

Package ‘mixlink’

July 9, 2017

Type Package

Title Mixture Link Regression

Version 0.1.5

Date 2017-07-08

Author Andrew M. Raim

Maintainer Andrew M. Raim <andrew.raim@gmail.com>

Description The Mixture Link model <arXiv:1612.03302> is a proposed extension to generalized linear models, where the outcome distribution is a finite mixture of $J > 1$ densities. This package supports Mixture Link computations for Poisson and Binomial outcomes. This includes the distribution functions, numerical maximum likelihood estimation, Bayesian analysis, and quantile residuals to assess model fit.

License GPL (≥ 2)

Depends mvtnorm, numDeriv, Rcpp

LinkingTo Rcpp, RcppGSL

RoxygenNote 6.0.1

NeedsCompilation yes

Repository CRAN

Date/Publication 2017-07-08 22:48:41 UTC

R topics documented:

DIC	2
Find Vertices	2
Metropolis-Hastings Sampler	3
Metropolis-within-Gibbs Sampler	5
Mixture Link Binomial Distribution	8
Mixture Link Poisson Distribution	9
mlogit	10
normalize	11
Numerical MLE	11
Randomized quantile residuals	13

S3 methods for gibbs.mixlink output objects	14
S3 methods for mle.fit objects	14
S3 methods for mle.fit.ci objects	15
S3 methods for rwmetrop output objects	16

Index	18
--------------	-----------

DIC	<i>Deviance Information Criteria for MCMC Draws</i>
-----	-----------------------------------------------------

Description

Deviance Information Criteria (DIC) computed on results from MCMC sampler.

Usage

DIC(object, ...)

Arguments

object	Results from sampler.
...	optional arguments (currently not used).

Value

DIC

Find Vertices	<i>Compute vertices for Mixture Link</i>
---------------	------------------------------------------

Description

Find vertices of the set $A(\vartheta, \pi)$, which characterizes link between finite mixture mean and regression function.

Usage

find.vertices.prob(mean, Pi, tol = 1e-08)

find.vertices.nonneg(mean, Pi)

Arguments

mean	Parameter ϑ of distribution.
Pi	Parameter π of distribution.
tol	A tolerance to determine if candidate vertices are distinct.

Details

For Mixture Link Binomial, the set $A(\vartheta, \boldsymbol{\pi}) = \{\boldsymbol{\mu} \in [0, 1]^J : \boldsymbol{\mu}^T \boldsymbol{\pi} = \vartheta\}$. For Mixture Link Poisson, the set $A(\vartheta, \boldsymbol{\pi}) = \{\boldsymbol{\mu} \in [0, \infty]^J : \boldsymbol{\mu}^T \boldsymbol{\pi} = \vartheta\}$.

Value

A $J \times k$ matrix whose columns are the vertices of $A(\vartheta, \boldsymbol{\pi})$.

References

Andrew M. Raim, Nagaraj K. Neerchal, and Jorge G. Morel. An Extension of Generalized Linear Models to Finite Mixture Outcomes. arXiv preprint: 1612.03302

Metropolis-Hastings Sampler

Metropolis-Hastings Sampler

Description

Random-Walk Metropolis Hastings sampler for Binomial and Poisson Mixture Link models.

Usage

```
rwmetrop.mixlink.binomial(y, X, m, R, burn = 0, thin = 1,
  invlink = plogis, Beta.init = NULL, Pi.init, kappa.init = NULL,
  hyper = NULL, report.period = R + 1, use.laplace.approx = TRUE,
  proposal = NULL, param.grp = NULL, fixed.kappa = FALSE)
```

```
rwmetrop.mixlink.poisson(y, X, offset = rep(0, length(y)), R, burn = 0,
  thin = 1, invlink = exp, Beta.init = NULL, Pi.init, kappa.init = NULL,
  hyper = NULL, report.period = R + 1, use.laplace.approx = TRUE,
  proposal = NULL, param.grp = NULL, fixed.kappa = FALSE)
```

Arguments

y	Observations.
X	Design matrix for regression.
m	Number of success/failure trials.
R	Number of MCMC draws to take.
burn	Number of initial MCMC draws to discard.
thin	After burn-in period, save one of every thin draws.
invlink	The inverse link function for the mean. Default is plogis for Binomial and exp for Poisson.
Beta.init	Starting value for β .

