

Package ‘boxcoxmix’

September 17, 2020

Type Package

Title Box-Cox-Type Transformations for Linear and Logistic Models with Random Effects

Version 0.28

Date 2020-09-17

Author Amani Almohameed and Jochen Einbeck

Maintainer Amani Almohameed <amani.almohameed@gmail.com>

Depends R (>= 3.3.0)

Imports statmod(>= 1.4.27), qicharts(>= 0.5.4), npmlreg(>= 0.46-1)

Suggests nlme, mdscore, flexmix, utils

LazyLoad yes

Description Box-Cox-type transformations for linear and logistic models with random effects using non-parametric profile maximum likelihood estimation. The main functions are `optim.boxcox()` for linear models with random effects and `boxcoxtype()` for logistic models with random effects.

License GPL (>= 3)

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

VignetteBuilder utils

NeedsCompilation no

Repository CRAN

Date/Publication 2020-09-17 07:00:02 UTC

R topics documented:

boxcoxmix-package	2
boxcoxtype	3
Kfind.boxcox	5

np.boxcoxmix	7
np.estep	10
optim.boxcox	13
plot	16
print.boxcoxmix	17
tolfind.boxcox	18

Index	21
--------------	-----------

boxcoxmix-package	<i>Box-Cox-Type Transformations for Linear and Logistic Models with Random Effects</i>
-------------------	--

Description

Box-Cox-type transformations for linear and logistic models with random effects using non-parametric profile maximum likelihood estimation. The main functions are `optim.boxcox()` for linear models with random effects and `boxcoxtype()` for logistic models with random effects.

Details

Package: boxcoxmix
 Type: Package
 Version: 0.28
 Date: 2020-09-17
 License: GPL (>=3)

Author(s)

Amani Almohameed 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.

R Core Team (2016). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna, Austria. URL <https://www.R-project.org/>.

Maintainer: Amani Almohameed <amani.almohameed@gmail.com>

 boxcoxtype

Box-Cox-type link function for logistic mixed-effects Models

Description

The `boxcoxtype()` performs a grid search over the parameter `Lambda` for logistic mixed-effects models and then optimizes over this grid, to calculate the maximum likelihood estimator of the transformation.

Usage

```

boxcoxtype(
  formula,
  random = ~1,
  k = 3,
  trials = 1,
  data,
  find.in.range = c(-2, 2),
  s = 20,
  plot.opt = 1,
  random.distribution = "np",
  ...
)

boxcoxpower(Lambda = 0)

binomial(link = boxcoxpower(0))

```

Arguments

<code>formula</code>	a formula describing the transformed response and the fixed effect model (e.g. $y \sim x$).
<code>random</code>	a formula defining the random model. Set <code>random= ~1</code> to model logistic-type overdispersion model. For a two-level logistic-type model, set <code>random= ~1 groups</code> , where groups are at the upper level.
<code>k</code>	the number of mass points.
<code>trials</code>	optional prior weights for the data. For Bernoulli distribution, set <code>trials=1</code> .
<code>data</code>	a data frame containing variables used in the fixed and random effect models.
<code>find.in.range</code>	search in a range of <code>Lambda</code> , with default <code>(-2,2)</code> in step of 0.1.
<code>s</code>	number of points in the grid search of <code>Lambda</code> .
<code>plot.opt</code>	Set <code>plot.opt=1</code> , to plot the profile log-likelihood against <code>Lambda</code> . if <code>plot.opt=0</code> , no plot is printed.
<code>random.distribution</code>	the mixing distribution, Gaussian Quadrature (gq) or NPML (np) can be set.

...	extra arguments will be ignored.
Lambda	the power of the transformation
link	the link function to be used.

Details

The Box-Cox transformation (Box & Cox, 1964) is applied to the logistic mixed-effects 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. This algorithm can be implemented using the `npmlreg` function `alldist` for the logistic-type overdispersion model and the `npmlreg` function `allvc` for the two-level logistic-type model, setting `family = binomial(link = boxcoxpower(Lambda))` where `Lambda` is the value of the power transformation. When $k=1$, the `npmlreg` function `alldist()` fits the logistic regression model without random effects.

`boxcoxtype()` 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 profile likelihood function that summarises information concerning `Lambda`, including a vertical line indicating the best value of `Lambda` that maximizes the profile log-likelihood.

Value

Maximum	the best estimate of <code>Lambda</code> found.
objective	the value of the profile log-likelihood corresponding to <code>Maximum</code> .
coef	the vector of coefficients.
profile.loglik	the profile log-likelihood of the fitted regression model.
fit	the fitted <code>alldist</code> object from the last EM iteration.
aic	the Akaike information criterion of the fitted regression model.
bic	the Bayesian information criterion of the fitted regression model.

