Package 'BCSreg'

February 9, 2026

Type Package

Title Box-Cox Symmetric Regression for Non-Negative Data

Version 1.1.1

Description A collection of tools for regression analysis of non-negative data, including strictly positive and zero-inflated observations, based on the class of the Box-Cox symmetric (BCS) distributions and its zero-adjusted extension. The BCS distributions are a class of flexible probability models capable of describing different levels of skewness and tail-heaviness. The package offers a comprehensive regression modeling framework, including estimation and tools for evaluating goodness-of-fit.

License GPL (>= 3)

Encoding UTF-8

LazyData true

RoxygenNote 7.3.2

URL https://github.com/ffqueiroz/BCSreg

BugReports https://github.com/ffqueiroz/BCSreg/issues

Imports distr, EnvStats, Formula, gamlss.dist, GeneralizedHyperbolic

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

Depends R (>= 3.5)

NeedsCompilation no

Author Felipe Queiroz [aut] (ORCID: <https://orcid.org/0000-0001-8368-0707>), Rodrigo Medeiros [aut, cre, cph] (ORCID: <https://orcid.org/0000-0001-8101-3038>)

Maintainer Rodrigo Medeiros <rodrigo.matheus@ufrn.br>

Repository CRAN

Date/Publication 2026-02-09 16:00:02 UTC

Contents

BCS *The Box-Cox Symmetric Distributions*

Description

Density function, distribution function, quantile function, and random generation for the class of the Box-Cox symmetric (BCS) distributions.

Usage

```
dBCS(x, mu, sigma, lambda, zeta, family = "NO", log = FALSE)

pBCS(
  q,
  mu,
  sigma,
  lambda,
  zeta,
  family = "NO",
  lower.tail = TRUE,
  log.p = FALSE
)

qBCS(
  p,
  mu,
  sigma,
  lambda,
  zeta,
```

```
  family = "NO",
  lower.tail = TRUE,
  log.p = FALSE
)

rBCS(n, mu, sigma, lambda, zeta, family = "NO")
```

Arguments

x, q	vector of positive quantiles.
mu	vector of strictly positive scale parameters.
sigma	vector of strictly positive relative dispersion parameters.
lambda	vector of real-valued skewness parameters. If `lambda = 0`, the BCS distribution reduces to the corresponding log-symmetric distribution with parameters `mu` and `sigma` (and a possible extra parameter `zeta`).
zeta	strictly positive extra parameter. It must be specified with only one value in cases where the BCS distribution has an extra parameter. See "Details" below.
family	a character that specifies the symmetric generating family of the BCS distribution. Available options are: `"NO"` (default), `"ST"`, `"LOI"`, `"LOII"`, `"PE"`, `"SN"`, `"HP"`, and `"SL"`, corresponding to the normal, Student-*t*, type I logistic, type II logistic, power exponential, sinh-normal, hyperbolic, and slash distributions, respectively.
log, log.p	logical; if `TRUE`, probabilities p are given as `log(p)`. Default is `FALSE`.
lower.tail	logical; if `TRUE` (default), probabilities are $P(X \leq x)$ otherwise, $P(X > x)$.
p	vector of probabilities.
n	number of observations. If 'n' is a vector, its length is used as the number of required observations.

Details

The class of the BCS distributions was introduced by Ferrari and Fumes (2017). It consists of a broad class of probability models for positive continuous data, which includes flexible distributions with different levels of skewness and tail-heaviness.

The BCS class includes, as special cases, the Box-Cox *t* (Rigby and Stasinopoulos, 2006), Box-Cox normal (or Box-Cox Cole-Green; Cole and Green, 1992), Box-Cox power exponential (Rigby and Stasinopoulos, 2004) distributions, as well as the log-symmetric distributions (Vanegas and Paula, 2016).

Let Y be a positive continuous random variable with a BCS distribution with parameters $\mu > 0$, $\sigma > 0$, and $\lambda \in \mathbb{R}$ and density generating function r. The probability density function of Y is given by

$$f(y;\mu,\sigma,\lambda) = \begin{cases} \dfrac{y^{\lambda-1}}{\mu^{\lambda}\sigma}\dfrac{r(z^2)}{R\left(\frac{1}{\sigma|\lambda|}\right)}, & \text{if } \lambda \neq 0, \\ \dfrac{1}{y\sigma}r(z^2), & \text{if } \lambda = 0, \end{cases} \quad , \quad y > 0,$$

where $z = \begin{cases} \frac{1}{\sigma\lambda}\left\{\left(\frac{y}{\mu}\right)^{\lambda} - 1\right\}, & \text{if } \lambda \neq 0, \\ \frac{1}{\sigma}\log\left(\frac{y}{\mu}\right), & \text{if } \lambda = 0, \end{cases}$

$r : [0, \infty) \longrightarrow [0, \infty)$ satisfies $\int_0^{\infty} u^{-1/2} r(u)\mathrm{d}u = 1$, and $R(x) = \int_{-\infty}^{x} r(u^2)\mathrm{d}u, x \in \mathbb{R}$.

