

Package: mederrRank (via r-universe)

November 5, 2024

Title Bayesian Methods for Identifying the Most Harmful Medication Errors

Version 0.1.0

Date 2023-09-05

Description Two distinct but related statistical approaches to the problem of identifying the combinations of medication error characteristics that are more likely to result in harm are implemented in this package: 1) a Bayesian hierarchical model with optimal Bayesian ranking on the log odds of harm, and 2) an empirical Bayes model that estimates the ratio of the observed count of harm to the count that would be expected if error characteristics and harm were independent. In addition, for the Bayesian hierarchical model, the package provides functions to assess the sensitivity of results to different specifications of the random effects distributions.

Author Sergio Venturini, Jessica Myers

Maintainer Sergio Venturini <sergio.venturini@unicatt.it>

Depends BB, methods, numDeriv, utils

Imports graphics, stats

License GPL (>= 2) | file LICENSE

LazyLoad yes

NeedsCompilation no

Repository CRAN

Date/Publication 2023-09-05 15:30:02 UTC

Contents

mederrRank-package	2
bayes.rank	3
bhm.constr.resamp	4
bhm.mcmc	5
bhm.resample	7

dmixnegbinom	9
dnegbinom	10
dst	11
EBGM	12
lIDiffD	14
lIDiffT	15
logp	16
logunpost	17
mederrData-class	18
mederrFit-class	19
mederrResample-class	20
MEDMARX	21
mixnegbinom.em	22
mixnegbinom.loglik	24
mixnegbinom.score	25
negbinom.em	26
negbinom.loglik	27
negbinom.score	28
p.value	29
plot-methods	30
post.rep	31
rmixnegbinom	32
simdata	33
summary-methods	34

Index 35

mederrRank-package *Bayesian Methods for Identifying the Most Harmful Medication Errors*

Description

Two distinct but related statistical approaches to the problem of identifying the combinations of medication error characteristics that are more likely to result in harm are implemented in this package: 1) a Bayesian hierarchical model with optimal Bayesian ranking on the log odds of harm, and 2) an empirical Bayes model that estimates the ratio of the observed count of harm to the count that would be expected if error characteristics and harm were independent. In addition, for the Bayesian hierarchical model, the package provides functions to assess the sensitivity of results to different specifications of the random effects distributions.

Details

The package is loaded with the usual `library(mederrRank)` command. The most important functions are `bhm.mcmc`, `bhm.resample` and `mixnegbinom.em`.

Author(s)

Sergio Venturini, Jessica Myers

Maintainer: Sergio Venturini <sergio.venturini@unicatt.it>

References

Myers, J. A., Venturini, S., Dominici, F. and Morlock, L. (2011), "Random Effects Models for Identifying the Most Harmful Medication Errors in a Large, Voluntary Reporting Database". Technical Report.

See Also

[bayes.rank](#), [bhm.mcmc](#), [bhm.resample](#), [mixnegbinom.em](#).

bayes.rank

Optimal Bayesian Ranking

Description

This function estimates the ranks of the log odds of harm of the various medication error profiles as described in Myers et al. (2011).

Usage

```
bayes.rank(model)
```

Arguments

model a [mederrFit](#) object.

Details

Using the posterior samples of the θ_i , the function estimates the ranks of the log odds of harm of the various error profiles. Optimal Bayesian ranking gives estimates of rank for profile i as

$$\hat{R}_i = \sum_{k=1}^n \hat{P}(\theta_k \leq \theta_i | \mathbf{y}, \mathbf{N}),$$

where $\hat{P}(\theta_k \leq \theta_i | \mathbf{y}, \mathbf{N})$ is the posterior probability that $\theta_k \leq \theta_i$.

Value

bayes.rank returns the numerical vector of Optimal Bayesian ranks for the chosen [mederrFit](#) model (see the references for the details).

Author(s)

Sergio Venturini <sergio.venturini@unicatt.it>
 Jessica A. Myers <jmyers6@partners.org>

References

Myers, J. A., Venturini, S., Dominici, F. and Morlock, L. (2011), "Random Effects Models for Identifying the Most Harmful Medication Errors in a Large, Voluntary Reporting Database". Technical Report.

See Also

[bhm.constr.resamp](#), [bhm.mcmc](#).

Examples

```
## Not run:
data("simdata", package = "mederrRank")
summary(simdata)

fit <- bhm.mcmc(simdata, nsim = 1000, burnin = 500, scale.factor = 1.1)
ranks <- bayes.rank(fit)

summary(ranks)

## End(Not run)
```

bhm.constr.resamp	<i>Markov Chain Monte Carlo Estimation (Step 2) of the Bayesian Hierarchical Model for Identifying the Most Harmful Medication Errors</i>
-------------------	---

Description

This function represents the "constructor" function for the resampling procedure used in this package. [bhm.resample](#) calculates the importance ratios, and performs the sampling, and then this function constructs the resampled model based on that information.

Usage

```
bhm.constr.resamp(model, resample, k, eta)
```

Arguments

model	an object of class " mederrFit ".
resample	an object of class " mederrResample ".
k	k (number of degrees of freedom) value to use in the resampling procedure.
eta	η (skewing parameter) value to use in the resampling procedure.

Details

Deviations from the normal, i.e. ($k = \infty, \eta = 1$), random effects distribution using a different pair of k and η values are considered. The methodology implemented here is the importance link function resampling approach introduced by MacEachern and Peruggia (2000): based on the ($k = \infty, \eta = 1$) chain, new posterior samples under a new set of (k, η) values is obtained.

Value

bhm.constr.resamp returns an object of the class "[mederrFit](#)".

Author(s)

Sergio Venturini <sergio.venturini@unicatt.it>,
 Jessica A. Myers <jmyers6@partners.org>

References

MacEachern, S. and Peruggia, M. (2000), "Importance Link Function Estimation for Markov Chain Monte Carlo Methods", *Journal of Computational and Graphical Statistics*, 9, 99-121.

Myers, J. A., Venturini, S., Dominici, F. and Morlock, L. (2011), "Random Effects Models for Identifying the Most Harmful Medication Errors in a Large, Voluntary Reporting Database". Technical Report.

See Also

[bhm.mcmc](#), [bhm.resample](#), [mederrData](#), [mederrFit](#).

Examples

```
## Not run:
data("simdata", package = "mederrRank")
summary(simdata)

fit <- bhm.mcmc(simdata, nsim = 1000, burnin = 500, scale.factor = 1.1)
resamp <- bhm.resample(fit, simdata, p.resample = .1,
  k = c(3, 6, 10, 30, 60, Inf), eta = c(.5, .8, 1, 1.25, 2))
fit2 <- bhm.constr.resamp(fit, resamp, k = 3, eta = .8)
plot(fit, fit2, simdata)

theta0 <- c(10, 6, 100, 100, .1)
ans <- mixnegbinom.em(simdata, theta0, 50000, 0.01, se = TRUE,
  stratified = TRUE)

summary(fit2, ans, simdata)

## End(Not run)
```

bhm.mcmc

Markov Chain Monte Carlo Estimation (Step 1) of the Bayesian Hierarchical Model for Identifying the Most Harmful Medication Errors

Description

This function implements the Markov Chain Monte Carlo estimation methodology for the Bayesian hierarchical model described in Myers et al. (2011).

