the functions IGAM, IGGM, IGHM, IGMD and IGMO also provide a framework to fit models with *gamlss*.

Value

an object of class "rregm" is returned. The object returned for these functions is a list containing the following components:

`estimate`	A matrix containing the estimates and standard errors.
`logLik`	the log-likelihood function evaluated at the corresponding estimators.
`AIC`	the Akaike information criterion.
`BIC`	the Bayesian information criterion.
`tau`	values for tau, depending on the considered parameterization.
`pearson.res`	Pearson's residuals.
`mod.pearson.res`	modified Pearson's residuals.
`quant.res`	quantile residuals.
`convergence`	logical. If convergence was attained.
`dist`	IG (the inverse gamma distribution).
`param`	The specified parameterization.
`mu.x`	design matrix for mu.
`sigma.x`	design matrix for sigma.

Author(s)

Diego Gallardo and Marcelo Bourguignon.

References

Bourguignon, M., Gallardo, D.I. (2025) A general and unified class of gamma regression models. Chemometrics and Intelligent Laboratory Systems, 261, 105382.

See Also

fit.RBE, fit.RGA, fit.RBP, fit.RLN, fit.RLL

Examples

```
set.seed(2100)
n=100; x1=rnorm(max(n)) ##drawing covariates, the same for mu and sigma
mu=exp(0.5-0.4*x1); sigma=exp(-0.1+0.05*x1)
y=rRIG(n, mu, sigma, param="MD") ## model parameterized in the median
data=list(y=y, x1=x1)
aux.RIG=fit.RIG(y~x1, sigma.formula=~x1, data=data, param="MD")
summary(aux.RIG)
qqnorm(res(aux.RIG, type="mod.pearson"))
#The inverse gamma model parameterized in the median also can be fitted using gamlss
#gamlss(y~x1, sigma.formula=~x1, data=data, family=IGMD)
```

RLL *Tools for a reparameterized log-logistic regression model*

Description

A set of functions related to the reparameterized log-logistic regression model based on different parameterizations1: standard (LL), proportional odds (LL2), mean (LL3), harmonic mean (LL4), mode (LL5) and quantile (LL6).

Usage

```
LL(mu.link = "log", sigma.link = "log")
LL2(mu.link = "log", sigma.link = "log")
LL3(mu.link = "log", sigma.link = "logshiftto1")
LL4(mu.link = "log", sigma.link = "logshiftto1")
LL5(mu.link = "log", sigma.link = "logshiftto1")
dLL(x, mu = 1, sigma = 1, log = FALSE)
dLL2(x, mu = 1, sigma = 1, log = FALSE)
dLL3(x, mu = 1, sigma = 1.5, log = FALSE)
dLL4(x, mu = 1, sigma = 1.5, log = FALSE)
dLL5(x, mu = 1, sigma = 1.5, log = FALSE)
dLL6(x, mu = 1, sigma = 1.5, nu=0.5, log = FALSE)
dRLL(x, mu=1, sigma=1.5, nu=0.5, param="LL", log=FALSE)
fit.RLL(formula = formula(data), sigma.formula = ~1, data, param = "LL", nu=0.5)
fit.RLL.cens(formula = formula(data), sigma.formula = ~1, data, param = "LL", nu = 0.5)
pLL(q, mu = 1, sigma = 1, lower.tail = TRUE, log.p = FALSE)
pLL2(q, mu = 1, sigma = 1, lower.tail = TRUE, log.p = FALSE)
pLL3(q, mu = 1, sigma = 1.5, lower.tail = TRUE, log.p = FALSE)
pLL4(q, mu = 1, sigma = 1.5, lower.tail = TRUE, log.p = FALSE)
pLL5(q, mu = 1, sigma = 1.5, lower.tail = TRUE, log.p = FALSE)
pLL6(q, mu = 1, sigma = 1.5, nu=0.5, lower.tail = TRUE, log.p = FALSE)
```

```
pRLL(q, mu=1, sigma=1.5, nu=0.5, param="LL", lower.tail = TRUE, log.p = FALSE)
qLL(p, mu = 1, sigma = 1, lower.tail = TRUE, log.p = FALSE)
qLL2(p, mu = 1, sigma = 1, lower.tail = TRUE, log.p = FALSE)
qLL3(p, mu = 1, sigma = 1.5, lower.tail = TRUE, log.p = FALSE)
qLL4(p, mu = 1, sigma = 1.5, lower.tail = TRUE, log.p = FALSE)
qLL5(p, mu = 1, sigma = 1.5, lower.tail = TRUE, log.p = FALSE)
qLL6(p, mu = 1, sigma = 1.5, nu=0.5, lower.tail = TRUE, log.p = FALSE)
qRLL(p, mu=1, sigma=1.5, nu=0.5, param="LL", lower.tail = TRUE, log.p = FALSE)
rLL(n, mu = 1, sigma = 1)
rLL2(n, mu = 1, sigma = 1)
rLL3(n, mu = 1, sigma = 1.5)
rLL4(n, mu = 1, sigma = 1.5)
rLL5(n, mu = 1, sigma = 1.5)
rLL6(n, mu = 1, sigma = 1.5, nu=0.5)
rRLL(n, mu=1, sigma=1.5, nu=0.5, param="LL")
```

