

# Package ‘boxcoxmix’

December 21, 2017

**Type** Package

**Title** Response Transformations for Random Effect and Variance  
Component Models

**Version** 0.15

**Date** 2017-12-21

**Author** Amani Almohaimeed and Jochen Einbeck

**Maintainer** Amani Almohaimeed <amani.almohaimeed@gmail.com>

**Depends** R (>= 3.3.2)

**Imports** statmod(>= 1.4.20), qicharts(>= 0.5.4)

**Suggests** npmlreg, nlme, mdscore, utils

**LazyLoad** yes

**Description** Response transformations for overdispersed generalized linear  
models and variance component models using nonparametric profile maximum  
likelihood estimation. The main function is `optim.boxcox()`.

**License** GPL (>= 3)

**RoxygenNote** 6.0.1

**VignetteBuilder** utils

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2017-12-21 19:00:20 UTC

## R topics documented:

boxcoxmix-package . . . . .	2
np.boxcoxmix . . . . .	2
np.estep . . . . .	5
optim.boxcox . . . . .	7
plot . . . . .	10
print.boxcoxmix . . . . .	11
tolfind.boxcox . . . . .	12

<b>Index</b>	<b>14</b>
--------------	-----------

---

boxcoxmix-package	<i>Response Transformations for Random Effect and Variance Component Models</i>
-------------------	---

---

### Description

Response transformations for overdispersed generalized linear models and variance component models using nonparametric profile maximum likelihood estimation. The main function is `optim.boxcox()`.

### Details

Package: boxcoxmix  
Type: Package  
Version: 0.15  
Date: 2017-12-21  
License: GPL (>=3)

### Author(s)

Amani Almohaimeed and Jochen Einbeck

### References

Box G. and Cox D. (1964). An analysis of transformations. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 211-252.

Aitkin, M. A., Francis, B., Hinde, J., and Darnell, R. (2009). *Statistical modelling in R*. Oxford University Press Oxford.

Jochen Einbeck, Ross Darnell and John Hinde (2014). *npmlreg: Nonparametric maximum likelihood estimation for random effect models*. R package version 0.46-1.

Maintainer: Amani Almohaimeed <amani.almohaimeed@gmail.com>

---

np.boxcoxmix	<i>Response Transformations for Random Effect and Variance Component Models</i>
--------------	---

---

### Description

The function `np.boxcoxmix()` fits an overdispersed generalized linear model and variance component models using nonparametric profile maximum likelihood.

**Usage**

```
np.boxcoxm(formula, groups = 1, data, K = 3, tol = 0.5, lambda = 1,
  steps = 500, EMdev.change = 1e-04, plot.opt = 1, verbose = TRUE,
  start = "quantile", ...)
```

**Arguments**

formula	a formula describing the transformed response and the fixed effect model (e.g. $y \sim x$ ).
groups	the random effects. To fit overdispersion models, set <code>groups = 1</code> .
data	a data frame containing variables used in the fixed and random effect models.
K	the number of mass points.
tol	a positive scalar (usually, $0 < \text{tol} \leq 2$ )
lambda	a transformation parameter, setting <code>lambda=1</code> means 'no transformation'.
steps	maximum number of iterations for the EM algorithm.
EMdev.change	a small scalar, with default 0.0001, used to determine when to stop EM algorithm.
plot.opt	Set <code>plot.opt=1</code> , to plot the disparity against iteration number. Use <code>plot.opt=2</code> for <code>tolfind.boxcox()</code> and <code>plot.opt=3</code> for <code>optim.boxcox()</code> .
verbose	If set to <code>FALSE</code> , no printed output on progress.
start	a description of the initial values to be used in the fitted model, Quantile-based version "quantile" or Gaussian Quadrature "gq" can be set.
...	extra arguments will be ignored.