<code>Pi.init</code>	Starting value for π .
<code>kappa.init</code>	Starting value for κ .
<code>hyper</code>	A list with hyperparameters corresponding to the prior from the Details section. <code>var.Beta</code> is V_β with default 1000. <code>alpha.Pi</code> is α_π with default <code>rep(1, J)</code> . <code>a.kappa</code> and <code>b.kappa</code> correspond to (a_κ, b_κ) which have default values <code>a.kappa = 1</code> and <code>b.kappa = 1/10</code> .
<code>report.period</code>	Report progress every <code>report.period</code> draws.
<code>use.laplace.approx</code>	Maximize a Laplace approximation to the posterior, to find a starting value for MCMC.
<code>proposal</code>	A list with two elements. <code>var</code> is the covariance matrix for a $d + (J - 1) + 1$ dimensional multivariate normal proposal, <code>scale</code> is a scalar multiplied with <code>var</code> . Defaults are <code>var = diag(d+(J-1)+1)</code> and <code>scale = 0.02</code> .
<code>param.grp</code>	A vector of integers of length $d + (J - 1) + 1$, where $d = \text{ncol}(X)$, which indicates the grouping of parameters in MCMC. Parameters with common integers are sampled together. The first d correspond to β , the next $J - 1$ correspond to π , and the last one corresponds to κ . At the default value (<code>param.grp = NULL</code>), β , π , and κ are sampled one at a time, each in their entirety.
<code>fixed.kappa</code>	Keep κ fixed at <code>kappa.init</code> (Default FALSE).
<code>offset</code>	Constant offset term to add to $x^T \beta$.

Details

Priors for Bayesian Mixture Link model are

- $\beta \sim N(\mathbf{0}, V_\beta \mathbf{I})$,
- $\pi \sim \text{Dirichlet}_J(\alpha_\pi)$,
- $\kappa \sim \text{Gamma}(a_\kappa, b_\kappa)$, parameterized with $E(\kappa) = a_\kappa / b_\kappa$.

Value

A list with the MCMC results:

<code>par.hist</code>	$R \times [d + (J - 1) + 1]$ matrix of saved MCMC draws before transformations are applied. Most users will not need this.
<code>Beta.hist</code>	$R \times d$ matrix of saved β draws
<code>Pi.hist</code>	$R \times J$ matrix of saved π draws
<code>kappa.hist</code>	$R \times 1$ vector of κ draws
<code>accept</code>	Percentages that MCMC proposals were accepted. Corresponds to <code>param.grp</code>
<code>elapsed.sec</code>	Elapsed time for sampling, in seconds.
<code>laplace.out</code>	Output of Laplace approximation.
<code>R.keep</code>	Number of draws kept, after thinning and burn-in.
<code>X.names</code>	Names of columns of design matrix.

Can be accessed with the functions `print`, `summary`, and `DIC`.

References

Andrew M. Raim, Nagaraj K. Neerchal, and Jorge G. Morel. An Extension of Generalized Linear Models to Finite Mixture Outcomes. arXiv preprint: 1612.03302

Examples

```
## Not run:
library(Matrix)

# ----- Generate data -----
n <- 200
x <- runif(n, 1, 3)
X <- model.matrix(~ x)
Beta.true <- c(0, 1)
mean.true <- exp(X %*% Beta.true)
kappa.true <- 0.95
Pi.true <- c(1,3)/4
d <- ncol(X)
J <- length(Pi.true)
y <- r.mixlink.pois(n, mean.true, Pi.true, kappa.true)

# ----- Run Metropolis-within-Gibbs sampler -----
hyper <- list(VBeta = diag(1000, d), alpha.Pi = rep(1, J),
kappa.a = 1, kappa.b = 1/2)
proposal <- list(
var = bdiag(solve(t(X) %*% X), diag(J-1), 1),
scale = 0.5)
metrop.out <- rwmeterop.mixlink.poisson(y, X, R = 20000, burn = 1000,
thin = 10, Pi.init = c(1,9)/10, hyper = hyper,
report.period = 1000, use.laplace.approx = TRUE, proposal = proposal)

print(metrop.out)
DIC(metrop.out)

## End(Not run)
```

Metropolis-within-Gibbs Sampler

Metropolis-within-Gibbs Sampler

Description

Metropolis-within-Gibbs sampler for Binomial and Poisson Mixture Link models.