Arguments

mu.link	the mu link function with default log.
sigma.link	the sigma link function with default log for parameterizations1 LL, LL2 and LL6 and logshiftto1 for parameterizations1 LL3, LL4 and LL5.
mu, sigma	vector of parameter values
formula	an object of class "*formula*" (or one that can be coerced to that class): a symbolic description of the model to be fitted. The details of model specification are given under 'Details'.
data	an optional data frame, list or environment (or object coercible by *as.data.frame* to a data frame) containing the variables in the model. If not found in data, the variables are taken from environment(formula), typically the environment from which lm is called.
sigma.formula	a formula object for fitting a model to the sigma parameter, as in the formula above, e.g. sigma.formula=~x1+x2.
param	parameterization used for the model. "LL" for standard, "LL2" for proportional odds, "LL3" for mean, "LL4" for harmonic mean, "LL5" for mode and "LL6" for quantile.
x, q	vector of quantiles
p	vector of probabilities
n	number of observations. If length$(n) > 1$, the length is taken to be the number required.
nu	quantile modeled (only for LL6 parametrization).
log, log.p	logical; if TRUE, probabilities p are given as log(p).
lower.tail	logical; if TRUE, probabilities are $P(X \leq x)$ otherwise, $P(X > x)$.

Details

The parameterization for the log-logistic distribution is given by

$$f(y;\rho,\lambda) = \frac{\rho}{\lambda}\frac{(t/\lambda)^{\rho-1}}{[1+(t/\lambda)^{\rho}]^2}, \quad y,\lambda,\rho > 0.$$

The LL2–LL6 parameterizations1 corresponds to the following cases ($\psi = \pi/\rho$):

- "LL2": $\sigma = \rho$ and $\mu = \lambda^{-\alpha}$ (proportional odds).

- "LL3": $\sigma = \rho$ and $\mu = \psi\lambda/\sin(\psi)$ represents the mean of the distribution.

- "LL4": $\sigma = \rho$ and $\mu = \lambda/[\psi\sin(\psi)]$ represents the harmonic mean of the distribution.

- "LL5": $\sigma = \rho$ and $\mu = \lambda\left(\frac{\rho-1}{\rho+1}\right)^{1/\rho}$ represents the mode of the distribution.

- "LL6": $\sigma = \rho$ and $\mu = \lambda\left(\frac{\nu}{1-\nu}\right)^{1/\rho}$ represents the ν-th quantile of the distribution.