Usage

```
bhm.mcmc(dat, nsim = 2000, burnin = 500, scale.factor = 1,
  adaptive.int = 100, adaptive.max = 1000, prior = NULL,
  init = NULL, tuneD = NULL, tuneT = NULL)
```

Arguments

dat	an object of class "mederrData".
nsim	number of iterations.
burnin	number of burn-in iterations.
scale.factor	scale factor of the random effects proposal distribution.
adaptive.int	iteration interval at which the standard error of the random effects proposal distribution is updated.
adaptive.max	last iteration at which the standard error of the random effects proposal distribution is updated.
prior	an optional list of the hyperparameters values; see the Details section below.
init	an optional list of initial values for the model parameters; see the Details section below.
tuneD	an optional vector of the δ_j proposal distribution variances.
tuneT	an optional vector of the θ_i proposal distribution variances.

Details

The Bayesian hierarchical model (with crossed random effects) implemented here for identifying the medication error profiles with the largest log odds of harm is

$$\begin{aligned}
 y_{ij} | N_{ij}, p_{ij} &\sim \text{Bin}(N_{ij}, p_{ij}) \\
 \text{logit}(p_{ij}) &= \gamma + \theta_i + \delta_j \\
 \theta_i | \sigma, \eta, k &\sim \text{St}(0, \sigma, k, \eta), \quad i = 1, \dots, n \\
 \delta_j | \tau^2 &\sim N(0, \tau^2), \quad j = 1, \dots, J \\
 \gamma &\sim N(g, G) \\
 \sigma^2 &\sim \text{IG}(a_1, b_1) \\
 \tau^2 &\sim \text{IG}(a_2, b_2) \\
 k &\sim \text{Unif}(0, \infty) \\
 \eta &\sim \text{Unif}(0, \infty),
 \end{aligned}$$

where N_{ij} denotes the number of times that the error profile i is cited on a report from hospital j and y_{ij} is the corresponding number of times that profile i in hospital j was reported with harm. This function implements the first model estimation step in which the values $k = \infty$ and $k = 1$, i.e. a symmetric normal distribution, is forced for the error profiles' random effects. A sample from the joint posterior distribution of all other parameters via Markov Chain Monte Carlo with adaptive Metropolis steps for each set of random effects is obtained. For more details see Myers et al. (2011).

Value

bhm.mcmc returns an object of the class "[mederrFit](#)".

Author(s)

Sergio Venturini <sergio.venturini@unicatt.it>,
 Jessica A. Myers <jmyers6@partners.org>

References

Myers, J. A., Venturini, S., Dominici, F. and Morlock, L. (2011), "Random Effects Models for Identifying the Most Harmful Medication Errors in a Large, Voluntary Reporting Database". Technical Report.

See Also

[bhm.resample](#), [mederrData](#), [mederrFit](#).

Examples

```
## Not run:
data("simdata", package = "mederrRank")
summary(simdata)

fit <- bhm.mcmc(simdata, nsim = 1000, burnin = 500, scale.factor = 1.1)
resamp <- bhm.resample(fit, simdata, p.resample = .1,
  k = c(3, 6, 10, 30, 60, Inf), eta = c(.5, .8, 1, 1.25, 2))
fit2 <- bhm.constr.resamp(fit, resamp, k = 3, eta = .8)
plot(fit, fit2, simdata)

theta0 <- c(10, 6, 100, 100, .1)
ans <- mixnegbinom.em(simdata, theta0, 50000, 0.01, se = TRUE,
  stratified = TRUE)

summary(fit2, ans, simdata)
## End(Not run)
```

bhm.resample

Resampling Transformation for the Markov Chain Monte Carlo Estimation Simulation of the Bayesian Hierarchical Model for Identifying the Most Harmful Medication Errors

Description

This function implements the transformation needed to apply the importance link function resampling methodology based on the Markov Chain Monte Carlo simulations obtained with the [bhm.mcmc](#) command (see the References).

Usage

```
bhm.resample(model, dat, p.resample = 0.1, k, eta)
```

Arguments

model	an object of class "mederrFit".
dat	an object of class "mederrData".
p.resample	proportion of simulations resampled from the model argument.
k	required vector of k values to be used in the resampling process.
eta	required vector of η values to be used in the resampling process.

Value

bhm.resample returns an object of the class "mederrResample".

Author(s)

Sergio Venturini <sergio.venturini@unicatt.it>
 Jessica A. Myers <jmyers6@partners.org>

References

MacEachern, S. and Peruggia, M. (2000), "Importance Link Function Estimation for Markov Chain Monte Carlo Methods", *Journal of Computational and Graphical Statistics*, 9, 99-121.

Myers, J. A., Venturini, S., Dominici, F. and Morlock, L. (2011), "Random Effects Models for Identifying the Most Harmful Medication Errors in a Large, Voluntary Reporting Database". Technical Report.

See Also

[mederrData](#), [mederrFit](#), [bhm.mcmc](#).

Examples

```
## Not run:
data("simdata", package = "mederrRank")
summary(simdata)

fit <- bhm.mcmc(simdata, nsim = 1000, burnin = 500, scale.factor = 1.1)
resamp <- bhm.resample(fit, simdata, p.resample = .1,
  k = c(3, 6, 10, 30, 60, Inf), eta = c(.5, .8, 1, 1.25, 2))
fit2 <- bhm.constr.resamp(fit, resamp, k = 3, eta = .8)
plot(fit, fit2, simdata)

theta0 <- c(10, 6, 100, 100, .1)
ans <- mixnegbinom.em(simdata, theta0, 50000, 0.01, se = TRUE,
  stratified = TRUE)

summary(fit2, ans, simdata)
```



```
## End(Not run)
```

dmixnegbinom *The Negative Binomial Mixture Distribution*

Description

Density function for a mixture of two negative binomial distributions.

Usage

```
dmixnegbinom(x, theta, E, log.p = FALSE)
```

Arguments

`x` vector of (non-negative integer) quantiles.
`theta` vector of parameters for the negative binomial distribution mixture.
`E` vector of (non-negative integer) expected counts.
`log.p` logical; if TRUE, probabilities `p` are given as `log(p)`.

Details

The mixture of two negative binomial distributions has density

$$P(N = x) = \text{theta}[5]f(x; \text{theta}[1], \text{theta}[2], E) + (1 - \text{theta}[5])f(x; \text{theta}[3], \text{theta}[4], E),$$

where

$$f(x; \alpha, \beta, E) = \frac{\Gamma(\alpha + x)}{\Gamma(\alpha)x!} \frac{1}{(1 + \beta/E)^x} \frac{1}{(1 + E/\beta)^\alpha}$$

for $x = 0, 1, \dots, \alpha$, $\alpha, \beta, E > 0$ and $0 < \text{theta}[5] \leq 1$. The mixture of two negative binomial distributions represents the marginal distribution of the counts N coming from Poisson data with parameter λ and a mixture of two gamma distributions as its prior. For details see the paper by Dumouchel (1999).