The function r is called density generating function, and it specifies the generating symmetric family of Y within the class of the BCS probability models. This function can also depend on extra parameters, such as the Box-Cox t and Box-Cox power exponential distributions. We call these extra parameters `zeta`. The currently available BCS distributions in the `BCSreg` package are listed below:

Distribution	**Family abbreviation**	**N. of extra parameters**
Box-Cox Hyperbolic	`"HP"`	1
Box-Cox Type I Logistic	`"LOI"`	0
Box-Cox Type II Logistic	`"LOII"`	0
Box-Cox Normal	`"NO"`	0
Box-Cox Power Exponential	`"PE"`	1
Box-Cox Sinh-Normal	`"SN"`	1
Box-Cox Slash	`"SL"`	1
Box-Cox *t*	`"ST"`	1

Value

`dBCS` returns the density function, `pBCS` gives the cumulative distribution function, `qBCS` provides the quantile function, and `rBCS` generates random variables.

Author(s)

Francisco F. de Queiroz <<felipeq@ime.usp.br>>

Rodrigo M. R. de Medeiros <<rodrigo.matheus@ufrn.br>>

References

Cole, T., and Green, P.J. (1992). Smoothing reference centile curves: the LMS method and penalized likelihood. *Statistics in medicine*, 11, 1305-1319.

Ferrari, S. L., and Fumes, G. (2017). Box-Cox symmetric distributions and applications to nutritional data. *AStA Advances in Statistical Analysis*, 101, 321-344.

Medeiros, R. M. R., and Queiroz, F. F. (2025). Flexible modeling of nonnegative continuous data: Box-Cox symmetric regression and its zero-adjusted extension.

Rigby, R. A., and Stasinopoulos, D. M. (2004). Smooth centile curves for skew and kurtotic data modelled using the Box-Cox power exponential distribution. *Statistics in medicine*, 23, 3053-3076.

Rigby, R. A., and Stasinopoulos, D. M. (2006). Using the Box-Cox t distribution in GAMLSS to model skewness and kurtosis. *Statistical Modelling*, 6, 209-229.

Vanegas, L. H., and Paula, G. A. (2016). Log-symmetric distributions: statistical properties and parameter estimation. *Brazilian Journal of Probability and Statistics*, 30, 196-220.

See Also

ZABCS to access the density function, distribution function, quantile function, and a random number generator for the zero-adjusted BCS distributions. BCSreg for estimating the parameters of a BCS regression model.

Examples

```
# Probability density function

## Right-skewed distributions
curve(dBCS(x, 3, 0.3, -1.5, family = "NO"), xlim = c(0, 7), ylim = c(0, 0.7), ylab = "Density")
curve(dBCS(x, 3, 0.3, -1.5, 4, family = "ST"), add = TRUE, col = 2)
curve(dBCS(x, 3, 0.3, -1.5, 5, family = "PE"), add = TRUE, col = 4)
legend("topright", legend = c("BCNO", "BCT", "BCPE"), lty = 1, col = c(1, 2, 4))

## Truncated symmetric distributions (with support on (0, Inf))
curve(dBCS(x, 3, 0.3, 1, family = "NO"), xlim = c(0, 7), ylim = c(0, 0.7), ylab = "Density")
curve(dBCS(x, 3, 0.3, 1, 4, family = "ST"), add = TRUE, col = 2)
curve(dBCS(x, 3, 0.3, 1, 5, family = "PE"), add = TRUE, col = 4)
legend("topright", legend = c("BCNO", "BCT", "BCPE"), lty = 1, col = c(1, 2, 4))

## Left-skewed distributions
curve(dBCS(x, 3, 0.3, 3, family = "NO"), xlim = c(0, 7), ylim = c(0, 0.7), ylab = "Density")
curve(dBCS(x, 3, 0.3, 3, 4, family = "ST"), add = TRUE, col = 2)
curve(dBCS(x, 3, 0.3, 3, 5, family = "PE"), add = TRUE, col = 4)
legend("topright", legend = c("BCNO", "BCT", "BCPE"), lty = 1, col = c(1, 2, 4))

# Random generation

## Parameter setting
mu <- 5        # scale parameter
sigma <- 0.2   # relative dispersion parameter
lambda <- -0.2 # skewness parameter

## Generating family
family <- "NO"

## Visualization
x <- rBCS(10000, mu, sigma, lambda, family = family)

hist(x, prob = TRUE, col = "white", main = "")
curve(dBCS(x, mu, sigma, lambda, family = family), col = "blue", add = TRUE)

plot(ecdf(x), main = "")
curve(pBCS(x, mu, sigma, lambda, family = family), col = "blue", add = TRUE)
```

BCSreg *Box-Cox Symmetric Regression for Positive Data*

Description

Fit a Box-Cox symmetric (BCS) or a zero-adjusted BCS regression model using maximum likelihood estimation.