**Details**

The Box-Cox transformation (Box & Cox, 1964) is applied to the overdispersed generalized linear models and variance component models with an unspecified mixing distribution. The NPML estimate of the mixing distribution is known to be a discrete distribution involving a finite number of mass-points and corresponding masses (Aitkin et al., 2009). An Expectation-Maximization (EM) algorithm is used for fitting the finite mixture distribution, one needs to specify the number of components  $K$  of the finite mixture in advance. To stop the EM-algorithm when it reached its convergence point, we need to defined the convergence criteria that is the absolute change in the successive log-likelihood function values being less than an arbitrary parameter such as `EMdev.change = 0.0001` (Einbeck et al., 2014). This algorithm can be implemented using the function `np.boxcoxm()`, which is designed to account for overdispersed generalized linear models and variance component models using the non-parametric profile maximum likelihood (NPPML) estimation.

The ability of the EM algorithm to locate the global maximum in fewer iterations can be affected by the choice of initial values, the function `np.boxcoxm()` allows us to choose from two different methods to set the initial value of the mass points. When option "gq" is set, then Gauss-Hermite masses and mass points are used as starting points in the EM algorithm, while setting `start= "quantile"` uses the Quantile-based version to select the starting points.

**Value**

mass.point	the fitted mass points.
p	the masses corresponding to the mixing proportions.
beta	the vector of coefficients.
sigma	the standard deviation of the mixing distribution (the square root of the variance).
se	the standard error of the estimate.
w	a matrix of posterior probabilities that element i comes from cluster k.
loglik	the log-likelihood of the fitted regression model.
profile.loglik	the profile log-likelihood of the fitted regression model.
disparity	the disparity of the fitted regression model.
EMiteration	provides the number of iterations of the EM algorithm.
EMconverged	TRUE means the EM algorithm converged.
call	the matched call.
formula	the formula provided.
data	the data argument.
aic	the Akaike information criterion of the fitted regression model.
fitted	the fitted values for the individual observations.
fitted.transformed	the fitted values for the individual transformed observations.
residuals	the difference between the observed values and the fitted values.
residuals.transformed	the difference between the transformed observed values and the transformed fitted values.
predicted.re	a vector of predicted residuals.

The other outcomes are not relevant to users and they are intended for internal use only.

**Author(s)**

Amani Almohaimeed and Jochen Einbeck

**References**

- Box G. and Cox D. (1964). An analysis of transformations. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 211-252.
- Aitkin, M. A., Francis, B., Hinde, J., and Darnell, R. (2009). *Statistical modelling in R*. Oxford University Press Oxford.
- Jochen Einbeck, Ross Darnell and John Hinde (2014). *npmlreg: Nonparametric maximum likelihood estimation for random effect models*. R package version 0.46-1.

**See Also**

[optim.boxcox](#), [tolfind.boxcox](#).

**Examples**

```

# The Pennsylvania Hospital Stay Data
data(hosp, package = "nplreg")
test1 <- np.boxcoxm(x ~ age + wbc1, data = hosp, K = 2, tol = 1,
  start = "quantile", lambda = 1)
round(summary(test1)$w, digits = 3)
# [1,] 1.000 0.000

# Refinery yield of gasoline
data(Gasoline, package = "nlme")
test2.vc <- np.boxcoxm(yield ~ endpoint + vapor, groups = Gasoline$Sample,
  data = Gasoline, K = 3, tol = 1.7, start = "quantile", lambda = 0)
test2.vc$disparity
# [1] 176.9827

```

---

np.estep

*Internal boxcoxm functions*


---

**Description**

auxiliary functions are not intended to be directly called from the user.