Value

dmixnegbinom gives the density corresponding to the `E` and `theta` values provided.

Author(s)

Sergio Venturini <sergio.venturini@unicatt.it>,
 Jessica A. Myers <jmyers6@partners.org>

References

DuMouchel W. (1999), "Bayesian Data Mining in Large Frequency Tables, with an Application to the FDA Spontaneous Reporting System". *The American Statistician*, 53, 177-190.

Myers, J. A., Venturini, S., Dominici, F. and Morlock, L. (2011), "Random Effects Models for Identifying the Most Harmful Medication Errors in a Large, Voluntary Reporting Database". Technical Report.

See Also

[dnbinom](#), [rmixnegbinom](#).

Examples

```
## Not run:
data("simdata", package = "mederrRank")
ni <- simdata@numi
theta0 <- c(10, 6, 100, 100, .1)
ans <- mixnegbinom.em(simdata, theta0, 50000, 0.01,
  se = FALSE, stratified = TRUE)
theta <- ans$theta.hat
N.E <- cbind(ans$N[1:ni], ans$E[1:ni])[sort(ans$N[1:ni], index.return = TRUE)$ix, ]
N.ix <- match(unique(N.E[, 1]), N.E[, 1])
N <- N.E[N.ix, 1]
E <- N.E[N.ix, 2]
dens <- dmixnegbinom(N, theta, E)
hist(N.E[, 1], breaks = 40, freq = FALSE)
points(N, dens)

## End(Not run)
```

dnegbinom

The Negative Binomial Distribution

Description

Density function for the negative binomial distribution with parameters alpha and prob.

Usage

```
dnegbinom(x, alpha, prob, log.p = FALSE)
```

Arguments

x	vector of (non-negative integer) quantiles.
alpha	target for number of successful trials. Must be strictly positive, need not be integer.
prob	probability of success in each trial. $0 < \text{prob} \leq 1$.
log.p	logical; if TRUE, probabilities p are given as log(p).

Details

The negative binomial distribution with parameters $\alpha = \alpha$ and $\text{prob} = p$ has density

$$\frac{\Gamma(x + \alpha)}{\Gamma(\alpha)x!} p^\alpha (1 - p)^x$$

for $x = 0, 1, \dots, \alpha > 0$ and $0 < p \leq 1$. This represents the number of failures which occur in a sequence of Bernoulli trials before a target number of successes is reached.

Value

`dnegbinom` gives the density corresponding to the `alpha` and `prob` values provided.

Author(s)

Sergio Venturini <sergio.venturini@unicatt.it>,
Jessica A. Myers <jmyers6@partners.org>

References

DuMouchel W. (1999), "Bayesian Data Mining in Large Frequency Tables, with an Application to the FDA Spontaneous Reporting System". *The American Statistician*, 53, 177-190.

Myers, J. A., Venturini, S., Dominici, F. and Morlock, L. (2011), "Random Effects Models for Identifying the Most Harmful Medication Errors in a Large, Voluntary Reporting Database". Technical Report.

See Also

[dmixnegbinom](#), [dnbinom](#).

 dst

The Skewed Student t Distribution

Description

Density function for the skewed t distribution with `k` degrees of freedom, scale parameter `sigma` and skewness `eta`.

Usage

```
dst(x, sigma, k, eta)
```

Arguments

<code>x</code>	vector of quantiles.
<code>sigma</code>	scale parameter (> 0).
<code>k</code>	degrees of freedom (> 0 , maybe non-integer). <code>df = Inf</code> is allowed.
<code>eta</code>	skewness parameter (> 0).

Details

This distribution is based on introducing skewing into the symmetric scaled t distribution, as described in Fernandez and Steel (1998). The parameters characterizing the center (here set at 0) and the spread (σ) refer to the mean and standard deviation of the underlying symmetric distribution. In the skewed t distribution, the centrality parameter defines the mode of the distribution, but it is no longer either the mean or the median. Similarly, in the skewed t distribution, σ still characterizes the spread, but it can no longer be interpreted directly as the standard deviation of the distribution.

Value

`dst` gives the density corresponding to the σ , k and η values provided.

Author(s)

Sergio Venturini <sergio.venturini@unicatt.it>
Jessica A. Myers <jmyers6@partners.org>

References

- Fernandez, C. and Steel, M. (1998), "On Bayesian Modeling of Fat Tails and Skewness". *Journal of the American Statistical Association*, 93, 359-371.
- Lee, K. and Thompson, S. (2008), "Flexible Parametric Models for Random-Effects Distributions". *Statistics in Medicine*, 27, 418-434.
- Myers, J. A., Venturini, S., Dominici, F. and Morlock, L. (2011), "Random Effects Models for Identifying the Most Harmful Medication Errors in a Large, Voluntary Reporting Database". Technical Report.

See Also

[dt](#).

EBGM

Geometric Mean of the Relative Risk Empirical Bayes Posterior Distribution

Description

This function computes the geometric mean of the empirical Bayes posterior distribution for the observed vs. expected count relative risk.

Usage

```
EBGM(eb.result)
```

Arguments

`eb.result` output of the `mixnegbinom.em` or `negbinom.em` commands.

Details

For further details see DuMouchel (1999).

Value

EBGM returns the vector of geometric means.

Author(s)

Sergio Venturini <sergio.venturini@unicatt.it>,
Jessica A. Myers <jmyers6@partners.org>

References

DuMouchel W. (1999), "Bayesian Data Mining in Large Frequency Tables, with an Application to the FDA Spontaneous Reporting System". *The American Statistician*, 53, 177-190.

Myers, J. A., Venturini, S., Dominici, F. and Morlock, L. (2011), "Random Effects Models for Identifying the Most Harmful Medication Errors in a Large, Voluntary Reporting Database". Technical Report.

See Also

[mixnegbinom.em](#), [negbinom.em](#).

Examples

```
## Not run:
data("simdata", package = "mederrRank")
summary(simdata)

fit <- bhm.mcmc(simdata, nsim = 1000, burnin = 500, scale.factor = 1.1)
resamp <- bhm.resample(fit, simdata, p.resample = .1,
  k = c(3, 6, 10, 30, 60, Inf), eta = c(.5, .8, 1, 1.25, 2))
fit2 <- bhm.constr.resamp(fit, resamp, k = 3, eta = .8)

theta0 <- c(10, 6, 100, 100, .1)
ans <- mixnegbinom.em(simdata, theta0, 50000, 0.01,
  se = FALSE, stratified = TRUE)

ni <- simdata@numi
rank(EBGM(ans)[1:ni])
summary(fit2, ans, simdata)
## End(Not run)
```

`l1DiffD`*Log-Likelihood Difference for the δ_j Parameters*

Description

This function computes the log-likelihood difference for the candidate δ_j random effects. It is a helper function and not meant to be used on its own.