Usage

```
gibbs.mixlink(y, X, R, burn = 0, thin = 1, invlink = NULL,
report.period = R + 1, save.psi = FALSE, Beta.init = NULL, Pi.init,
```

```
kappa.init = NULL, psi.init = NULL, proposal.VBeta = NULL,
proposal.VPi = NULL, proposal.Vkappa = NULL, proposal.Vpsi = NULL,
hyper = NULL, trials = NULL, offset = rep(0, length(y)),
family = NULL, fixed.kappa = FALSE)
```

Arguments

y	Observations.
X	Design matrix for regression.
R	Number of MCMC draws to take.
burn	Number of initial MCMC draws to discard.
thin	After burn-in period, save one of every thin draws.
invlink	The inverse link function for the mean. Default is NULL, which indicates to use plogis for Binomial and exp for Poisson.
report.period	Report progress every report.period draws.
save.psi	Save draws for ψ_1, \dots, ψ_n . Default is FALSE.
Beta.init	Starting value for β .
Pi.init	Starting value for π . The length of Pi.init determines J used in MCMC.
kappa.init	Starting value for κ .
psi.init	Starting value for ψ_1, \dots, ψ_n
proposal.VBeta	Covariance matrix for d dimensional multivariate normal proposal. If left at the default (NULL), we use $\text{diag}(0.01^2, d)$.
proposal.VPi	Covariance matrix for $J - 1$ dimensional multivariate normal proposal. If left at the default (NULL), we use $\text{diag}(0.01^2, J-1)$.
proposal.Vkappa	Covariance matrix for univariate normal proposal. If left at the default (NULL), we use 0.01^2 .
proposal.Vpsi	Covariance matrix for univariate normal proposal. If left at the default (NULL), we use $\text{diag}(0.01^2, J)$.
hyper	A list with hyperparameters corresponding to the prior from the Details section. var.Beta is V_β with default 1000. alpha.Pi is α_π with default $\text{rep}(1, J)$. a.kappa and b.kappa correspond to (a_κ, b_κ) which have default values a.kappa = 1 and b.kappa = 1/10.
trials	Number of success/failure trials.
offset	Offset term to add to regression function, as commonly used in count models.
family	Can be either binomial or poisson.
fixed.kappa	Keep κ fixed at kappa.init (Default FALSE).

Details

Priors for Bayesian Mixture Link model are

- $\beta \sim N(\mathbf{0}, V_\beta \mathbf{I})$,
- $\pi \sim \text{Dirichlet}_J(\alpha_\pi)$,
- $\kappa \sim \text{Gamma}(a_\kappa, b_\kappa)$, parameterized with $E(\kappa) = a_\kappa / b_\kappa$.

Value

A list with the MCMC results:

Beta.hist	$R \times d$ matrix of saved β draws.
Pi.hist	$R \times J$ matrix of saved π draws.
kappa.hist	$R \times 1$ vector of κ draws.
psi.hist	A list of J $R \times n$ matrices. The j th matrix contains the MCMC draws for ψ_{ij} .
accept.Beta	Percentages of MCMC proposals for β were accepted.
accept.Pi	Percentages of MCMC proposals for π were accepted.
accept.kappa	Percentages of MCMC proposals for κ were accepted.
accept.psi	Percentages of MCMC proposals for ψ were accepted.
elapsed.sec	Elapsed time for sampling, in seconds.
R	Number of total draws.
R.keep	Number of draws kept, after thinning and burn-in.
X.names	Names of columns of design matrix.
family	Name of family / outcome.