**Usage**

```
np.estep(y, x, lambda, p, beta, z, sigma)
```

```
np.zk(y, x, w, beta, lambda)
```

```
fik(y, x, lambda, beta, z, sigma)
```

```
np.theta(y, x, lambda, beta, z)
```

```
yhat(v, lambda = 1)
```

```
ytrans(y, lambda = 1)
```

```
np.bhat(y, x, w, z, lambda)
```

```
np.mstep(y, x, beta, lambda, w)
```

```

np.em(y, x, K, lambda = 1, steps = 500, tol = 0.5, start = "gq",
      EMdev.change = 1e-04, plot.opt = 1, verbose = TRUE, ...)

np.boxcox(formula, groups = 1, data, K = 3, tol = 0.5, lambda = 1,
          steps = 500, EMdev.change = 1e-04, plot.opt = 1, verbose = TRUE,
          start = "quantile", ...)

vc.estep(Y, X, sizes = 1, lambda, p, beta, z, sigma)

zk(Y, X, sizes, w, beta, lambda)

bhat(Y, X, sizes, w, z, lambda)

mik(Y, X, sizes, lambda, beta, z, sigma)

vc.theta(Y, X, sizes, lambda, beta, z)

vc.mstep(Y, X, sizes = 1, beta, lambda, w)

vc.em(y, x, sizes = 1, K, lambda, steps = 500, tol = 0.5, start = "gq",
      EMdev.change = 1e-04, plot.opt = 1, verbose = TRUE, ...)

vc.boxcox(formula, groups = 1, data, K = 3, tol = 0.5, lambda = 1,
          steps = 500, EMdev.change = 1e-04, plot.opt = 1, verbose = TRUE,
          start = "quantile", ...)

gqz(numnodes = 20, minweight = 1e-06)

masspoint.class(object)

```

### Arguments

y	..
x	..
lambda	a transformation parameter, setting lambda=1 means 'no transformation'.
p	..
beta	..
z	..
sigma	..
w	..
v	..
K	the number of mass points.
steps	maximum number of iterations for the EM algorithm.
tol	a positive scalar (usually, $0 < \text{tol} \leq 2$ )
start	a description of the initial values to be used in the fitted model, Quantile-based version "quantile" or Gaussian Quadrature "gq" can be set.

EMdev.change	a small scalar, with default 0.0001, used to determine when to stop EM algorithm.
plot.opt	Set plot.opt=1, to plot the disparity against iteration number. Use plot.opt=2 for tolfind.boxcox and plot.opt=3 for optim.boxcox.
verbose	If set to FALSE, no printed output on progress.
...	extra arguments will be ignored.
formula	a formula describing the transformed response and the fixed effect model (e.g. $y \sim x$ ).
groups	the random effects. To fit overdispersion models, set groups = 1.
data	a data frame containing variables used in the fixed and random effect models.
Y	..
X	..
sizes	..
numnodes	..
minweight	..
object	..

## Details

Internal boxcoxmix functions

## Author(s)

Amani Almohaimeed and Jochen Einbeck

---

optim.boxcox	<i>Response Transformations for Random Effect and Variance Component Models</i>
--------------	---

---

## Description

The `optim.boxcox()` performs a grid search over the parameter `lambda` for overdispersed generalized linear models and variance component models and then optimizes over this grid, to calculate the maximum likelihood estimator of the transformation.

## Usage

```
optim.boxcox(formula, groups = 1, data, K = 3, steps = 500, tol = 0.5,
  start = "gq", EMdev.change = 1e-04, find.in.range = c(-3, 3), s = 60,
  plot.opt = 3, verbose = FALSE, noformat = FALSE, ...)
```

**Arguments**

formula	a formula describing the transformed response and the fixed effect model (e.g. $y \sim x$ ).
groups	the random effects. To fit overdispersion models, set <code>groups = 1</code> .
data	a data frame containing variables used in the fixed and random effect models.
K	the number of mass points.
steps	maximum number of iterations for the EM algorithm.
tol	a positive scalar (usually, $0 < \text{tol} \leq 2$ )
start	a description of the initial values to be used in the fitted model, Quantile-based version "quantile" or Gaussian Quadrature "gq" can be set.
EMdev.change	a small scalar, with default 0.0001, used to determine when to stop EM algorithm.
find.in.range	search in a range of <code>lambda</code> , with default (-3,3) in step of 0.1.
s	number of points in the grid search of <code>lambda</code> .
plot.opt	Set <code>plot.opt=3</code> , to plot the disparity against iteration number and the profile log-likelihood against <code>lambda</code> . Use <code>plot.opt=0</code> , to only plot the profile log-likelihood against <code>lambda</code> .
verbose	If set to FALSE, no printed output on progress.
noformat	Set <code>noformat = TRUE</code> , to change the formatting of the plots.
...	extra arguments will be ignored.