Usage

```
l1DiffD(dat, deltaj, cand, thetai, gamma, tau2)
```

Arguments

<code>dat</code>	data frame containing the observed sample counts.
<code>deltaj</code>	vector of previous accepted values for the δ_j random effects.
<code>cand</code>	vector of candidate values for the δ_j random effects.
<code>thetai</code>	vector of previous accepted values for the θ_i random effects.
<code>gamma</code>	last sampled value for the γ parameter.
<code>tau2</code>	last sampled value for the τ^2 parameter.

Details

For further details see Myers et al. (2011).

Value

`l1DiffD` returns the vector of log-likelihood differences.

Author(s)

Sergio Venturini <sergio.venturini@unicatt.it>,
Jessica A. Myers <jmyers6@partners.org>

References

Myers, J. A., Venturini, S., Dominici, F. and Morlock, L. (2011), "Random Effects Models for Identifying the Most Harmful Medication Errors in a Large, Voluntary Reporting Database". Technical Report.

See Also

[bhm.constr.resamp](#), [bhm.mcmc](#), [bhm.resample](#).

`l1DiffT`*Log-Likelihood Difference for the θ_i Parameters*

Description

This function computes the log-likelihood difference for the candidate θ_i random effects. It is a helper function and not meant to be used on its own.

Usage

```
l1DiffT(dat, thetai, cand, deltax, gamma, sigma2)
```

Arguments

<code>dat</code>	data frame containing the observed sample counts.
<code>thetai</code>	vector of previous accepted values for the θ_i random effects.
<code>cand</code>	vector of candidate values for the δ_j random effects.
<code>deltax</code>	vector of previous accepted values for the δ_j random effects.
<code>gamma</code>	last sampled value for the γ parameter.
<code>sigma2</code>	last sampled value for the σ^2 parameter.

Details

For further details see Myers et al. (2011).

Value

`l1DiffT` returns the vector of log-likelihood differences.

Author(s)

Sergio Venturini <sergio.venturini@unicatt.it>,
Jessica A. Myers <jmyers6@partners.org>

References

Myers, J. A., Venturini, S., Dominici, F. and Morlock, L. (2011), "Random Effects Models for Identifying the Most Harmful Medication Errors in a Large, Voluntary Reporting Database". Technical Report.

See Also

[bhm.constr.resamp](#), [bhm.mcmc](#), [bhm.resample](#).

logp *Negative Log-Posterior Function of the Bayesian Hierarchical Model for Identifying the Most Harmful Medication Errors*

Description

This function computes the negative log-posterior distribution of the Bayesian hierarchical model described in Myers et al (2011). It is a helper function and not meant to be used on its own.

Usage

```
logp(theta, deltaj, sigma2, i, k, eta, dat)
```

Arguments

theta	value of the error profile random effect at which the log.posterior distribution is calculated.
deltaj	vector of hospital random effect values.
sigma2	scale parameter (> 0).
i	error profile index for which the calculate of the log.posterior distribution is needed.
k	degrees of freedom (> 0 , maybe non-integer). $df = Inf$ is allowed.
eta	skewness parameter (> 0).
dat	an object of class "mederrData" .

Details

For further details see Myers et al. (2011).

Value

logp returns a vector of log-posterior values.

Author(s)

Sergio Venturini <sergio.venturini@unicatt.it>
Jessica A. Myers <jmyers6@partners.org>

References

Myers, J. A., Venturini, S., Dominici, F. and Morlock, L. (2011), "Random Effects Models for Identifying the Most Harmful Medication Errors in a Large, Voluntary Reporting Database". Technical Report.

See Also

[bhm.constr.resamp](#), [bhm.mcmc](#), [bhm.resample](#).

logunpost

Unnormalized Marginal Posterior Distributions for k and η

Description

This functions computes the unnormalized marginal posterior distributions for the k and η parameters as described in Myers et al (2011).

Usage

```
logunpost(resample)
```

Arguments

resample an object of class "[mederrResample](#)".

Details

logunpost is used in the plot method for a [mederrResample](#) object.

Value

logunpost returns an array with the posterior distribution values.

Author(s)

Sergio Venturini <sergio.venturini@unicatt.it>,
Jessica A. Myers <jmyers6@partners.org>

References

Myers, J. A., Venturini, S., Dominici, F. and Morlock, L. (2011), "Random Effects Models for Identifying the Most Harmful Medication Errors in a Large, Voluntary Reporting Database". Technical Report.

See Also

[bhm.constr.resamp](#), [bhm.mcmc](#), [bhm.resample](#), [mederrResample](#).

mederrData-class	<i>Class "mederrData". Data Specification for Identifying the Most Harmful MEDication Errors using a Bayesian Hierarchical Model.</i>
------------------	---

Description

This class encapsulates the data specification for a Bayesian Hierarchical Model used to identify the most harmful medication errors as described in Myers et al. (2011).

Objects from the Class

Objects can be created by calls of the form `new("mederrData", data)`, where the `data` argument has to be a matrix or a data frame object that contains the following (*numeric*) information for each error profile/hospital combination:

1. the number of times (y) that profile i in hospital j was reported with harm;
2. the total number of times (N) that the error profile i is cited on a report from hospital j ,
3. the error profile i identification code,
4. the hospital j identification code.

Slots

`data`: Object of class "data.frame"; data in the standard `data.frame` form.
`size`: Object of class "numeric"; total number of observations in the data set.
`numi`: Object of class "numeric"; number of error profiles available in the data set.
`numj`: Object of class "numeric"; number of hospitals available in the data set.

Methods

plot signature(`x = "mederrData"`, `y = "missing"`): Provides a pictorial representation for a sample of error profiles reported by some hospitals.

summary signature(`object = "mederrData"`): Summarizes information about an `mederrData` object.

Author(s)

Sergio Venturini <sergio.venturini@unicatt.it>,
Jessica A. Myers <jmyers6@partners.org>

References

Myers, J. A., Venturini, S., Dominici, F. and Morlock, L. (2011), "Random Effects Models for Identifying the Most Harmful Medication Errors in a Large, Voluntary Reporting Database". Technical Report.

See Also

[bayes.rank](#), [bhm.mcmc](#), [bhm.resample](#), [mixnegbinom.em](#).

Examples

```
ng <- 50
i <- rep(1:ng, ng)
j <- rep(1:ng, each = ng)
N <- rpois(ng^2, 3 + .05*i - .01*j) + 1

theta_i <- rgamma(ng, 5, 5) - 4/5
delta_j <- rnorm(ng, 0, .2)
logit <- -3 + theta_i[i] + delta_j[j]
y <- rbinom(ng^2, N, exp(logit)/(1 + exp(logit)))

simdata <- new("mederrData", data = cbind(y, N, i, j))
```

mederrFit-class	<i>Class "mederrFit". Simulated Monte Carlo Chains (Step 1) for the Bayesian Hierarchical Model Used to Identify the Most Harmful Medication Errors.</i>
-----------------	--

Description

This class encapsulates the simulated Monte Carlo chains for the Bayesian Hierarchical Model as described in Myers et al. (2011) forcing a symmetric normal distribution on the $\theta_i, i = 1, \dots, n$.