For the LL, LL2 and LL6 parameterizations1, suppose the parameters μ and σ satisfies the following functional relations

$$\log(\mu_i) = \mathbf{x}_i^{\top}\boldsymbol{\xi} \quad \text{and} \quad \log(\sigma_i) = \eta_{2i} = \mathbf{z}_i^{\top}\boldsymbol{\nu},$$

where $\boldsymbol{\xi} = (\xi_1,\ldots,\xi_p)^{\top}$ and $\boldsymbol{\nu} = (\nu_1,\ldots,\nu_q)^{\top}$ are vectors of unknown regression coefficients which are assumed to be functionally independent, $\boldsymbol{\xi} \in \mathbb{R}^p$ and $\boldsymbol{\nu} \in \mathbb{R}^q$, with $p+q < n$, and $\mathbf{x}_i = (x_{i1},\ldots,x_{ip})^{\top}$ and $\mathbf{z}_i = (z_{i1},\ldots,z_{iq})^{\top}$ are observations on p and q known regressors, for $i = 1,\ldots,n$. Furthermore, we assume that the covariate matrices $\mathbf{X} = (\mathbf{x}_1,\ldots,\mathbf{x}_n)^{\top}$ and $\mathbf{Z} = (\mathbf{z}_1,\ldots,\mathbf{z}_n)^{\top}$ have rank p and q, respectively.

For the LL3, LL4 and LL5 parameterizations1, suppose σ satisfies the following functional relations

$$\log(\sigma_i - 1) = \eta_{2i} = \mathbf{z}_i^{\top}\boldsymbol{\nu},$$

For this model, the Pearson's residuals are given by

$$r_i = \frac{y_i - m_{y_i}}{\sqrt{(v_i)}}, \quad i = 1,\ldots,n,$$

where $m_{y_i} = lambda_i{*}(pi/sigma_i)/sin(pi/sigma_i)$ and $v_i = (2{*}lambda_i^2{*}(pi/sigma_i)/sin(2{*}pi/sigma_i) - m_{y_i}^2)$. On the other hand, the modified Pearson's residuals are given by

$$r_i^* = \frac{y_i - \lambda_i}{\sqrt{(V_i)}}, \quad i = 1,\ldots,n,$$

where $V_i = pi/[sqrt(3) * sigma_i]$. dRLL gives the density, pRLL gives the distribution function, qRLL gives the quantile function, and rRLL generates random deviates from the log-logistic distribution with the specified parameterization.

Value

an object of class "rregm" is returned. The object returned for these functions is a list containing the following components:

`estimate` A matrix containing the estimates and standard errors.

logLik	the log-likelihood function evaluated at the corresponding estimators.
AIC	the Akaike information criterion.
BIC	the Bayesian information criterion.
pearson.res	Pearson's residuals.
mod.pearson.res	modified Pearson's residuals.
quant.res	quantile residuals.
convergence	logical. If convergence was attained.
dist	LL (the log-logistic distribution).
param	The specified parameterization.
mu.x	design matrix for mu.
sigma.x	design matrix for sigma.

Author(s)

Diego Gallardo and Marcelo Bourguignon.

References

Bourguignon, M., Gallardo, D.I. (2025) A general and unified class of gamma regression models. Chemometrics and Intelligent Laboratory Systems, 261, 105382.

See Also

fit.RBE, fit.RGA, fit.RBP, fit.RIG, fit.RLL

Examples

```
set.seed(2100)
n=100; x1=rnorm(max(n)) ##drawing covariates, the same for mu and sigma
mu=exp(0.5-0.4*x1); sigma=exp(-0.1+0.05*x1)
y=rRLL(n, mu, sigma, param="LL2") ## model parameterized in the median
data=list(y=y, x1=x1)
aux.RLL=fit.RLL(y~x1, sigma.formula=~x1, data=data, param="LL2")
summary(aux.RLL)
qqnorm(res(aux.RLL, type="mod.pearson"))
#The log-logistic model with this parameterization also can be fitted using gamlss
#gamlss(y~x1, sigma.formula=~x1, data=data, family=LL2)
```

RLN *Tools for a reparameterized log-normal regression model*

Description

A set of functions related to the reparameterized log-normal regression model based on different measures of central tendency: mean, median, mode, geometric mean or harmonic mean.