**Details**

The Box-Cox transformation (Box & Cox, 1964) is applied to the overdispersed generalized linear models and variance component models with an unspecified mixing distribution. The NPML estimate of the mixing distribution is known to be a discrete distribution involving a finite number of mass-points and corresponding masses (Aitkin et al., 2009). An Expectation-Maximization (EM) algorithm is used for fitting the finite mixture distribution, one needs to specify the number of components  $K$  of the finite mixture in advance. To stop the EM-algorithm when it reached its convergence point, we need to defined the convergence criteria that is the absolute change in the successive log-likelihood function values being less than an arbitrary parameter such as `EMdev.change = 0.0001` (Einbeck et al., 2014). This algorithm can be implemented using the function `np.boxcoxmix()`, which is designed to account for overdispersed generalized linear models and variance component models using the non-parametric profile maximum likelihood (NPPML) estimation.

The ability of the EM algorithm to locate the global maximum in fewer iterations can be affected by the choice of initial values, the function `optim.boxcox()` allows us to choose from two different methods to set the initial value of the mass points. When option "gq" is set, then Gauss-Hermite masses and mass points are used as starting points in the EM algorithm, while setting `start= "quantile"` uses the Quantile-based version to select the starting points.

`optim.boxcox()` performs a grid search over the parameter `lambda` and then optimizes over this grid, to calculate the maximum likelihood estimator of the transformation. It produces a plot of the non-parametric profile likelihood function that summarises information concerning `lambda`, including a vertical line indicating the best value of `lambda` that maximizes the non-parametric profile log-likelihood.



**Value**

All.lambda	list of lambda values used in the grid.
Maximum	the best estimate of lambda found.
objective	the value of the profile log-likelihood corresponding to Maximum.
EMconverged	1 is TRUE, means the EM algorithm converged.
EMiteration	provides the number of iterations of the EM algorithm.
mass.point	the fitted mass points.
p	the masses corresponding to the mixing proportions.
beta	the vector of coefficients.
sigma	the standard deviation of the mixing distribution (the square root of the variance).
se	the standard error of the estimate.
w	a matrix of posterior probabilities that element i comes from cluster k.
loglik	the log-likelihood of the fitted regression model.
profile.loglik	the profile log-likelihood of the fitted regression model.
disparity	the disparity of the fitted regression model.
call	the matched call.
formula	the formula provided.
data	the data argument.
aic	the Akaike information criterion of the fitted regression model.
fitted	the fitted values for the individual observations.
fitted.transformed	the fitted values for the individual transformed observations.
residuals	the difference between the observed values and the fitted values.
residuals.transformed	the difference between the transformed observed values and the transformed fitted values.
predicted.re	a vector of predicted residuals.

The other outcomes are not relevant to users and they are intended for internal use only.

**Author(s)**

Amani Almohaimeed and Jochen Einbeck

**References**

- Box G. and Cox D. (1964). An analysis of transformations. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 211-252.
- Aitkin, M. A., Francis, B., Hinde, J., and Darnell, R. (2009). *Statistical modelling in R*. Oxford University Press Oxford.
- Jochen Einbeck, Ross Darnell and John Hinde (2014). *npmlreg: Nonparametric maximum likelihood estimation for random effect models*. R package version 0.46-1.

**See Also**

[np.boxcoxm](#), [tolfind.boxcox](#).