Objects from the Class

Objects can be created by calls of the form `new("mederrFit", thetai, deltaj, gamma, sigma2, tau2, p.acc.i, p.acc.j, tune.theta, tune.delta, k, eta)`, but most often as the result of a call to [bhm.mcmc](#) or to [bhm.constr.resamp](#).

Slots

thetai: Object of class "matrix"; simulated chains for the $\theta_i, i = 1, \dots, n$, error profiles random effects; see [bhm.mcmc](#).

deltaj: Object of class "matrix"; simulated chains for the $\delta_j, j = 1, \dots, J$, hospitals random effects; see [bhm.mcmc](#).

gamma: Object of class "numeric"; simulated chain for the γ parameter; see [bhm.mcmc](#).

sigma2: Object of class "numeric"; simulated chain for the σ^2 parameter; see [bhm.mcmc](#).

tau2: Object of class "numeric"; simulated chain for the τ^2 parameter; see [bhm.mcmc](#).

p.acc.i: Object of class "numeric"; acceptance rates for the error profiles random effects.

p.acc.j: Object of class "numeric"; acceptance rates for the hospitals random effects.

tune.theta: Object of class "numeric"; last updated values of the θ_i working variances for the Metropolis step.

`tune.delta`: Object of class "numeric"; last updated values of the δ_j working variances for the Metropolis step.

`k`: Object of class "numeric"; k value used in the simulation.

`eta`: Object of class "numeric"; η value used in the simulation.

Methods

plot signature(`x = "mederrFit"`, `y = "mederrFit"`): Provides a graphical representation of the estimates.

summary signature(`object = "mederrFit"`): Summarizes the information regarding the estimates.

Author(s)

Sergio Venturini <sergio.venturini@unicatt.it>,

Jessica A. Myers <jmyers6@partners.org>

References

Myers, J. A., Venturini, S., Dominici, F. and Morlock, L. (2011), "Random Effects Models for Identifying the Most Harmful Medication Errors in a Large, Voluntary Reporting Database". Technical Report.

See Also

[bayes.rank](#), [bhm.constr.resamp](#), [bhm.mcmc](#).

mederrResample-class *Class "mederrResample". Simulated Monte Carlo Chains (Step 2) for the Bayesian Hierarchical Model Used to Identify the Most Harmful Medication Errors.*

Description

This class encapsulates the information needed to resample the Monte Carlo chains for the Bayesian Hierarchical Model as described in Myers et al. (2011) using user defined values for k and η .

Objects from the Class

Objects can be created by calls of the form `new("mederrResample", log.ir, samp, A, t.new, t.old, grd)`, but most often as the result of a call to [bhm.resample](#).

Slots

`log.ir`: Object of class "array"; logarithm of the importance ratio for each pair of (k, η) values.

`samp`: Object of class "array"; resampled MCMC simulation indexes.

`A`: Object of class "array"; transformation ratio for each pair of (k, η) values.

`t.new`: Object of class "array"; θ_i posterior modes using $(k = \infty, \eta = 1)$.

`t.old`: Object of class "numeric"; θ_i posterior modes using user defined (k, η) values.

`grd`: Object of class "list"; grid of required (k, η) values.

Methods

`plot` signature(x = "mederrResample", y = "missing"): : Provides a graphical representation of a `mederrResample` object.

Author(s)

Sergio Venturini <sergio.venturini@unicatt.it>,
Jessica A. Myers <jmyers6@partners.org>

References

Myers, J. A., Venturini, S., Dominici, F. and Morlock, L. (2011), "Random Effects Models for Identifying the Most Harmful Medication Errors in a Large, Voluntary Reporting Database". Technical Report.

See Also

[bayes.rank](#), [bhm.constr.resamp](#), [bhm.mcmc](#).

MEDMARX

Subset of the MEDMARX Data

Description

Subset of the MEDMARX data included for illustrative purposes only in the `mederrRank` package.

Usage

```
data(MEDMARX)
```

Format

An object of class `mederrData`.

Details

The data contained in this object are reproduced by gentle permission of Quantros, Inc., 690 N. McCarthy Blvd., Suite 200, Milpitas, CA 95035.

Author(s)

Sergio Venturini <sergio.venturini@unicatt.it>,
 Jessica A. Myers <jmyers6@partners.org>

References

Myers, J. A., Venturini, S., Dominici, F. and Morlock, L. (2011), "Random Effects Models for Identifying the Most Harmful Medication Errors in a Large, Voluntary Reporting Database". Technical Report.

See Also

[bhm.mcmc](#), [mederrData](#), [mederrFit](#).

Examples

```
data("MEDMARX", package = "mederrRank")
summary(MEDMARX)
plot(MEDMARX, nbins.err = 20, nbins.hosp = 10)
```

mixnegbinom.em

Expectation-Maximization Algorithm for the Mixture of Negative Binomial Distributions

Description

This function provides the empirical Bayes estimates for the parameters θ of a mixture of two negative binomial distributions (see [dmixnegbinom](#)) using an Expectation-Maximization algorithm.

Usage

```
mixnegbinom.em(dat, theta0, maxiter = 50000, toler = 0.01,
  se = TRUE, stratified = FALSE)
```

Arguments

<code>dat</code>	an object of class <code>"mederrData"</code> .
<code>theta0</code>	initial values for the parameters to be optimized over.
<code>maxiter</code>	a positive integer specifying the maximum number of iterations to be performed before the program is terminated.
<code>toler</code>	a positive scalar giving the tolerance at which the change in the log-likelihood is considered close enough to zero to terminate the algorithm.
<code>se</code>	logical; if TRUE the standard errors of the estimates are also returned.
<code>stratified</code>	logical; if TRUE the analysis will be performed by stratifying on the hospitals.

Details

For further details see Myers et al. (2011).

Value

mixnegbinom.em returns a list with components:

theta.hat	The best set of parameters found.
final.err	The last change in the log-likelihood; it has to be smaller than the toler argument.
final.ll	The likelihood value corresponding to theta.hat.
final.score	The log-likelihood score value corresponding to theta.hat.
num.iter	The number of iterations performed to find the proposed solution.
se	Only if argument se is true. A vector of estimates standard errors for the solution found.
N	The vector of observed error profiles counts.
E	The vector of expected error profiles counts.
prior	A character string giving the prior used; for this function is set to "mixgamma", i.e. a mixture of two gamma distributions as in DuMouchel (1999).

Author(s)

Sergio Venturini <sergio.venturini@unicatt.it>,
 Jessica A. Myers <jmyers6@partners.org>

References

DuMouchel W. (1999), "Bayesian Data Mining in Large Frequency Tables, with an Application to the FDA Spontaneous Reporting System". *The American Statistician*, 53, 177-190.