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

Examples

```
#Beta blockers data
data("betablocker", package = "flexmix")
library(npmlreg)
betavc <- allvc(cbind(Deaths, Total - Deaths) ~ Treatment, data = betablocker, random=~1|Center,
  k=3, random.distribution='np', family = binomial(link = boxcoxpower(0)))
betavc$disparity
#[1] 318.7211
betavc3 <- boxcoxtype(cbind(Deaths, Total - Deaths) ~ Treatment, random=~1|Center,
  data = betablocker, find.in.range = c(-2,0.4), s=40, k=3, random.distribution='np')
#Maximum Profile Log-likelihood: -158.6025 at lambda= -0.56
betavc3$fit$disparity
#[1] 317.2049
betavc3$aic
#[1] 331.2049
betavc3$bic
#[1] 343.6942
```

Kfind.boxcox

Grid search over K for NPML estimation of random effect and variance component models

Description

A grid search over the parameter K, to set the best number of mass-points.

Usage

```
Kfind.boxcox(
  formula,
  groups = 1,
  data,
  lambda = 1,
  EMdev.change = 1e-04,
  steps = 500,
  find.k = c(2, 10),
  model.selection = "aic",
  start = "gq",
  find.tol = c(0, 1.5),
  steps.tol = 15,
  ...
)
```

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.
lambda	a transformation parameter, setting <code>lambda=1</code> means 'no transformation'.
EMdev.change	a small scalar, with default 0.0001, used to determine when to stop EM algorithm.
steps	maximum number of iterations for the EM algorithm.
find.k	search in a range of K, with default (2,10) in step of 1.
model.selection	Set <code>model.selection="aic"</code> , to use Akaike information criterion as model selection criterion or <code>model.selection="bic"</code> , to use Bayesian information criterion as model selection criterion.
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.
find.tol	search in a range of tol, with default (0,1.5) in step of 1.
steps.tol	number of points in the grid search of tol.
...	extra arguments will be ignored.

Details

Not only the shape of the distribution causes the skewness it may due to the use of an insufficient number of classes, K. For this, the `Kfind.boxcox()` function was created to search over a selected range of K and find the best. For each number of classes, a grid search over `tol` is performed and the `tol` with the lowest `aic` or `bic` value is considered as the optimal. Having the minimal `aic` or `bic` values for a whole range of K that have been specified beforehand, the `Kfind.boxcox()` function can find the best number of the component as the one with the smallest value. It also plots the `aic` or `bic` values against the selected range of K, including a vertical line indicating the best value of K that minimizes the model selection criteria. The full range of classes and their corresponding optimal `tol` can be printed off from the `Kfind.boxcox()`'s output and used with other **boxcoxmix** functions as starting points.

Value

MinDisparity	the minimum disparity found.
Best.K	the value of K corresponding to MinDisparity.
AllMinDisparities	a vector containing all minimum disparities calculated on the grid.
AllMintol	list of <code>tol</code> values used in the grid.
All.K	list of K values used in the grid.
All.aic	the Akaike information criterion of all fitted regression models.
All.bic	the Bayesian information criterion of all fitted regression models.

Author(s)

Amani Almohaimeed and Jochen Einbeck

See Also

[tolfind.boxcox](#).

Examples

```
# Fabric data
data(fabric, package = "npmlreg")
teststr<-Kfind.boxcox(y ~ x, data = fabric, start = "gq", groups=1,
find.k = c(2, 3), model.selection = "aic", steps.tol=5)
# Minimal AIC: 202.2114 at K= 2
```

np.boxcoxm

Response Transformations for Random Effect and Variance Component Models

Description

The function `np.boxcoxm()` 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 = "gq",
  ...
)
```

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.
tol	a positive scalar (usually, $0 < \text{tol} \leq 2$)
lambda	a transformation parameter, setting lambda=1 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 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.
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.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 `np.boxcoxmix()` 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.
complete.loglik	the complete 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.
bic	the Bayesian 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

```
#Pennsylvanian 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
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 = "gq",  
  ...  
)  
  
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 = "gq",
  ...
)

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 λ , with default (-3,3) in step of 0.1.
s	number of points in the grid search of λ .
plot.opt	Set <code>plot.opt=3</code> , to plot the disparity against iteration number and the profile log-likelihood against λ . Use <code>plot.opt=0</code> , to only plot the profile log-likelihood against λ .
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

<code>All.lambda</code>	list of <code>lambda</code> values used in the grid.
<code>Maximum</code>	the best estimate of <code>lambda</code> found.
<code>objective</code>	the value of the profile log-likelihood corresponding to <code>Maximum</code> .
<code>EMconverged</code>	1 is TRUE, means the EM algorithm converged.
<code>EMiteration</code>	provides the number of iterations of the EM algorithm.
<code>mass.point</code>	the fitted mass points.
<code>p</code>	the masses corresponding to the mixing proportions.
<code>beta</code>	the vector of coefficients.
<code>sigma</code>	the standard deviation of the mixing distribution (the square root of the variance).
<code>se</code>	the standard error of the estimate.
<code>w</code>	a matrix of posterior probabilities that element <code>i</code> comes from cluster <code>k</code> .
<code>loglik</code>	the profile log-likelihood of the fitted regression model.
<code>profile.loglik</code>	the profile complete log-likelihood of the fitted regression model.
<code>disparity</code>	the disparity of the fitted regression model.
<code>call</code>	the matched call.
<code>formula</code>	the formula provided.
<code>data</code>	the data argument.
<code>aic</code>	the Akaike information criterion of the fitted regression model.
<code>fitted</code>	the fitted values for the individual observations.
<code>fitted.transformed</code>	the fitted values for the individual transformed observations.
<code>residuals</code>	the difference between the observed values and the fitted values.
<code>residuals.transformed</code>	the difference between the transformed observed values and the transformed fitted values.
<code>predicted.re</code>	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 strength Data
data(strength, package = "mdscore")
maxlam <- optim.boxcox(y ~ cut*lot, data = strength, K = 3,
  start = "gq", find.in.range = c(-2, 2), s = 5)
# Maximum profile log-likelihood: 33.6795 at lambda= -0.4