Usage

```
BCSreg(
  formula,
  data,
  subset,
  na.action,
  family = "NO",
  zeta,
  link = "log",
  sigma.link = "log",
  alpha.link,
  control = BCSreg.control(...),
  model = FALSE,
  y = FALSE,
  x = FALSE,
  ...
)

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

Arguments

`formula`	a symbolic description of the model, allowing the specification of different regression structures for model parameters using the `Formula` package. By default, the formula defines the regression structure for the scale parameter. It can include up to three parts, separated by the '|' operator: • **First part:** specifies the model for the scale parameter. • **Second part (optional):** defines a regression structure for the relative dispersion parameter. • **Third part (only applicable for zero-adjusted positive data):** models the zero-adjustment parameter. See "Details" for further explanation.
`data`, `subset`, `na.action`	arguments controlling formula processing via `model.frame`.
`family`	a character that specifies the symmetric generating family of the BCS distribution. Available options are: `"NO"` (default), `"ST"`, `"LOI"`, `"LOII"`, `"PE"`, `"SN"`, `"HP"`, and `"SL"`, corresponding to the normal, Student-*t*, type I logistic, type II logistic, power exponential, sinh-normal, hyperbolic, and slash distributions, respectively.
`zeta`	Strictly positive extra parameter. It must be specified with a single value in cases where the BCS distribution has an extra parameter.

`link`, `sigma.link`
: character specification of the link functions for the scale and relative dispersion regression structures, respectively. Currently, `"log"` (default), `"sqrt"`, `"1/mu^2"`, `"inverse"`, and `"identity"` are supported.

`alpha.link`
: character specification of the link function for the zero-adjustment regression structure (only applicable for zero-inflated positive data). Currently, `"logit"` (default), `"probit"`, `"cloglog"`, `"cauchit"`, and `"identity"` are supported.

`control`
: a list of control parameters passed as arguments for the `optim` function specified via `BCSreg.control`.

`model`, `y`, `x`
: logicals. If `TRUE`, the corresponding components of the fit (the model frame, the response, and the model matrices, respectively) are returned. For `print()`, `x` is a fitted model object of class `"BCSreg"`.

`...`
: arguments passed to `BCSreg.control`. A particularly relevant argument is `lambda = 0`, which specifies the correspondent log-symmetric or zero-adjusted regression model.

`digits`
: a non-null value for digits specifies the minimum number of significant digits to be printed in values.

Details

The `BCSreg` function implements maximum likelihood estimation in the class of the BCS and zero-adjusted BCS regression models for the analysis of positive or zero-inflated data (Medeiros and Queiroz, 2025). The BCS distributions (Ferrari and Fumes, 2017) are a broad class of flexible distributions that achieve different levels of skewness and tail-heaviness. See details in BCS.

The distributions currently implemented in the `BCSreg` package, along with their abbreviations used in the `family` argument, are listed below:

Distribution	**Family abbreviation**	**N. of extra parameters**
Box-Cox Hyperbolic	`"HP"`	1
Box-Cox Type I Logistic	`"LOI"`	0
Box-Cox Type II Logistic	`"LOII"`	0
Box-Cox Normal	`"NO"`	0
Box-Cox Power Exponential	`"PE"`	1
Box-Cox Sinh-Normal	`"SN"`	1
Box-Cox Slash	`"SL"`	1
Box-Cox *t*	`"ST"`	1

The BCS distributions have at least three parameters: scale (`mu`), relative dispersion (`sigma`), and skewness (`lambda`) parameters. Some distributions may also depend on an additional parameter (`zeta`), such as the Box-Cox *t* and Box-Cox power exponential distributions. The BCS distributions reduce to the log-symmetric distributions (Vanegas and Paula, 2016) when `lambda` is fixed at zero. The log-symmetric distributions are an important class of probability models for positive data, which includes well-known distributions such as the log-normal and log-*t* distributions. The `BCSreg` function allows fitting a log-symmetric regression using `lambda = 0` as an argument (see `BCSreg.control`).

The BCSreg function also implements the class of zero-adjusted BCS regression models, for analyzing mixed data that is positive but can assume zero. In this case, the zero-adjusted BCS model corresponding to the family is fitted to the data. In addition to the parameters mu, sigma, lambda, and possibly zeta, the zero-adjusted BCS distributions have a zero-adjustment parameter (alpha) that corresponds to the probability of observing a zero response. This parameter can also be modeled with a regression structure.

The formula argument defines the regression structures for different model parameters using the Formula package (Zeileis and Croissant, 2010). It can have up to three parts, separated by the "|" operator:

- **First part:** specifies the model for the scale parameter.
- **Second part (optional):** defines a regression structure for the relative dispersion parameter.
- **Third part (only applicable for zero-inflated positive data):** models the zero-adjustment parameter.

If only the first part is provided, the model applies only to the scale parameter. When a second part is included, a regression structure is defined for the relative dispersion parameter. If the data contain zero inflation, a third part can be specified to model the zero-adjustment parameter. If the third part is provided but the response variable contains no zeros, this part is ignored.

For instance, consider a dataset where y is the zero-inflated dependent variable, and x, s, and z are explanatory variables associated with the scale, relative dispersion, and zero-adjustment parameters, respectively. The following formulas illustrate different model structures:

- y ~ x # Scale parameter only
- y ~ x | s # Scale and relative dispersion parameters
- y ~ x | s | z # Scale, relative dispersion, and zero-adjustment parameters
- y ~ x | 1 | z # Scale and zero-adjustment parameters
- y ~ 1 | s | z # Relative dispersion and zero-adjustment parameters

Value

The BCSreg function returns an object of class "BCSreg", which consists of a list with the following components:

coefficients a list containing the elements "mu" and "sigma" that consist of the estimates of the coefficients associated with the scale and relative dispersion regression structures, respectively. If the model is zero-adjusted, the element "alpha" will also be returned with the estimated coefficients for the regression structure of the zero-adjustment parameter.

fitted.values a vector with the fitted median responses. Not to be confused with the fitted values for mu.

mu a vector with the fitted scale parameters (or conditional scale parameters for a zero-adjusted fit).

sigma a vector with the fitted relative dispersion parameters (or conditional relative dispersion parameters for a zero-adjusted fit).

lambda the maximum likelihood estimate of the skewness parameter (lambda), or its fixed value specified in the BCSreg function.

zeta the specified value for the extra parameter of the corresponding BCS distribution, if applicable.

family the generating family of the fitted BCS distribution.

link a list with elements "mu" and "sigma" with the specified link functions for the mu and sigma regression structures, respectively. If the model is zero-adjusted, the element "alpha" will also be returned with the link function for the regression structure of the zero-adjustment parameter.

logLik log-likelihood of the fitted model.

vcov asymptotic covariance matrix of the estimators. By default, the asymptotic covariance matrix is based on a analytical expression of the observed information matrix. It can be obtained numerically based on the Hessian matrix via optim if the argument hessian = TRUE is used in the BCSreg function.

nobs number of observations.

df.null residual degrees of freedom in the null model (a model without any regression structure).

df.residual residual degrees of freedom in the fitted model, that is, the sample size minus the number of model parameters.

control the control arguments passed to the optim call.

start a vector with the starting values used in the iterative process.

optim a list with the output from optim.

converged logical indicating successful convergence of the iterative process.

call the original function call.

formula the formula used.

terms a list with elements "mu", "sigma", and "full" containing the term objects for the respective models. If the model is zero-adjusted, the element "alpha" will also be returned with the term object for the zero-adjustment model.

levels a list with elements "mu", "sigma", and "full" containing the levels of the categorical regressors. If the model is zero-adjusted, the element "alpha" will also be returned.

contrasts a list with elements "mu" and "sigma" containing the contrasts corresponding to levels from the respective models. If the model is zero-adjusted, the element "alpha" will also be returned.

model the full model frame (if y = TRUE).

y the response variable (if y = TRUE).

x a list with elements "mu" and "sigma" with the model matrices from the mu and sigma submodels (if x = TRUE). If the model is zero-adjusted, the element "alpha" will also be returned with the model matrix for the zero-adjustment submodel.

alpha a vector with the fitted zero-adjustment parameters when a zero-adjusted model is considered; and NULL, otherwise.

Author(s)

Francisco F. de Queiroz <<felipeq@ime.usp.br>>

Rodrigo M. R. de Medeiros <<rodrigo.matheus@ufrn.br>>

References

Cribari-Neto F, Zeileis A (2010). Beta Regression in R. *Journal of Statistical Software*, **34**, 1—24

Ferrari, S. L. P., and Fumes, G. (2017). Box-Cox symmetric distributions and applications to nutritional data. *AStA Advances in Statistical Analysis*, **101**, 321—344.

Medeiros, R. M. R., and Queiroz, F. F. (2025). Flexible modeling of nonnegative continuous data: Box-Cox symmetric regression and its zero-adjusted extension.

Vanegas, L. H., and Paula, G. A. (2016). Log-symmetric distributions: statistical properties and parameter estimation. *Brazilian Journal of Probability and Statistics*, **30**,196—220

Zeileis A, Croissant Y (2010). Extended Model Formulas in R: Multiple Parts and Multiple Responses. *Journal of Statistical Software*, **34**, 1—13.

See Also

`summary.BCSreg` for more detailed summaries, `residuals.BCSreg` to extract residuals from the fitted model, `plot.BCSreg` for diagnostic plots, and `extra.parameter` for selection of the extra parameter `zeta` via Upsilon goodness-of-fit statistic and profile likelihood. Information on additional methods for `"BCSreg"` objects can be found at `BCSreg-methods`.

Examples