Can be accessed with the functions `print`, `summary`, and `DIC`.

References

Andrew M. Raim, Nagaraj K. Neerchal, and Jorge G. Morel. An Extension of Generalized Linear Models to Finite Mixture Outcomes. arXiv preprint: 1612.03302

Examples

```
# ----- Generate data -----
n <- 200
m <- rep(20, n)
x <- rnorm(n, 0, 1)
X <- model.matrix(~ x)
Beta.true <- c(-1, 1)
mean.true <- plogis(X %*% Beta.true)
kappa.true <- 1
Pi.true <- c(1,3)/4
d <- ncol(X)
J <- length(Pi.true)
y <- r.mixlink.binom(n, mean.true, Pi.true, kappa.true, m)

# ----- Run Metropolis-within-Gibbs sampler -----
hyper <- list(VBeta = diag(1000, d), alpha.Pi = rep(1, J),
kappa.a = 1, kappa.b = 1/2)
gibbs.out <- gibbs.mixlink(y, X, R = 10, burn = 5, thin = 1,
invlink = plogis, report.period = 100, Pi.init = c(1,9)/10,
proposal.VBeta = solve(t(X) %*% X), proposal.VPi = diag(0.25^2, J-1),
proposal.Vkappa = 0.5^2, proposal.Vpsi = diag(0.5^2, J),
hyper = hyper, family = "binomial", trials = m)
```

```
print(gibbs.out)
DIC(gibbs.out)
```

Mixture Link Binomial Distribution
Distribution functions

Description

Functions for Mixture Link Binomial distribution

Usage

```
r.mixlink.binom(n, mean, Pi, kappa, m, save.latent = FALSE)
d.mixlink.binom(y, m, mean, Pi, kappa, log = FALSE)
p.mixlink.binom(y, m, mean, Pi, kappa)
```

Arguments

n	Number of observations to draw
mean	Parameter ϑ of distribution
Pi	Parameter π of distribution
kappa	Parameter κ of distribution
m	Number of success/failure trials
save.latent	Save intermediate latent variables used during draw.
y	Argument of pdf or cdf
log	Return log of the result (TRUE or FALSE)

Value

d.mixlink.binom gives the density, p.mixlink.binom gives the distribution function, and r.mixlink.binom generates random deviates.

References

Andrew M. Raim, Nagaraj K. Neerchal, and Jorge G. Morel. An Extension of Generalized Linear Models to Finite Mixture Outcomes. arXiv preprint: 1612.03302

Examples

```

mean.true <- 1/3
Pi.true <- c(1/5, 4/5)
kappa.true <- 0.5
m <- 10
r.mixlink.binom(n = 30, mean.true, Pi.true, kappa.true, m)
d.mixlink.binom(y = 5, m, mean.true, Pi.true, kappa.true)
d.mixlink.binom(y = 5, m, mean.true, Pi.true, kappa.true, log = TRUE)
p.mixlink.binom(y = 5, m, mean.true, Pi.true, kappa.true)

```

Mixture Link Poisson Distribution
Distribution functions

Description

Functions for Mixture Link Poisson distribution

Usage

```

r.mixlink.pois(n, mean, Pi, kappa, save.latent = FALSE)

d.mixlink.pois(y, mean, Pi, kappa, log = FALSE)

p.mixlink.pois(y, mean, Pi, kappa)

```

Arguments

<code>n</code>	Number of observations to draw
<code>mean</code>	Parameter ϑ of distribution
<code>Pi</code>	Parameter π of distribution
<code>kappa</code>	Parameter κ of distribution
<code>save.latent</code>	Save intermediate latent variables used during draw.
<code>y</code>	Argument of pdf or cdf
<code>log</code>	Return log of the result (TRUE or FALSE)

Value

`d.mixlink.pois` gives the density, `p.mixlink.pois` gives the distribution function, and `r.mixlink.pois` generates random deviates.