Myers, J. A., Venturini, S., Dominici, F. and Morlock, L. (2011), "Random Effects Models for Identifying the Most Harmful Medication Errors in a Large, Voluntary Reporting Database". Technical Report.

See Also

[dmixnegbinom](#), [EBGM](#), [negbinom.em](#).

Examples

```
## Not run:
data("simdata", package = "mederrRank")
summary(simdata)

fit <- bhm.mcmc(simdata, nsim = 1000, burnin = 500, scale.factor = 1.1)
resamp <- bhm.resample(fit, simdata, p.resample = .1,
  k = c(3, 6, 10, 30, 60, Inf), eta = c(.5, .8, 1, 1.25, 2))
fit2 <- bhm.constr.resamp(fit, resamp, k = 3, eta = .8)
```

```
plot(fit, fit2, simdata)

theta0 <- c(10, 6, 100, 100, .1)
ans <- mixnegbinom.em(simdata, theta0, 50000, 0.01,
  se = TRUE, stratified = TRUE)
ans$theta
ans$se

summary(fit2, ans, simdata)
## End(Not run)
```

mixnegbinom.loglik *Log-Likelihood Function for the Mixture of Negative Binomial Distributions*

Description

This function computes the log-likelihood function for the mixture of two negative binomial distributions as described in [dmixnegbinom](#).

Usage

```
mixnegbinom.loglik(theta, N, E)
```

Arguments

theta	vector of parameter values.
N	vector of observed error profiles counts.
E	vector of expected error profiles counts.

Details

For further details see Myers et al. (2011).

Value

mixnegbinom.loglik returns the log-likelihood value for the negative binomial mixture.

Author(s)

Sergio Venturini <sergio.venturini@unicatt.it>,
Jessica A. Myers <jmyers6@partners.org>

References

DuMouchel W. (1999), "Bayesian Data Mining in Large Frequency Tables, with an Application to the FDA Spontaneous Reporting System". *The American Statistician*, 53, 177-190.

Myers, J. A., Venturini, S., Dominici, F. and Morlock, L. (2011), "Random Effects Models for Identifying the Most Harmful Medication Errors in a Large, Voluntary Reporting Database". Technical Report.

See Also

[dmixnegbinom](#), [mixnegbinom.em](#), [mixnegbinom.score](#).

mixnegbinom.score	<i>Log-Likelihood Score Function for the Mixture of Negative Binomial Distributions</i>
-------------------	---

Description

This function computes the log-likelihood score for the mixture of two negative binomial distributions as described in [dmixnegbinom](#).

Usage

```
mixnegbinom.score(theta, N, E)
```

Arguments

theta	vector of parameter values.
N	vector of observed error profiles counts.
E	vector of expected error profiles counts.

Details

For further details see Myers et al. (2011).

Value

mixnegbinom.score returns the vector of log-likelihood score values for the negative binomial mixture.

Author(s)

Sergio Venturini <sergio.venturini@unicatt.it>,
Jessica A. Myers <jmyers6@partners.org>

References

DuMouchel W. (1999), "Bayesian Data Mining in Large Frequency Tables, with an Application to the FDA Spontaneous Reporting System". *The American Statistician*, 53, 177-190.

Myers, J. A., Venturini, S., Dominici, F. and Morlock, L. (2011), "Random Effects Models for Identifying the Most Harmful Medication Errors in a Large, Voluntary Reporting Database". Technical Report.

See Also

[dmixnegbinom](#), [mixnegbinom.em](#), [mixnegbinom.loglik](#).

negbinom.em	<i>Expectation-Maximization Algorithm for the Negative Binomial Distribution</i>
-------------	--

Description

This function provides the empirical Bayes estimates for the parameters θ of a negative binomial distribution (see [dnegbinom](#)) using an Expectation-Maximization algorithm.

Usage

```
negbinom.em(dat, theta0, maxiter = 50000, toler = 0.01,
  se = TRUE, stratified = FALSE)
```

Arguments

dat	an object of class "mederrData".
theta0	initial values for the parameters to be optimized over.
maxiter	a positive integer specifying the maximum number of iterations to be performed before the program is terminated.
toler	a positive scalar giving the tolerance at which the scaled gradient is considered close enough to zero to terminate the algorithm.
se	logical; if TRUE the standard errors of the estimates are also returned.
stratified	logical; if TRUE the analysis will be performed by stratifying on the hospitals.

Details

For further details see Myers et al. (2011).

Value

negbinom.em returns a list with components:

theta.hat	The best set of parameters found.
final.err	The last change in the log-likelihood; it has to be smaller than the toler argument.
final.ll	The likelihood value corresponding to theta.hat.
final.score	The log-likelihood score value corresponding to theta.hat.
num.iter	The number of iterations performed to find the proposed solution.
se	Only if argument se is true. A vector of estimates standard errors for the solution found.
N	The vector of observed error profiles counts.
E	The vector of expected error profiles counts.
prior	A character string giving the prior used; for this function is set to "gamma", i.e. a gamma distribution.

Author(s)

Sergio Venturini <sergio.venturini@unicatt.it>,
 Jessica A. Myers <jmyers6@partners.org>

References

DuMouchel W. (1999), "Bayesian Data Mining in Large Frequency Tables, with an Application to the FDA Spontaneous Reporting System". *The American Statistician*, 53, 177-190.

Myers, J. A., Venturini, S., Dominici, F. and Morlock, L. (2011), "Random Effects Models for Identifying the Most Harmful Medication Errors in a Large, Voluntary Reporting Database". Technical Report.

See Also

[dnegbinom](#), [EBGM](#), [mixnegbinom.em](#).

Examples

```
data("simdata", package = "mederrRank")
summary(simdata)

## Not run:
fit <- bhm.mcmc(simdata, nsim = 1000, burnin = 500, scale.factor = 1.1)
resamp <- bhm.resample(fit, simdata, p.resample = .1,
  k = c(3, 6, 10, 30, 60, Inf), eta = c(.5, .8, 1, 1.25, 2))
fit2 <- bhm.constr.resamp(fit, resamp, k = 3, eta = .8)
plot(fit, fit2, simdata)

## End(Not run)

theta0 <- runif(2, 0, 5)
ans <- negbinom.em(simdata, theta0, 50000, 0.01,
  se = TRUE, stratified = TRUE)
ans$theta
ans$se

## Not run:
summary(fit2, ans, simdata)

## End(Not run)
```

negbinom.loglik

Log-Likelihood Function for the Mixture of Negative Binomial Distributions

Description

This function computes the log-likelihood function for the mixture of two negative binomial distribution as described in [dmixnegbinom](#).

Usage

```
negbinom.loglik(theta, N, E)
```

Arguments

theta	vector of parameter values.
N	vector of observed error profiles counts.
E	vector of expected error profiles counts.

Details

For further details see Myers et al. (2011).

Value

negbinom.loglik returns the log-likelihood value for the negative binomial distribution.