```
# BCS regression for strictly positive response variables

## Data set: raycatch (for description, run ?raycatch)
hist(raycatch$cpue, xlab = "Catch per unit effort")
plot(cpue ~ tide_phase, raycatch, pch = 16,
     xlab = "Tide phase", ylab = "Catch per unit effort")
plot(cpue ~ location, raycatch, pch = 16,
     xlab = "Location", ylab = "Catch per unit effort")
plot(cpue ~ max_temp, raycatch, pch = 16,
     xlab = "Maximum temperature", ylab = "Catch per unit effort")

## Fit examples

### Fit a single Box-Cox normal regression model:
fit_bcno1 <- BCSreg(cpue ~ location + tide_phase + max_temp, raycatch)
fit_bcno1

### Fit a double Box-Cox normal regression model:
fit_bcno2 <- BCSreg(cpue ~ location + tide_phase |
                      location + tide_phase + max_temp, raycatch)
fit_bcno2


### Fit a double Box-Cox power exponential regression model (family = "PE"):
fit_bcpe <- BCSreg(cpue ~ location + tide_phase + max_temp |
                     location + tide_phase + max_temp, raycatch, family = "PE", zeta = 4)
fit_bcpe

### Fit a double log-power exponential regression model (lambda = 0):
fit_lpe <- BCSreg(cpue ~ location + tide_phase + max_temp |
```

```
                    location + tide_phase + max_temp, raycatch, family = "PE",
                  zeta = 4, lambda = 0)
fit_lpe