References

Andrew M. Raim, Nagaraj K. Neerchal, and Jorge G. Morel. An Extension of Generalized Linear Models to Finite Mixture Outcomes. arXiv preprint: 1612.03302

Examples

```

mean.true <- 20
Pi.true <- c(1/4, 3/4)
kappa.true <- 0.5
r.mixlink.pois(n = 30, mean.true, Pi.true, kappa.true)
d.mixlink.pois(y = 21, mean.true, Pi.true, kappa.true)
d.mixlink.pois(y = 21, mean.true, Pi.true, kappa.true, log = TRUE)
p.mixlink.pois(y = 21, mean.true, Pi.true, kappa.true)

```

mlogit

Multinomial logit transformation

Description

mlogit transforms from probability simplex S^J to $R^{(J-1)}$ inv.mlogit transforms from $R^{(J-1)}$ to probability simplex S^J

Usage

```
mlogit(p)
```

```
inv.mlogit(x)
```

Arguments

p A J-dimensional numeric vector in the probability simplex

x A (J-1)-dimensional numeric vector

Value

A numeric vector

Examples

```

x <- c(1,1,1,1)
p <- inv.mlogit(x)
x2 <- mlogit(p)

```

`normalize`*Normalize*

Description

Scale a numeric vector by the sum of its elements

Usage

```
normalize(x)
```

Arguments

`x` A numeric vector

Value

$x / \text{sum}(x)$

Examples

```
x <- c(1,1,1,1)
normalize(x)
```

`Numerical MLE`*Numerical MLE*

Description

Numerical maximum likelihood estimation for Mixture Link Binomial and Poisson models.

Usage

```
mle.mixlink.binom(y, m, J, extra.tx = null.tx, var.names = NULL,
  phi.init = NULL)
```

```
mle.mixlink.binom.x(y, m, X, J, extra.tx = null.tx, var.names = NULL,
  phi.init = NULL, invlink.mean = plogis)
```

```
mle.mixlink.pois(y, J, extra.tx = null.tx, var.names = NULL,
  phi.init = NULL)
```

```
mle.mixlink.pois.x(y, X, J, extra.tx = null.tx, var.names = NULL,
  phi.init = NULL, invlink.mean = exp)
```

Arguments

<code>y</code>	Argument of pdf or cdf.
<code>m</code>	Number of success/failure trials.
<code>J</code>	Number of mixture components to use.
<code>extra.tx</code>	If additional functions of θ are to be estimated, they can be specified here. The default <code>null.tx</code> indicates that no extra functions are desired.
<code>var.names</code>	A vector of strings to use for parameter names. The default (NULL) indicates to leave the names at some defaults.
<code>phi.init</code>	Initial value of the unconstrained ϕ parameters. Internally, a transformation θ is applied to ϕ to obtain parameters in the correct space. The default (NULL) selects a default initial value.
<code>X</code>	Design matrix for regression case.
<code>invlink.mean</code>	The inverse link function for the mean. Default is <code>plogis</code> for Binomial and <code>exp</code> for Poisson.

Value

A list with MLE results. Can be accessed through the functions `confint`, `print`, `summary`, `coef`, `logLik` AIC, BIC, and `vcov`.

References

Andrew M. Raim, Nagaraj K. Neerchal, and Jorge G. Morel. An Extension of Generalized Linear Models to Finite Mixture Outcomes. arXiv preprint: 1612.03302

Examples

```
## Not run:
n <- 400
mean.true <- rep(20, n)
Pi.true <- c(1/5, 4/5)
kappa.true <- 2
y <- r.mixlink.pois(n, mean.true, Pi.true, kappa.true)

mle.out <- mle.mixlink.pois(y, J = 2)
coef(mle.out)
print(mle.out)
confint(mle.out)

## End(Not run)
```

Randomized quantile residuals

Randomized quantile residuals for Mixture Link

Description

Compute randomized quantile residuals for the Mixture Link Binomial and Mixture Link Poisson distributions.