Usage

```
LNAM(mu.link = "log", sigma.link = "log")
LNGM(mu.link = "log", sigma.link = "log")
LNHM(mu.link = "log", sigma.link = "log")
LNMD(mu.link = "log", sigma.link = "log")
LNMO(mu.link = "log", sigma.link = "log")
dLNAM(x, mu = 1, sigma = 1, log = FALSE)
dLNGM(x, mu = 1, sigma = 1, log = FALSE)
dLNHM(x, mu = 1, sigma = 1, log = FALSE)
dLNMD(x, mu = 1, sigma = 1, log = FALSE)
dLNMO(x, mu = 1, sigma = 1, log = FALSE)
dRLN(x, mu=1, sigma=1, param="AM", log=FALSE)
fit.RLN(formula = formula(data), sigma.formula=~1, data, param="AM")
pLNAM(q, mu = 1, sigma = 1, lower.tail = TRUE, log.p = FALSE)
pLNGM(q, mu = 1, sigma = 1, lower.tail = TRUE, log.p = FALSE)
pLNHM(q, mu = 1, sigma = 1, lower.tail = TRUE, log.p = FALSE)
pLNMD(q, mu = 1, sigma = 1, lower.tail = TRUE, log.p = FALSE)
pLNMO(q, mu = 1, sigma = 1, lower.tail = TRUE, log.p = FALSE)
pRLN(q, mu=1, sigma=1, param="AM", lower.tail = TRUE, log.p = FALSE)
qLNAM(p, mu = 1, sigma = 1, lower.tail = TRUE, log.p = FALSE)
qLNGM(p, mu = 1, sigma = 1, lower.tail = TRUE, log.p = FALSE)
qLNHM(p, mu = 1, sigma = 1, lower.tail = TRUE, log.p = FALSE)
qLNMD(p, mu = 1, sigma = 1, lower.tail = TRUE, log.p = FALSE)
qLNMO(p, mu = 1, sigma = 1, lower.tail = TRUE, log.p = FALSE)
qRLN(p, mu=1, sigma=1, param="AM", lower.tail = TRUE, log.p = FALSE)
rLNAM(n, mu = 1, sigma = 1)
rLNGM(n, mu = 1, sigma = 1)
rLNHM(n, mu = 1, sigma = 1)
rLNMD(n, mu = 1, sigma = 1)
rLNMO(n, mu = 1, sigma = 1)
rRLN(n, mu=1, sigma=1, param="AM")
```

Arguments

mu.link	the mu link function with default logit
sigma.link	the sigma link function with default log
mu, sigma	vector of parameter values

formula	an object of class "*formula*" (or one that can be coerced to that class): a symbolic description of the model to be fitted. The details of model specification are given under 'Details'.
data	an optional data frame, list or environment (or object coercible by *as.data.frame* to a data frame) containing the variables in the model. If not found in data, the variables are taken from environment(formula), typically the environment from which lm is called.
sigma.formula	a formula object for fitting a model to the sigma parameter, as in the formula above, e.g. sigma.formula=~x1+x2.
param	parameterization used for the model. "AM" for mean, "MD" for median, "MO" for mode, "GM" for geometric mean, and "HM" for harmonic mean.
x, q	vector of quantiles
p	vector of probabilities
n	number of observations. If length$(n) > 1$, the length is taken to be the number required.
log, log.p	logical; if TRUE, probabilities p are given as log(p).
lower.tail	logical; if TRUE, probabilities are $P(X \leq x)$ otherwise, $P(X > x)$.

Details

The parameterization for the reparameterized log-normal distribution is given by

$$f(x;\mu,\sigma) = \frac{1}{x\,\phi\sqrt{2\pi}} \exp\left[-\frac{(\ln x - \theta)^2}{2\phi^2}\right], \quad y > 0,$$

where $\theta = \log(\mu) - \tau\phi$, $\phi = \sqrt{\sigma}$, $\mu > 0$, $\sigma > 0$ and τ is a constant. The following cases are highlighted:

- param="MO": $\tau = -1$ and μ represents the mode of the distribution.

- param="HM": $\tau = -1/2$ and μ represents the harmonic mean of the distribution.

- param="MD" or "GM": $\tau = 0$ and μ represents the median or the geometric mean of the distribution.

- param="AM": $\tau = 1/2$ and μ represents the mean of the distribution.