Author(s)

Sergio Venturini <sergio.venturini@unicatt.it>,
Jessica A. Myers <jmyers6@partners.org>

References

DuMouchel W. (1999), "Bayesian Data Mining in Large Frequency Tables, with an Application to the FDA Spontaneous Reporting System". *The American Statistician*, 53, 177-190.

Myers, J. A., Venturini, S., Dominici, F. and Morlock, L. (2011), "Random Effects Models for Identifying the Most Harmful Medication Errors in a Large, Voluntary Reporting Database". Technical Report.

See Also

[dnegbinom](#), [negbinom.em](#), [negbinom.score](#).

negbinom.score

Log-Likelihood Score Function for the Negative Binomial Distribution

Description

This function computes the log-likelihood score for the negative binomial distribution as described in [dmixnegbinom](#).

Usage

```
negbinom.score(theta, N, E)
```

Arguments

theta	vector of parameter values.
N	vector of observed error profiles counts.
E	vector of expected error profiles counts.

Details

For further details see Myers et al. (2011).

Value

negbinom.score returns the vector of log-likelihood score values for the negative binomial distribution.

Author(s)

Sergio Venturini <sergio.venturini@unicatt.it>,
Jessica A. Myers <jmyers6@partners.org>

References

DuMouchel W. (1999), "Bayesian Data Mining in Large Frequency Tables, with an Application to the FDA Spontaneous Reporting System". *The American Statistician*, 53, 177-190.

Myers, J. A., Venturini, S., Dominici, F. and Morlock, L. (2011), "Random Effects Models for Identifying the Most Harmful Medication Errors in a Large, Voluntary Reporting Database". Technical Report.

See Also

[dnegbinom](#), [negbinom.em](#), [negbinom.loglik](#).

p.value

Posterior Predictive Test statistics

Description

This function computes posterior predictive test statistics as described in Myers et al. (2011).

Usage

```
p.value(reps)
```

Arguments

reps list of replications created with the [post.rep](#) function.

Details

For further details see Myers et al. (2011).

Value

p-value creates a list of p-values.

Author(s)

Sergio Venturini <sergio.venturini@unicatt.it>,
Jessica A. Myers <jmyers6@partners.org>

References

Myers, J. A., Venturini, S., Dominici, F. and Morlock, L. (2011), "Random Effects Models for Identifying the Most Harmful Medication Errors in a Large, Voluntary Reporting Database". Technical Report.

See Also

[bhm.constr.resamp](#), [bhm.mcmc](#), [bhm.resample](#), [post.rep](#).

Examples

```
## Not run:
data("simdata", package = "mederrRank")
summary(simdata)

fit <- bhm.mcmc(simdata, nsim = 1000, burnin = 500, scale.factor = 1.1)
resamp <- bhm.resample(fit, simdata, p.resample = .1,
  k = c(3, 6, 10, 30, 60, Inf), eta = c(.5, .8, 1, 1.25, 2))
fit2 <- bhm.constr.resamp(fit, resamp, k = 3, eta = .8)
reps <- post.rep(fit2, simdata)
pvalues <- p.value(reps)

## End(Not run)
```

Description

Methods for function plot in Package 'graphics' to be used with "[mederrData](#)", [mederrFit](#) and "[mederrResample](#)" objects.

Methods

signature(x = "mederrData", y = "missing") Pictorial representation for a "[mederrData](#)" object.

signature(x = "mederrFit", y = "mederrFit") Graphical representation of Markov Chain Monte Carlo simulations for a "[mederrFit](#)" object.

signature(x = "mederrResample", y = "missing") Graphical representation of the resampling transformation for a "[mederrResample](#)" object.

Author(s)

Sergio Venturini <sergio.venturini@unicatt.it>,

Jessica A. Myers <jmyers6@partners.org>

References

Myers, J. A., Venturini, S., Dominici, F. and Morlock, L. (2011), "Random Effects Models for Identifying the Most Harmful Medication Errors in a Large, Voluntary Reporting Database". Technical Report.

See Also

[bhm.mcmc](#), [bhm.resample](#).

post.rep

Posterior Predictive Data Replications

Description

This function creates a list of replicated data for posterior predictive checking as described in Myers et al. (2011).

Usage

```
post.rep(model, dat)
```

Arguments

model an object of class "[mederrFit](#)".

dat an object of class "[mederrData](#)".

Details

For further details see Myers et al. (2011).

Value

post.rep returns a list of replicated data.

Author(s)

Sergio Venturini <sergio.venturini@unicatt.it>,
Jessica A. Myers <jmyers6@partners.org>

References

Myers, J. A., Venturini, S., Dominici, F. and Morlock, L. (2011), "Random Effects Models for Identifying the Most Harmful Medication Errors in a Large, Voluntary Reporting Database". Technical Report.

See Also

[bhm.constr.resamp](#), [bhm.mcmc](#), [bhm.resample](#), [p.value](#).

rmixnegbinom

The Negative Binomial Mixture Distribution

Description

Random generation for a mixture of two negative binomial distributions.

Usage

```
rmixnegbinom(n, theta, E)
```

Arguments

n	number of observations.
theta	vector of parameters for the negative binomial distribution mixture.
E	vector of (non-negative integer) expected counts.

Value

rmixnegbinom generates random deviates corresponding to the E and theta values provided.

Author(s)

Sergio Venturini <sergio.venturini@unicatt.it>,
Jessica A. Myers <jmyers6@partners.org>

References

Myers, J. A., Venturini, S., Dominici, F. and Morlock, L. (2011), "Random Effects Models for Identifying the Most Harmful Medication Errors in a Large, Voluntary Reporting Database". Technical Report.

See Also

[dmixnegbinom](#), [rnbinom](#).

simdata

Simulated Data

Description

Simulated data to use for illustrative purposes in the [mederrRank](#) package.

Usage

```
data(simdata)
```

Format

An object of class [mederrData](#).

Author(s)

Sergio Venturini <sergio.venturini@unicatt.it>,
Jessica A. Myers <jmyers6@partners.org>

References

Myers, J. A., Venturini, S., Dominici, F. and Morlock, L. (2011), "Random Effects Models for Identifying the Most Harmful Medication Errors in a Large, Voluntary Reporting Database". Technical Report.

See Also

[bhm.mcmc](#), [mederrData](#), [mederrFit](#).

Examples

```
data("simdata", package = "mederrRank")  
summary(simdata)  
plot(simdata)
```

Description

Methods for function `summary` in Package 'base' to be used with "[mederrData](#)" and "[mederrFit](#)" objects.

Methods

`signature(object = "mederrData")` Extracts summary information about the slots of a "[mederrData](#)" object.

`signature(object = "mederrFit")` Extracts summary information about the slots of a "[mederrFit](#)" object.

Author(s)

Sergio Venturini <sergio.venturini@unicatt.it>,
Jessica A. Myers <jmyers6@partners.org>

References

Myers, J. A., Venturini, S., Dominici, F. and Morlock, L. (2011), "Random Effects Models for Identifying the Most Harmful Medication Errors in a Large, Voluntary Reporting Database". Technical Report.