# Zero-adjusted BCS (ZABCS) regression for nonnegative response variables

## Data set: renewables2015 (for description, run ?renewables2015)
plot(ecdf(renewables2015$renew_elec_output), cex = 0.3, main = "Empirical CDF")
abline(h = mean(renewables2015$renew_elec_output == 0), col = "grey", lty = 3)
text(1250, 0.155, paste0("prop. of zeros: ~0.12"), col = "blue")

plot(renew_elec_output ~ adj_sav_edu, renewables2015, pch = 16,
     xlab = "Education expenditure (percent of GNI)",
     ylab = "Renewable electricity output (in TWh)")
plot(renew_elec_output ~ agri_land, renewables2015, pch = 16,
     xlab = "Matural logarithm of total agricultural land area",
     ylab = "Renewable electricity output (in TWh)")

## Fit examples

### Fit a single zero-adjusted Box-Cox normal regression model:
fit_zabcno1 <- BCSreg(renew_elec_output ~ adj_sav_edu + agri_land, renewables2015)
fit_zabcno1

### Fit a double zero-adjusted Box-Cox normal regression model:
fit_zabcno2 <- BCSreg(renew_elec_output ~ adj_sav_edu + agri_land |
                        adj_sav_edu + agri_land, renewables2015)
fit_zabcno2

### Fit a triple zero-adjusted Box-Cox normal regression model:
fit_zabcno3 <- BCSreg(renew_elec_output ~ adj_sav_edu + agri_land |
                        adj_sav_edu + agri_land |
                        adj_sav_edu + agri_land, renewables2015)
fit_zabcno3


### Fit a triple zero-adjusted Box-Cox power exponential regression model (family = "PE"):
fit_zabcpe <- BCSreg(renew_elec_output ~ adj_sav_edu + agri_land |
                      adj_sav_edu + agri_land |
                      adj_sav_edu + agri_land, renewables2015, family = "PE", zeta = 4)
fit_zabcpe

### Fit a triple zero-adjusted log-power exponential regression model (lambda = 0):
fit_zalpe <- BCSreg(renew_elec_output ~ adj_sav_edu + agri_land |
                    adj_sav_edu + agri_land |
                    adj_sav_edu + agri_land, renewables2015, family = "PE",
                  zeta = 4, lambda = 0)
fit_zalpe
```

BCSreg-methods *Extract Information From a Box-Cox Symmetric Regression Fit*

Description

Methods for "BCSreg" objects.

Usage

```
## S3 method for class 'BCSreg'
model.frame(formula, ...)

## S3 method for class 'BCSreg'
model.matrix(object, model = c("mu", "sigma", "alpha"), ...)

## S3 method for class 'BCSreg'
coef(object, model = c("mu", "sigma", "alpha", "full"), ...)

## S3 method for class 'BCSreg'
vcov(object, model = c("mu", "sigma", "alpha", "full"), ...)

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

## S3 method for class 'ugrpl'
AIC(object, ..., k = 2)
```

Arguments

formula	a model formula or terms object or an R object.
...	further arguments passed to or from other methods.
object	an object of class "BCSreg", a result of a call to BCSreg.
model	a character indicating which regression structure should be used. It can be "mu" for the scale regression structure, "sigma" for the relative dispersion regression structure, or "full" (when applicable) for both regression structures.
k	numeric, the penalty per parameter to be used; the default k = 2 is the classical Akaike information criteria (AIC).

Value

- model.frame returns a data.frame containing the variables required by formula and any additional arguments provided via ....
- model.matrix returns the design matrix used in the regression structure, as specified by the model argument.
- coef returns a numeric vector of estimated regression coefficients, based on the model argument. If parm = "full", it returns a list with the components "mu" and "sigma", each containing the corresponding coefficient estimates. If the model is zero-adjusted, it will also have a "alpha" component with the estimates of the associated regression coefficients.
- vcov returns the asymptotic covariance matrix of the regression coefficients, based on the model argument.

- `logLik` returns the log-likelihood value of the fitted model.
- `AIC` returns a numeric value representing the Akaike Information Criterion (AIC), Bayesian Information Criterion, or another criterion, depending on `k`.

Author(s)

Francisco F. de Queiroz <<felipeq@ime.usp.br>>

Rodrigo M. R. de Medeiros <<rodrigo.matheus@ufrn.br>>

References

Ferrari, S. L. P., and Fumes, G. (2017). Box-Cox symmetric distributions and applications to nutritional data. *AStA Advances in Statistical Analysis*, **101**, 321—344.

Medeiros, R. M. R., and Queiroz, F. F. (2025). Flexible modeling of nonnegative continuous data: Box-Cox symmetric regression and its zero-adjusted extension.

Examples

```
## Data set: renewables2015 (for description, run ?renewables2015)
plot(ecdf(renewables2015$renew_elec_output), cex = 0.3, main = "Empirical CDF")
abline(h = mean(renewables2015$renew_elec_output == 0), col = "grey", lty = 3)
text(1250, 0.155, paste0("prop. of zeros: ~0.12"), col = "blue")

plot(renew_elec_output ~ adj_sav_edu, renewables2015, pch = 16,
    xlab = "Education expenditure (percent of GNI)",
    ylab = "Renewable electricity output (in TWh)")
plot(renew_elec_output ~ agri_land, renewables2015, pch = 16,
    xlab = "Matural logarithm of total agricultural land area",
    ylab = "Renewable electricity output (in TWh)")