**Examples**

```
# The Pennsylvanian Hospital Stay Data
data(hosp, package = "npmlreg")
test1 <- optim.boxcox(duration ~ age , data = hosp, K =2, tol = 0,
                      start = "quantile", steps = 800,
                      find.in.range = c(-2, 2), s = 15, plot.opt = 0)
# Maximum profile Log-likelihood: -86.15186 at lambda= -0.1333333

# Effect of Phenylbiguanide on Blood Pressure
data(PBG, package = "nlme")
test3<- optim.boxcox(deltaBP ~ dose , groups = PBG$Rabbit,
                     data = PBG, K = 2, tol = 0, start = "gq",
                     find.in.range = c(-1,2), s = 12, plot.opt = 0)
# Maximum profile Log-likelihood: -174.6545 at lambda= 0.25
```

---

plot

*Plot diagnostics for boxcoxm functions*

---

**Description**

plot() is a generic function used to produce some useful diagnostic plotting of the functions: np.boxcoxm(), optim.boxcox() and tolfind.boxcox().

**Usage**

```
## S3 method for class 'boxcoxm'
plot(x, plot.opt = 1, ...)
```

**Arguments**

x                    an object for which a plot is desired.  
 plot.opt            an integer value between 1 and 8.  
 ...                 additional arguments.

**Details**

Plot diagnostics for boxcoxm functions

**Value**

The plots to be printed depend on the number given in `plot.opt`, for the `np.boxcoxmix()`, `optim.boxcox()` and `tolfind.boxcox()` functions:

- |   |  |
|---|--|
| 1 | the disparities with the iteration number against the mass points  |
| 2 | the fitted value against the response of the original and the transformed Data.  |
| 3 | probability plot of residuals of the original against the transformed data.  |
| 4 | individual posterior probabilities.  |
| 5 | control charts of residuals of the original against the transformed data.  |
| 6 | The histograms of residuals of the original against the transformed data.  |
| 7 | works only for the <code>tolfind.boxcox()</code> function and plots the specified range of <code>tol</code> against the disparities                            |
| 8 | works only for the <code>optim.boxcox()</code> function and gives the profile likelihood function that summarises information concerning <code>lambda</code> . |

---

print.boxcoxmix

*Summary of boxcoxmix functions*


---

**Description**

`summary()` and `print()` are generic functions used to produce the results of the functions: `np.boxcoxmix()`, `optim.boxcox()` and `tolfind.boxcox()`.

**Usage**

```
## S3 method for class 'boxcoxmix'
print(x, digits = max(3, getOption("digits") - 3),
      na.print = "", ...)

## S3 method for class 'boxcoxmixpure'
print(x, digits = max(3, getOption("digits") - 3),
      na.print = "", ...)

## S3 method for class 'boxcoxmix'
summary(object, digits = max(3, getOption("digits") - 3),
        ...)

## S3 method for class 'boxcoxmixpure'
summary(object, digits = max(3, getOption("digits") -
3), ...)
```

**Arguments**

x	an object for which a summary is desired.
digits	an integer number format.
na.print	a character string which is used to indicate NA values output format.
...	additional arguments.
object	an object for which a summary is desired.

**Details**

Summary of boxcoxmix functions

---

tolfind.boxcox	<i>Grid search over tol for NPPML estimation of random effect and variance component models</i>
----------------	---

---

**Description**

A grid search over the parameter tol, to set the initial values of the EM algorithm.