See Also

[bayes.rank](#), [bhm.mcmc](#).

Index

- * **Bayesian Hierarchical Models**
 - mederrRank-package, 2
- * **Bayesian Hierarchical Model**
 - bhm.constr.resamp, 4
 - bhm.mcmc, 5
 - bhm.resample, 7
 - MEDMARX, 21
 - simdata, 33
- * **Counts**
 - dmixnegbinom, 9
 - dnegbinom, 10
 - rmixnegbinom, 32
- * **DuMocuhel**
 - dmixnegbinom, 9
 - dnegbinom, 10
 - rmixnegbinom, 32
- * **EM**
 - mederrRank-package, 2
 - mixnegbinom.em, 22
 - mixnegbinom.loglik, 24
 - mixnegbinom.score, 25
 - negbinom.em, 26
 - negbinom.loglik, 27
 - negbinom.score, 28
- * **Empirical Bayes**
 - EBGM, 12
 - mederrRank-package, 2
 - mixnegbinom.em, 22
 - mixnegbinom.loglik, 24
 - mixnegbinom.score, 25
 - negbinom.em, 26
 - negbinom.loglik, 27
 - negbinom.score, 28
- * **Markov Chain Monte Carlo**
 - bhm.constr.resamp, 4
 - bhm.mcmc, 5
 - bhm.resample, 7
- * **Medication Errors**
 - bayes.rank, 3
 - bhm.constr.resamp, 4
 - bhm.mcmc, 5
 - bhm.resample, 7
 - dmixnegbinom, 9
 - dnegbinom, 10
 - EBGM, 12
 - l1DiffD, 14
 - l1DiffT, 15
 - logp, 16
 - logunpost, 17
 - MEDMARX, 21
 - mixnegbinom.em, 22
 - mixnegbinom.loglik, 24
 - mixnegbinom.score, 25
 - negbinom.em, 26
 - negbinom.loglik, 27
 - negbinom.score, 28
 - p.value, 29
 - plot-methods, 30
 - post.rep, 31
 - rmixnegbinom, 32
 - simdata, 33
 - summary-methods, 34
- * **Mixture Distributions**
 - mederrRank-package, 2
- * **Mixture**
 - dmixnegbinom, 9
 - mixnegbinom.loglik, 24
 - mixnegbinom.score, 25
 - negbinom.loglik, 27
 - negbinom.score, 28
 - rmixnegbinom, 32
- * **Negative Binomial Distribution**
 - dmixnegbinom, 9
 - rmixnegbinom, 32
- * **Optimal Bayes ranking**
 - mederrRank-package, 2
- * **Optimal Bayesian Ranking**
 - bayes.rank, 3

- bhm.constr.resamp, 4
- bhm.mcmc, 5
- bhm.resample, 7
- EBGM, 12
- l1DiffD, 14
- l1DiffT, 15
- logp, 16
- logunpost, 17
- MEDMARX, 21
- mixnegbinom.em, 22
- mixnegbinom.loglik, 24
- mixnegbinom.score, 25
- negbinom.em, 26
- negbinom.loglik, 27
- negbinom.score, 28
- p.value, 29
- post.rep, 31
- simdata, 33
- * classes**
 - mederrData-class, 18
 - mederrFit-class, 19
 - mederrResample-class, 20
- * methods**
 - mederrData-class, 18
 - mederrFit-class, 19
 - mederrResample-class, 20
 - plot-methods, 30
 - summary-methods, 34
- * ranking**
 - bayes.rank, 3
 - bhm.constr.resamp, 4
 - bhm.mcmc, 5
 - bhm.resample, 7
 - EBGM, 12
 - l1DiffD, 14
 - l1DiffT, 15
 - logp, 16
 - logunpost, 17
 - MEDMARX, 21
 - mixnegbinom.em, 22
 - mixnegbinom.loglik, 24
 - mixnegbinom.score, 25
 - negbinom.em, 26
 - negbinom.loglik, 27
 - negbinom.score, 28
 - p.value, 29
 - post.rep, 31
 - simdata, 33
- * ranks**
 - bayes.rank, 3
 - l1DiffD, 14
 - l1DiffT, 15
 - logp, 16
 - logunpost, 17
 - p.value, 29
 - post.rep, 31
- * t Distribution**
 - dst, 11
- bayes.rank, 3, 3, 19–21, 34
- bhm.constr.resamp, 4, 4, 14–17, 19–21, 30, 32
- bhm.mcmc, 2–5, 5, 7, 8, 14–17, 19–22, 30–34
- bhm.resample, 2–5, 7, 7, 14–17, 19, 20, 30–32
- class, 4, 6, 8, 16, 17, 22, 26, 31
- data.frame, 18
- dmixnegbinom, 9, 11, 22–25, 27, 28, 33
- dnbinom, 10, 11
- dnegbinom, 10, 26–29
- dst, 11
- dt, 12
- EBGM, 12, 23, 27
- initialize, mederrData-method (mederrData-class), 18
- initialize, mederrFit-method (mederrFit-class), 19
- initialize, mederrResample-method (mederrResample-class), 20
- l1DiffD, 14
- l1DiffT, 15
- logp, 16
- logunpost, 17
- mederrData, 5–8, 16, 18, 21, 22, 26, 30, 31, 33, 34
- mederrData (mederrData-class), 18
- mederrData-class, 18
- mederrFit, 3–5, 7, 8, 22, 30, 31, 33, 34
- mederrFit (mederrFit-class), 19
- mederrFit-class, 19
- mederrRank, 21, 33
- mederrRank (mederrRank-package), 2
- mederrRank-package, 2

mederrResample, [4](#), [8](#), [17](#), [21](#), [30](#), [31](#)
mederrResample (mederrResample-class),
 [20](#)
mederrResample-class, [20](#)
MEDMARX, [21](#)
mixnegbinom.em, [2](#), [3](#), [13](#), [19](#), [22](#), [25](#), [27](#)
mixnegbinom.loglik, [24](#), [25](#)
mixnegbinom.score, [25](#), [25](#)

negbinom.em, [13](#), [23](#), [26](#), [28](#), [29](#)
negbinom.loglik, [27](#), [29](#)
negbinom.score, [28](#), [28](#)

p.value, [29](#), [32](#)
plot, ANY, ANY-method (plot-methods), [30](#)
plot, mederrData, missing-method
 (plot-methods), [30](#)
plot, mederrFit, mederrFit-method
 (plot-methods), [30](#)
plot, mederrResample, missing-method
 (plot-methods), [30](#)
plot-methods, [30](#)
post.rep, [29](#), [30](#), [31](#)

rmixnegbinom, [10](#), [32](#)
rnbinom, [33](#)

simdata, [33](#)
summary, ANY-method (summary-methods), [34](#)
summary, mederrData-method
 (summary-methods), [34](#)
summary, mederrFit-method
 (summary-methods), [34](#)
summary-methods, [34](#)