 ## Fit a zero-adjusted Box-Cox normal regression
 fit <- BCSreg(renew_elec_output ~ adj_sav_edu + agri_land |
                 adj_sav_edu + agri_land |
                 adj_sav_edu + agri_land, renewables2015)

 ## coef
 coef(fit)                  # regression coefficients of the scale submodel
 coef(fit, model = "sigma") # regression coefficients of the relative dispersion submodel
 coef(fit, model = "alpha") # regression coefficients of the zero-adjustment submodel
 coef(fit, model = "full")  # all regression coefficients

 ## vcov
 vcov(fit)                  # covariance matrix for the scale submodel coefficients
vcov(fit, model = "sigma") # covariance matrix for the relative dispersion submodel coefficients
vcov(fit, model = "alpha") # covariance matrix for the zero-adjustment submodel coefficients
vcov(fit, model = "full")  # full covariance matrix of the model (including the skewness parameter)

 ## Log-likelihood value
 logLik(fit)

 ## AIC and BIC
```

```
AIC(fit)
AIC(fit, k = log(fit$nobs))

## Model matrices
model.matrix(fit)                 # design matrix for the scale submodel
model.matrix(fit, model = "sigma") # design matrix for the relative dispersion submodel
model.matrix(fit, model = "alpha") # design matrix for the zero-adjustment submodel
```

BCSregcontrol *Auxiliary for Controlling a Box-Cox Symmetric Fitting*

Description

Optimization parameters that control the fitting of Box-Cox symmetric or zero-adjusted Box-Cox symmetric regression models using the BCSreg function.

Usage

```
BCSreg.control(
  lambda = NULL,
  method = "BFGS",
  maxit = 2000,
  hessian = FALSE,
  trace = FALSE,
  start = NULL,
  ...
)
```

Arguments

lambda	numeric indicating the value of lambda (if NULL, lambda will be estimated).
method	character specifying the method argument passed to optim.
maxit, trace, ...	arguments passed to optim.
hessian	logical. Should the numerical Hessian matrix from the optim output be used for estimation of the covariance matrix. By default the analytical solution is employed.
start	an optional vector with starting values for the regression coefficients associated with the mu and sigma submodels (starting value for the lambda or the zero-adjustment parameters must not be included).

Details

The BCSreg.control controls the fitting process of Box-Cox symmetric models. Almost all the arguments are passed directly to optim, which is used to estimate the parameters. Starting values for the regression coefficients associated with the mu and sigma submodels can be provided via start argument. The starting value for the skewness parameter lambda is zero; that is, the fitting

process starts from the corresponding log-symmetric regression model. If the estimation process is to be performed with a fixed `lambda`, a value must be specified for the `lambda` argument. A natural value is `lambda = 0`, where a log-symmetric regression model will be estimated.

When a regression fit with a zero-adjusted BCS distribution is performed, the regression coefficients associated with the zero-adjustment parameter are estimated using the `glm` function.

Value

A list with components named as the arguments.

Author(s)

Francisco F. de Queiroz <<felipeq@ime.usp.br>>

Rodrigo M. R. de Medeiros <<rodrigo.matheus@ufrn.br>>

See Also

`BCSreg`

Examples

```
## Data set: raycatch (for description, run ?raycatch)
hist(raycatch$cpue, xlab = "Catch per unit effort")
plot(cpue ~ tide_phase, raycatch,
     xlab = "Tide phase", ylab = "Catch per unit effort")

### Fit a Box-Cox normal regression model:
fit_bcno <- BCSreg(cpue ~ tide_phase, raycatch)
fit_bcno

### Fit a log-normal regression model (setting lambda = 0):
fit_lno <- BCSreg(cpue ~ tide_phase, raycatch, lambda = 0)
fit_lno

### Standard errors obtained from the numerical Hessian matrix
### provided by optim instead of the analytical solution:
fit_bcno2 <- BCSreg(cpue ~ tide_phase, raycatch, hessian = TRUE)
fit_lno2 <- BCSreg(cpue ~ tide_phase, raycatch, lambda = 0, hessian = TRUE)

# In this case, there are differences only in the eighth decimal place:
vcov(fit_bcno)
vcov(fit_bcno2)

# In this case, there are no differences:
vcov(fit_lno)
vcov(fit_lno2)
```

`education` *Household Expenditures on Basic Education (POF 2017–2018, São Paulo)*

Description

A dataset containing household-level information on expenditures on basic education and associated socioeconomic characteristics, derived from the 2017–2018 Brazilian Consumer Expenditure Survey (Pesquisa de Orçamentos Familiares, POF), conducted by the Instituto Brasileiro de Geografia e Estatística (IBGE).

This dataset is used to illustrate regression modeling for zero-adjusted data with a substantial proportion of zero observations. The response variable corresponds to total household expenditure on basic education, measured over the 12 months preceding the interview and assigned to the household reference person. Expenditures include childcare, preschool, regular primary and secondary education, youth and adult education, and supplementary (equivalency) programs at the primary and secondary levels.

Households reporting no expenditure on basic education are assigned a value of zero. The final sample consists of 4,232 households residing in the state of São Paulo, Brazil. Approximately 93% of the observations correspond to zero expenditure, indicating a highly zero-inflated distribution.

Usage