data(Oxboys, package = "nlme")
Oxboys$boy <- gl(26,9)
maxlamvc <- optim.boxcox(height ~ age, groups = Oxboys$boy,
  data = Oxboys, K = 2, start = "gq",
  find.in.range=c(-1.2,1), s=6, plot.opt = 0)

maxlamvc$Maximum
#[1] -0.8333333
plot(maxlamvc,8)
```

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.boxcoxm()`, `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> . |
| 9 | works only for the <code>Kfind.boxcox()</code> function and plots the specified range of <code>K</code> against the AIC or BIC information criteria |
| 10 | works only for the <code>boxcoxmtype()</code> function and gives the profile likelihood function that summarises information concerning <code>lambda</code> for generalized linear Mixed-effects Models. |

print.boxcoxm

Summary of boxcoxm functions

Description

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

Usage

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

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

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

## S3 method for class 'boxcoxmixture'
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 boxcoxmixture 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 = 15,
  steps = 500,
  find.in.range = c(0, 1.5),
```

```

    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 EM trajectories and the development of the disparity over iteration number. And plot.opt=0, for none of them.
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 (0,1.5) 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.boxcoxmix() 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.
aic	the Akaike information criterion of the fitted regression model.
bic	the Bayesian information criterion of the fitted regression model.

Author(s)

Amani Almohaimeed and Jochen Einbeck

See Also

[np.boxcoxmix](#).

Examples

```
# The Pennsylvanian Hospital Stay Data
data(hosp, package = "nplmreg")
test1 <- tolfind.boxcox(duration ~ age , data = hosp, K = 2, lambda = 0,
  find.in.range = c(0, 2), s = 10, start = "gq")
# Minimal Disparity: 137.8368 at tol= 2
# Minimal Disparity with EM converged: 137.8368 at tol= 2

# Effect of Phenylbiguanide on Blood Pressure
data(PBG, package = "nlme")
test2 <- tolfind.boxcox(deltaBP ~ dose , groups = PBG$Rabbit, find.in.range = c(0, 2),
  data = PBG, K = 2, lambda = -1, s = 15, start = "quantile", plot.opt = 0)
test2$Mintol
# [1] 1.6
test2$MinDisparity
# [1] 449.5876
```

Index

- * **Kfind**
 - Kfind.boxcox, 5
- * **boxcoctype**
 - boxcoctype, 3
- * **boxcox**
 - Kfind.boxcox, 5
 - np.boxcoxmim, 7
 - optim.boxcox, 13
 - tolfind.boxcox, 18
- * **em**
 - np. estep, 10
- * **optim**
 - optim.boxcox, 13
- * **package**
 - boxcoxmim-package, 2
- * **random**
 - np. boxcoxmim, 7
- * **tolfind**
 - tolfind.boxcox, 18
- * **variance**
 - np. boxcoxmim, 7

- alldist, 4
- allvc, 4

- bhat (np. estep), 10
- binomial (boxcoctype), 3
- boxcoxmim (boxcoxmim-package), 2
- boxcoxmim-package, 2
- boxcoxpower (boxcoctype), 3
- boxcoctype, 3

- fik (np. estep), 10

- gqz (np. estep), 10

- Kfind.boxcox, 5, 5

- masspoint.class (np. estep), 10
- mik (np. estep), 10

- nb.se (np. estep), 10
- np.bhat (np. estep), 10
- np.boxcox (np. estep), 10
- np.boxcoxmim, 5, 7, 16, 20
- np.em (np. estep), 10
- np. estep, 10
- np.mstep (np. estep), 10
- np.theta (np. estep), 10
- np.zk (np. estep), 10

- optim.boxcox, 5, 9, 13

- plot, 16
- print.boxcoxmim, 17
- print.boxcoxmimpure (print.boxcoxmim), 17

- summary.boxcoxmim (print.boxcoxmim), 17
- summary.boxcoxmimpure (print.boxcoxmim), 17

- tolfind.boxcox, 5, 7, 9, 16, 18

- vc.boxcox (np. estep), 10
- vc.em (np. estep), 10
- vc. estep (np. estep), 10
- vc.mstep (np. estep), 10
- vc.se (np. estep), 10
- vc.theta (np. estep), 10

- yhat (np. estep), 10
- ytrans (np. estep), 10

- zk (np. estep), 10