**Usage**

```
tolfind.boxcox(formula, groups = 1, data, K = 3, lambda = 1,
  EMdev.change = 1e-04, plot.opt = 2, s = 20, steps = 500,
  find.in.range = c(0, 2), start = "gq", verbose = FALSE,
  noformat = FALSE, ...)
```

**Arguments**

formula	a formula describing the transformed response and the fixed effect model (e.g. $y \sim x$ ).
groups	the random effects. To fit overdispersion models, set groups = 1.
data	a data frame containing variables used in the fixed and random effect models.
K	the number of mass points.
lambda	a transformation parameter, setting lambda=1 means 'no transformation'.
EMdev.change	a small scalar, with default 0.0001, used to determine when to stop EM algorithm.
plot.opt	Set plot.opt=2, to plot the disparity against iteration number.
s	number of points in the grid search of tol.
steps	maximum number of iterations for the EM algorithm.
find.in.range	search in a range of tol, with default (0,2) in step of 0.1.
start	a description of the initial values to be used in the fitted model, Quantile-based version "quantile" or Gaussian Quadrature "gq" can be set.
verbose	If set to FALSE, no printed output on progress.
noformat	Set noformat = TRUE, to change the formatting of the plots.
...	extra arguments will be ignored.

**Details**

A grid search over tol can be performed using tolfind.boxcox() function, which works for np.boxcoxmixin() to find the optimal solution.

**Value**

MinDisparity the minimum disparity found.  
 Mintol the value of tol corresponding to MinDisparity.  
 AllDisparities a vector containing all disparities calculated on the grid.  
 Alltol list of tol values used in the grid.  
 AllEMconverged 1 is TRUE, means the EM algorithm converged.

**Author(s)**

Amani Almohaimeed and Jochen Einbeck

**See Also**

[np.boxcoxmixin](#).

**Examples**

```
# The Pennsylvanian Hospital Stay Data
data(hosp, package = "npmlreg")
test1 <- tolfind.boxcox(duration ~ age , data = hosp, K = 2, lambda = 0,
  s = 10, steps = 600, start = "quantile", plot.opt = 0)
# Minimal Disparity: 137.8364 at tol= 1.4
# Minimal Disparity with EM converged: 137.8364 at tol= 1.4

# Effect of Phenylbiguanide on Blood Pressure
data(PBG, package = "nlme")
test3 <- tolfind.boxcox(deltaBP ~ dose , groups = PBG$Rabbit,
  data = PBG, K = 2, lambda = -1, s = 15, steps = 500, start = "quantile", plot.opt = 0)
# Minimal Disparity: 449.5876 at tol= 1.6
# Minimal Disparity with EM converged: 449.5876 at tol= 1.6
```

# Index

- \*Topic **boxcox**
  - np.boxcoxmix, 2
  - optim.boxcox, 7
  - tolfind.boxcox, 12
- \*Topic **em**
  - np.estep, 5
- \*Topic **optim**
  - optim.boxcox, 7
- \*Topic **package**
  - boxcoxmix-package, 2
- \*Topic **random**
  - np.boxcoxmix, 2
- \*Topic **tolfind**
  - tolfind.boxcox, 12
- \*Topic **variance**
  - np.boxcoxmix, 2

bhat (np.estep), 5

boxcoxmix (boxcoxmix-package), 2

boxcoxmix-package, 2

fik (np.estep), 5

gqz (np.estep), 5

masspoint.class (np.estep), 5

mik (np.estep), 5

nb.se (np.estep), 5

np.bhat (np.estep), 5

np.boxcox (np.estep), 5

np.boxcoxmix, 2, 10, 13

np.em (np.estep), 5

np.estep, 5

np.mstep (np.estep), 5

np.theta (np.estep), 5

np.zk (np.estep), 5

optim.boxcox, 4, 7

plot, 10

print.boxcoxmix, 11

print.boxcoxmixpure (print.boxcoxmix), 11

summary.boxcoxmix (print.boxcoxmix), 11

summary.boxcoxmixpure (print.boxcoxmix), 11

tolfind.boxcox, 4, 10, 12

vc.boxcox (np.estep), 5

vc.em (np.estep), 5

vc.estep (np.estep), 5

vc.mstep (np.estep), 5

vc.se (np.estep), 5

vc.theta (np.estep), 5

yhat (np.estep), 5

ytrans (np.estep), 5

zk (np.estep), 5