```
data(education)
```

Format

A data frame with 4,232 observations and 11 variables:

expense Total household expenditure on basic education (in BRL) over the 12 months preceding the interview. Values equal to zero indicate no reported expenditure.

residence_type Type of household residence, categorized as `"Urban"` or `"Rural"`.

age Age of the household reference person, in years.

sex Sex of the household reference person, coded as `"Male"` or `"Female"`.

race Self-reported race or ethnicity of the reference person, according to IBGE classification.

health_plan Indicator of whether the reference person is covered by a private health plan (`"Yes"` or `"No"`).

literacy Literacy status of the reference person (`"Yes"` or `"No"`).

years_schooling Number of completed years of formal education of the reference person.

education_level Highest educational level attained by the reference person.

income_pc Per capita disposable household income (in BRL), calculated as total disposable household income divided by the number of residents. Disposable income includes monetary and non-monetary earnings, net of direct taxes, social contributions, and other mandatory deductions.

sons Number of children living in the household, including children of the reference person and/or the spouse.

Details

In each household, the POF survey designates a reference person, typically responsible for financial and administrative decisions. All individual-level covariates refer to this reference person. Expenditure values are aggregated at the household level but attributed to the reference person for modeling purposes.

Summary statistics indicate that the median education expenditure is zero, while the mean expenditure is BRL 108.8, reflecting the presence of a small number of households with substantially higher spending levels. The maximum observed expenditure is BRL 48,000.

Source

Instituto Brasileiro de Geografia e Estatística (IBGE). Pesquisa de Orçamentos Familiares (POF) 2017–2018. Available (in Portuguese) at: https://www.ibge.gov.br/estatisticas/sociais/populacao/24786-pesquisa-de-orcamentos-familiares-2.html

Examples

```
data(education)
summary(education)
```

`envelope` *Normal Probability Plots with Simulated Envelope for a Box-Cox Symmetric Regression Fit*

Description

Produce the normal probability plot with simulated envelope of the quantile residuals obtained from a Box-Cox symmetric regression fit.

Usage

```
envelope(object, rep = 60, conf = 0.95, envcol, ...)
```

Arguments

`object`	a fitted model object of class "`BCSreg`".
`rep`	a positive integer representing the number of iterations to calculate the simulated envelopes. Default is `rep = 60`.
`conf`	a numeric value in the interval (0,1) that represents the confidence level of the simulated envelope. Default is `conf = 0.95`.
`envcol`	character specifying the color of the envelope.
`...`	additional graphical parameters (see par).

Details

The envelope uses the idea of Atkinson (1985) to create normal probability plots with simulated envelope. Under the correct model, approximately 100 * conf of the residuals are expected to be inside the envelope.

Currently, the envelope() function, when used in a zero-adjusted Box-Cox symmetric regression fit, returns only the quantile plot for the quantile residuals obtained under a combined approach (see residuals.BCSreg).

Value

envelope returns normal probability plot with simulated envelopes for the quantile residuals.

Author(s)

Francisco F. de Queiroz <<felipeq@ime.usp.br>>

Rodrigo M. R. de Medeiros <<rodrigo.matheus@ufrn.br>>

References

Atkinson, A. C. (1985). *Plots, Transformations and Regression: An Introduction to Graphical Methods of Diagnostic Regression Analysis*. Oxford Science Publications, Oxford.

Medeiros, R. M. R., and Queiroz, F. F. (2025). Flexible modeling of nonnegative continuous data: Box-Cox symmetric regression and its zero-adjusted extension.

See Also

BCSreg, residuals.BCSreg

Examples

```
## Data set: raycatch (for description, run ?raycatch)
hist(raycatch$cpue, xlab = "Catch per unit effort")
plot(cpue ~ tide_phase, raycatch, pch = 16,
   xlab = "Tide phase", ylab = "Catch per unit effort")
plot(cpue ~ location, raycatch, pch = 16,
   xlab = "Location", ylab = "Catch per unit effort")
plot(cpue ~ max_temp, raycatch, pch = 16,
   xlab = "Maximum temperature", ylab = "Catch per unit effort")

## Fit a double Box-Cox normal regression model:
fit <- BCSreg(cpue ~ location + tide_phase |
               location + tide_phase + max_temp, raycatch)
envelope(fit)
```

`extra.parameter` *Select the Extra Parameter of a Box-Cox Symmetric Regression Model*

Description

Estimation of the extra parameter in a Box-Cox symmetric or zero-adjusted Box-Cox symmetric regression model based on the Upsilon goodness-of-fit statistic and the profile log-likelihood.

Usage

```
extra.parameter(
  object,
  family,
  grid = seq(1, 30, 2),
  trace = TRUE,
  plot = TRUE,
  control = BCSreg.control(...),
  ...
)