Suppose the central tendency and the concentration parameter of Y_i satisfies the following functional relations

$$\log(\mu_i) = \mathbf{x}_i^\top \boldsymbol{\xi} \quad \text{and} \quad \log(\sigma_i) = \eta_{2i} = \mathbf{z}_i^\top \boldsymbol{\nu},$$

where $\boldsymbol{\xi} = (\xi_1, \ldots, \xi_p)^\top$ and $\boldsymbol{\nu} = (\nu_1, \ldots, \nu_q)^\top$ are vectors of unknown regression coefficients which are assumed to be functionally independent, $\boldsymbol{\xi} \in \mathbb{R}^p$ and $\boldsymbol{\nu} \in \mathbb{R}^q$, with $p + q < n$, and $\mathbf{x}_i = (x_{i1}, \ldots, x_{ip})^\top$ and $\mathbf{z}_i = (z_{i1}, \ldots, z_{iq})^\top$ are observations on p and q known regressors, for $i = 1, \ldots, n$. Furthermore, we assume that the covariate matrices $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)^\top$ and $\mathbf{Z} = (\mathbf{z}_1, \ldots, \mathbf{z}_n)^\top$ have rank p and q, respectively.

For this model, the Pearson's residuals are given by

$$r_i = \frac{y_i - exp(\theta_i + \phi_i^2/2)}{\sqrt{((\exp(\phi_i^2) - 1)\exp(2 * \theta_i + \phi_i^2))}}, \quad i = 1, \ldots, n.$$

On the other hand, the modified Pearson's residuals are given by

$$r_i^* = \frac{\log(y_i) - \theta_i}{\phi_i}, \quad i = 1, \ldots, n,$$

which coincide with the quantile residuals. dRLN gives the density, pRLN gives the distribution function, qRLN gives the quantile function, and rRLN generates random deviates from the log-normal distribution with the specified parameterization.

Value

an object of class "rregm" is returned. The object returned for these functions is a list containing the following components:

`estimate`	A matrix containing the estimates and standard errors.
`logLik`	the log-likelihood function evaluated at the corresponding estimators.
`AIC`	the Akaike information criterion.
`BIC`	the Bayesian information criterion.
`tau1`, `tau2`	values for tau1 and tau2, depending on the considered parameterization.
`pearson.res`	Pearson's residuals.
`mod.pearson.res`	modified Pearson's residuals.
`quant.res`	quantile residuals.
`convergence`	logical. If convergence was attained.
`dist`	LN (the log-normal distribution).
`param`	The specified parameterization.
`mu.x`	design matrix for mu.
`sigma.x`	design matrix for sigma.

Author(s)

Diego Gallardo and Marcelo Bourguignon.

References

Bourguignon, M., Gallardo, D.I. (2025) A general and unified class of gamma regression models. Chemometrics and Intelligent Laboratory Systems, 261, 105382.

See Also

fit.RBE, fit.RGA, fit.RBP, fit.RIG, fit.RLL

Examples

```
set.seed(2100)
n=100; x1=rnorm(max(n)) ##drawing covariates, the same for mu and sigma
mu=exp(0.5-0.4*x1); sigma=exp(-0.1+0.05*x1)
y=rRLN(n, mu, sigma, param="MD") ## model parameterized in the median
data=list(y=y, x1=x1)
aux.RLN=fit.RLN(y~x1, sigma.formula=~x1, data=data, param="MD")
summary(aux.RLN)
qqnorm(res(aux.RLN, type="mod.pearson"))
#The log-normal model parameterized in the median also can be fitted using gamlss
#gamlss(y~x1, sigma.formula=~x1, data=data, family=LNMD)
```

`tools.rregm` *Print a summary for a object of the "rregm" class.*

Description

Tools for a objects of the "rregm" class.

Usage

```
res(object, type="pearson")
## S3 method for class 'rregm'
AIC(object, ..., k=2)
## S3 method for class 'rregm'
BIC(object, ...)
## S3 method for class 'rregm'
coef(object, ...)
## S3 method for class 'rregm'
logLik(object, ...)
## S3 method for class 'LRskov'
print(x, digits = max(3L, getOption("digits") - 3L), ...)
## S3 method for class 'rregm'
print(x, digits = max(3L, getOption("digits") - 3L), ...)
## S3 method for class 'rregm'
summary(object, ...)
```

Arguments

x, object	an object of the "rregm" class.
type	type of residuals to be presented: pearson (default), mod.pearson or quantile.
digits	minimal number of significant digits
k	numeric, the penalty per parameter to be used; the default k = 2 is the classical AIC.
...	for extra arguments.