Usage

```
rqres.mixlink.binom(y, m, mean, Pi, kappa)
```

```
rqres.mixlink.pois(y, mean, Pi, kappa)
```

Arguments

y	The observations
m	Number of success/failure trials
mean	Estimate for parameter ϑ of distribution
Pi	Estimate for parameter π of distribution
kappa	Estimate for parameter κ of distribution

Value

Vector of residuals

References

Peter K. Dunn and Gordon K. Smyth. Randomized quantile residuals. *Journal of Computational and Graphical Statistics*, 5(3):236-244, 1996.

Examples

```
n <- 400
mean.true <- rep(20, n)
Pi.true <- c(1/4, 3/4)
kappa.true <- 1.5
y <- r.mixlink.pois(n, mean.true, Pi.true, kappa.true)
r <- rqres.mixlink.pois(y, mean.true, Pi.true, kappa.true)
qqnorm(r); qqline(r)
```

S3 methods for gibbs.mixlink output objects
S3 methods for gibbs.mixlink output objects

Description

S3 methods for gibbs.mixlink output objects

Usage

```
## S3 method for class 'gibbs.mixlink'  
summary(object, ...)  
  
## S3 method for class 'gibbs.mixlink'  
print(x, ...)  
  
## S3 method for class 'gibbs.mixlink'  
DIC(object, ...)  
  
## S3 method for class 'gibbs.mixlink'  
residuals(object, ...)
```

Arguments

object	An gibbs.mixlink object from gibbs.mixlink.
...	Not currently used.
x	Same argument type as object.

Details

DIC returns the Deviance Information Criterion computed from MCMC draws. residuals returns randomized quantile residuals averaged over MCMC draws.

S3 methods for mle.fit objects
S3 methods for mle.fit objects

Description

S3 methods for mle.fit objects

Usage

```
## S3 method for class 'mle.fit'
confint(object, parm = NULL, level = 0.95, ...)

## S3 method for class 'mle.fit'
print(x, ...)

## S3 method for class 'mle.fit'
summary(object, ...)

## S3 method for class 'mle.fit'
coef(object, ...)

## S3 method for class 'mle.fit'
logLik(object, ...)

## S3 method for class 'mle.fit'
AIC(object, ..., k = 2)

## S3 method for class 'mle.fit'
BIC(object, ...)

## S3 method for class 'mle.fit'
vcov(object, ...)
```

Arguments

object	An <i>mle.fit</i> object from <i>fit.mle</i> .
parm	Not currently used.
level	Desired confidence level.
...	Not currently used.
x	An <i>mle.fit</i> object from <i>fit.mle</i> .
k	Penalty term is kq where $q = \dim \theta$. $k = 2$ is the default for AIC calculation.

S3 methods for *mle.fit.ci* objects

*Print summary of an *mle.fit.ci* object*

Description

See generic print function for usage.

Usage

```
## S3 method for class 'mle.fit.ci'
print(x, ...)
```

Arguments

x	An <code>mle.fit.ci</code> object from <code>confint.fit.mle</code> .
...	Not currently used.

S3 methods for *rwmetrop* output objects

*S3 methods for *rwmetrop* output objects*

Description

S3 methods for *rwmetrop* output objects

Usage

```
## S3 method for class 'rwmetrop.mixlink.binomial'
summary(object, ...)
```

```
## S3 method for class 'rwmetrop.mixlink.binomial'
print(x, ...)
```

```
## S3 method for class 'rwmetrop.mixlink.binomial'
DIC(object, ...)
```

```
## S3 method for class 'rwmetrop.mixlink.binomial'
residuals(object, ...)
```

```
## S3 method for class 'rwmetrop.mixlink.poisson'
summary(object, ...)
```

```
## S3 method for class 'rwmetrop.mixlink.poisson'
print(x, ...)
```

```
## S3 method for class 'rwmetrop.mixlink.poisson'
DIC(object, ...)
```

```
## S3 method for class 'rwmetrop.mixlink.poisson'
residuals(object, ...)
```

Arguments

object	An <code>rwmetrop.mixlink.binomial</code> object from <code>rwmetrop.mixlink.binomial</code> or an <code>rwmetrop.mixlink.poisson</code> object from <code>rwmetrop.mixlink.poisson</code> .
...	Not currently used.
x	Same argument type as <code>object</code> .