Details

Supported regression models are: - reparametrized beta - reparametrized gamma - reparametrized beta prime - reparametrized inverse gamma - reparametrized log-normal

Value

A complete summary for the coefficients extracted from a "rregm" or "LRskov" object.

Author(s)

Diego Gallardo and Marcelo Bourguignon.

References

Bourguignon, M., Gallardo, D.I. (2025) A general and unified class of gamma regression models. Chemometrics and Intelligent Laboratory Systems, 261, 105382.

Examples

```
set.seed(2100)
n=100; x1=rnorm(max(n)) ##drawing covariates, the same for mu and sigma
mu=exp(0.5-0.4*x1); sigma=exp(-0.1+0.05*x1)
y=rRGA(n, mu, sigma, param="MD") ## model parameterized in the median
data=list(y=y, x1=x1)
aux.RGA=fit.RGA(y~x1, sigma.formula=~x1, data=data, param="MD")
summary(aux.RGA)
qqnorm(res(aux.RGA, type="mod.pearson"))
```

Index

AIC.rregm *(tools.rregm)*, 25

BEAM *(RBE)*, 2
BEGM *(RBE)*, 2
BEHM *(RBE)*, 2
BEMD *(RBE)*, 2
BEMO *(RBE)*, 2
BIC.rregm *(tools.rregm)*, 25
BPAM *(RBP)*, 7
BPGM *(RBP)*, 7
BPHM *(RBP)*, 7
BPMD *(RBP)*, 7
BPMO *(RBP)*, 7

coef.rregm *(tools.rregm)*, 25

dBEAM *(RBE)*, 2
dBEGM *(RBE)*, 2
dBEHM *(RBE)*, 2
dBEMD *(RBE)*, 2
dBEMO *(RBE)*, 2
dBPAM *(RBP)*, 7
dBPGM *(RBP)*, 7
dBPHM *(RBP)*, 7
dBPMD *(RBP)*, 7
dBPMO *(RBP)*, 7
dGAM *(RGA)*, 11
dGGM *(RGA)*, 11
dGHM *(RGA)*, 11
dGMD *(RGA)*, 11
dGMO *(RGA)*, 11
dIGAM *(RIG)*, 14
dIGGM *(RIG)*, 14
dIGHM *(RIG)*, 14
dIGMD *(RIG)*, 14
dIGMO *(RIG)*, 14
dLL *(RLL)*, 18
dLL2 *(RLL)*, 18
dLL3 *(RLL)*, 18
dLL4 *(RLL)*, 18
dLL5 *(RLL)*, 18
dLL6 *(RLL)*, 18
dLNAM *(RLN)*, 22
dLNGM *(RLN)*, 22
dLNHM *(RLN)*, 22
dLNMD *(RLN)*, 22
dLNMO *(RLN)*, 22
dRBE *(RBE)*, 2
dRBP *(RBP)*, 7
dRGA *(RGA)*, 11
dRIG *(RIG)*, 14
dRLL *(RLL)*, 18
dRLN *(RLN)*, 22

fit.RBE *(RBE)*, 2
fit.RBP *(RBP)*, 7
fit.RGA *(RGA)*, 11
fit.RIG *(RIG)*, 14
fit.RLL *(RLL)*, 18
fit.RLN *(RLN)*, 22

GAM *(RGA)*, 11
GGM *(RGA)*, 11
GHM *(RGA)*, 11
GMD *(RGA)*, 11
GMO *(RGA)*, 11

IGAM *(RIG)*, 14
IGGM *(RIG)*, 14
IGHM *(RIG)*, 14
IGMD *(RIG)*, 14
IGMO *(RIG)*, 14

LL *(RLL)*, 18
LL1 *(RLL)*, 18
LL2 *(RLL)*, 18
LL3 *(RLL)*, 18
LL4 *(RLL)*, 18
LL5 *(RLL)*, 18
LL6 *(RLL)*, 18

LNAM (RLN), 22
LNGM (RLN), 22
LNHM (RLN), 22
LNMD (RLN), 22
LNMO (RLN), 22
logLik.rregm (tools.rregm), 25

pBEAM (RBE), 2
pBEGM (RBE), 2
pBEHM (RBE), 2
pBEMD (RBE), 2
pBEMO (RBE), 2
pBPAM (RBP), 7
pBPGM (RBP), 7
pBPHM (RBP), 7
pBPMD (RBP), 7
pBPMO (RBP), 7
pGAM (RGA), 11
pGGM (RGA), 11
pGHM (RGA), 11
pGMD (RGA), 11
pGMO (RGA), 11
pIGAM (RIG), 14
pIGGM (RIG), 14
pIGHM (RIG), 14
pIGMD (RIG), 14
pIGMO (RIG), 14
pLL (RLL), 18
pLL2 (RLL), 18
pLL3 (RLL), 18
pLL4 (RLL), 18
pLL5 (RLL), 18
pLL6 (RLL), 18
pLNAM (RLN), 22
pLNGM (RLN), 22
pLNHM (RLN), 22
pLNMD (RLN), 22
pLNMO (RLN), 22
pRBE (RBE), 2
pRBP (RBP), 7
pRGA (RGA), 11
pRIG (RIG), 14
print.LRskov (tools.rregm), 25
print.rregm (tools.rregm), 25
pRLL (RLL), 18
pRLN (RLN), 22

qBEAM (RBE), 2
qBEGM (RBE), 2
qBEHM (RBE), 2
qBEMD (RBE), 2
qBEMO (RBE), 2
qBPAM (RBP), 7
qBPGM (RBP), 7
qBPHM (RBP), 7
qBPMD (RBP), 7
qBPMO (RBP), 7
qGAM (RGA), 11
qGGM (RGA), 11
qGHM (RGA), 11
qGMD (RGA), 11
qGMO (RGA), 11
qIGAM (RIG), 14
qIGGM (RIG), 14
qIGHM (RIG), 14
qIGMD (RIG), 14
qIGMO (RIG), 14
qLL (RLL), 18
qLL2 (RLL), 18
qLL3 (RLL), 18
qLL4 (RLL), 18
qLL5 (RLL), 18
qLL6 (RLL), 18
qLNAM (RLN), 22
qLNGM (RLN), 22
qLNHM (RLN), 22
qLNMD (RLN), 22
qLNMO (RLN), 22
qRBE (RBE), 2
qRBP (RBP), 7
qRGA (RGA), 11
qRIG (RIG), 14
qRLL (RLL), 18
qRLN (RLN), 22

RBE, 2
RBE.predictive (RBE.skov), 5
RBE.skov, 5
RBE.skovgaard (RBE.skov), 5
rBEAM (RBE), 2
rBEGM (RBE), 2
rBEHM (RBE), 2
rBEMD (RBE), 2
rBEMO (RBE), 2
RBP, 7
rBPAM (RBP), 7
rBPGM (RBP), 7
rBPHM (RBP), 7

rBPMD (RBP), 7
rBPMO (RBP), 7
res (tools.rregm), 25
RGA, 11
rGAM (RGA), 11
rGGM (RGA), 11
rGHM (RGA), 11
rGMD (RGA), 11
rGMO (RGA), 11
RIG, 14
rIGAM (RIG), 14
rIGGM (RIG), 14
rIGHM (RIG), 14
rIGMD (RIG), 14
rIGMO (RIG), 14
RLL, 18
rLL (RLL), 18
rLL2 (RLL), 18
rLL3 (RLL), 18
rLL4 (RLL), 18
rLL5 (RLL), 18
rLL6 (RLL), 18
RLN, 22
rLNAM (RLN), 22
rLNGM (RLN), 22
rLNHM (RLN), 22
rLNMD (RLN), 22
rLNMO (RLN), 22
rRBE (RBE), 2
rRBP (RBP), 7
rRGA (RGA), 11
rRIG (RIG), 14
rRLL (RLL), 18
rRLN (RLN), 22

summary.rregm (tools.rregm), 25

tools.rregm, 25