Details

DIC returns the Deviance Information Criterion computed from MCMC draws. residuals returns randomized quantile residuals averaged over MCMC draws.

Index

- AIC.mle.fit (S3 methods for mle.fit objects), [14](#)
- BIC.mle.fit (S3 methods for mle.fit objects), [14](#)
- coef.mle.fit (S3 methods for mle.fit objects), [14](#)
- confint.mle.fit (S3 methods for mle.fit objects), [14](#)
- d.mixlink.binom (Mixture Link Binomial Distribution), [8](#)
- d.mixlink.pois (Mixture Link Poisson Distribution), [9](#)
- DIC, [2](#)
- DIC.gibbs.mixlink (S3 methods for gibbs.mixlink output objects), [14](#)
- DIC.rwmetrop.mixlink.binomial (S3 methods for rwmetrop output objects), [16](#)
- DIC.rwmetrop.mixlink.poisson (S3 methods for rwmetrop output objects), [16](#)
- Find Vertices, [2](#)
- find.vertices.nonneg (Find Vertices), [2](#)
- find.vertices.prob (Find Vertices), [2](#)
- gibbs.mixlink (Metropolis-within-Gibbs Sampler), [5](#)
- inv.mlogit (mlogit), [10](#)
- logLik.mle.fit (S3 methods for mle.fit objects), [14](#)
- Metropolis-Hastings Sampler, [3](#)
- Metropolis-within-Gibbs Sampler, [5](#)
- Mixture Link Binomial Distribution, [8](#)
- Mixture Link Poisson Distribution, [9](#)
- mle.mixlink.binom (Numerical MLE), [11](#)
- mle.mixlink.pois (Numerical MLE), [11](#)
- mlogit, [10](#)
- normalize, [11](#)
- Numerical MLE, [11](#)
- p.mixlink.binom (Mixture Link Binomial Distribution), [8](#)
- p.mixlink.pois (Mixture Link Poisson Distribution), [9](#)
- print.gibbs.mixlink (S3 methods for gibbs.mixlink output objects), [14](#)
- print.mle.fit (S3 methods for mle.fit objects), [14](#)
- print.mle.fit.ci (S3 methods for mle.fit.ci objects), [15](#)
- print.rwmetrop.mixlink.binomial (S3 methods for rwmetrop output objects), [16](#)
- print.rwmetrop.mixlink.poisson (S3 methods for rwmetrop output objects), [16](#)
- r.mixlink.binom (Mixture Link Binomial Distribution), [8](#)
- r.mixlink.pois (Mixture Link Poisson Distribution), [9](#)
- Randomized quantile residuals, [13](#)
- residuals.gibbs.mixlink (S3 methods for gibbs.mixlink output objects), [14](#)
- residuals.rwmetrop.mixlink.binomial (S3 methods for rwmetrop output objects), [16](#)
- residuals.rwmetrop.mixlink.poisson (S3 methods for rwmetrop output objects), [16](#)

rqres.mixlink.binom (Randomized
quantile residuals), [13](#)

rqres.mixlink.pois (Randomized
quantile residuals), [13](#)

rwmetrop.mixlink.binomial
(Metropolis-Hastings Sampler),
[3](#)

rwmetrop.mixlink.poisson
(Metropolis-Hastings Sampler),
[3](#)

S3 methods for gibbs.mixlink output
objects, [14](#)

S3 methods for mle.fit objects, [14](#)

S3 methods for mle.fit.ci objects, [15](#)

S3 methods for rwmetrop output
objects, [16](#)

summary.gibbs.mixlink (S3 methods for
gibbs.mixlink output objects),
[14](#)

summary.mle.fit (S3 methods for
mle.fit objects), [14](#)

summary.rwmetrop.mixlink.binomial (S3
methods for rwmetrop output
objects), [16](#)

summary.rwmetrop.mixlink.poisson (S3
methods for rwmetrop output
objects), [16](#)

vcov.mle.fit (S3 methods for mle.fit
objects), [14](#)