## S3 method for class 'extra.parameter'
print(x, ...)

## S3 method for class 'extra.parameter'
plot(x, which = 1, ...)
```

Arguments

`object`	an object of class `"BCSreg"`, resulting from a call to `BCSreg`.
`family`	a character string specifying the symmetric generating family of the BCS distribution. The available options are: `"ST"`, `"PE"`, `"SN"`, `"HP"`, and `"SL"`. The families `"NO"`, `"LOI"`, and `"LOII"` do not depend on an additional parameter.
`grid`	a numeric vector of positive values at which the Upsilon statistic and the profile log-likelihood function will be evaluated.
`trace`	logical; if `TRUE`, a summary displaying the profile log-likelihood values and the Upsilon statistics is shown.
`plot`	logical; if `TRUE`, a plot of the Upsilon statistics evaluated over the specified grid of values is displayed.
`control`	a list of control arguments specified via `BCSreg.control`.
`...`	further arguments passed to `BCSreg.control`.
`x`	an object of class `"extra.parameter"`.
`which`	numeric; if `which = 1`, the plot shows the Upsilon statistic versus `zeta`; if `which = 2`, the plot shows the profile log-likelihood versus `zeta`.

Value

An object of class "extra.parameter", which is a list containing the fits of the BCS regression model for each value of the extra parameter zeta specified in grid. The object also includes:

- logLik: a vector with the log-likelihood values for each fit;
- Upsilon: a vector with the Upsilon statistic values for each fit;
- grid: the specified grid of values.

The value of the extra parameter (zeta) can be selected using two alternative approaches:

- the value that minimizes the Upsilon goodness-of-fit statistic;
- the value that maximizes the log-likelihood.

The print method summarizes the fits by displaying, for each value in grid, the corresponding log-likelihood value and Upsilon statistic. The plot method returns a graph of the Upsilon statistics, highlighting its minimum.

Author(s)

Francisco F. de Queiroz <<felipeq@ime.usp.br>>

Rodrigo M. R. de Medeiros <<rodrigo.matheus@ufrn.br>>

References

Medeiros, R. M. R., and Queiroz, F. F. (2025). Flexible modeling of nonnegative continuous data: Box-Cox symmetric regression and its zero-adjusted extension.

Examples

```
## Data set: raycatch (for description, run ?raycatch)
hist(raycatch$cpue, xlab = "Catch per unit effort")
plot(cpue ~ tide_phase, raycatch, pch = 16,
   xlab = "Tide phase", ylab = "Catch per unit effort")
plot(cpue ~ location, raycatch, pch = 16,
   xlab = "Location", ylab = "Catch per unit effort")
plot(cpue ~ max_temp, raycatch, pch = 16,
   xlab = "Maximum temperature", ylab = "Catch per unit effort")

## Fit the Box-Cox normal regression as a reference model
fit_bcno <- BCSreg(cpue ~ location + tide_phase |
                location + tide_phase + max_temp, raycatch)

## Use the specifications of the reference model to change the distribution
## to, for example, Box-Cox t, and select the value of the extra parameter:
select_bct <- extra.parameter(fit_bcno, family = "ST", grid = 1:20)

## Class
class(select_bct)

## It is possible to recover the plots:
plot(select_bct)
```

```
plot(select_bct, which = 2)

## and the trace:
select_bct

## Selected fit based on the Upsilon statistic
fit_bct <- select_bct[["zeta = 19"]]
summary(fit_bct)
```

`influence` *Influence Diagnostics for BCSreg Objects*

Description

The `influence` function provides two influence measures for a Box-Cox symmetric or a zero-adusted Box-Cox symmetric regression fit.

Usage

```
influence(object, plot = TRUE, ask = grDevices::dev.interactive(), ...)
```

Arguments

`object`	an object of class `"BCSreg"`.
`plot`	logical. If `plot = TRUE` (default), the plots are shown.
`ask`	logical; if `TRUE`, the user is asked before each plot, if `plot = TRUE`.
`...`	currently not used.

Value

`influence` returns a list with two objects:

`case.weights`	The values of d_{max} eigenvector based on case weights perturbation scheme (see Medeiros and Queiroz (2025)).
`totalLI`	The total local influence (see Lesaffre and Verbeke (1998)).

Author(s)

Francisco F. de Queiroz <<felipeq@ime.usp.br>>

Rodrigo M. R. de Medeiros <<rodrigo.matheus@ufrn.br>>

References

Lesaffre, E., and Verbeke, G. (1998). Local influence in linear mixed models. *Biometrics*, 570–582.

Medeiros, R. M. R., and Queiroz, F. F. (2025). Flexible modeling of nonnegative continuous data: Box-Cox symmetric regression and its zero-adjusted extension.

See Also

BCSreg for parameter estimation in the class of the Box-Cox symmetric or zero-adjusted Box-Cox symmetric regression models, residuals.BCSreg for extracting residuals for a fitted model, and plot.BCSreg for diagnostic plots.

Examples

```
## Data set: raycatch (for description, run ?raycatch)
hist(raycatch$cpue, xlab = "Catch per unit effort")
plot(cpue ~ tide_phase, raycatch, pch = 16,
   xlab = "Tide phase", ylab = "Catch per unit effort")
plot(cpue ~ location, raycatch, pch = 16,
   xlab = "Location", ylab = "Catch per unit effort")
plot(cpue ~ max_temp, raycatch, pch = 16,
   xlab = "Maximum temperature", ylab = "Catch per unit effort")

## Fit a double Box-Cox normal regression model:
fit <- BCSreg(cpue ~ location + tide_phase |
               location + tide_phase + max_temp, raycatch)

## Influence measures under case-weights perturbation scheme:
cw <- influence(fit) ## two index plots are shown
str(cw)
```

---

plot.BCSreg *Diagnostic Plots for a Box-Cox Symmetric Regression Fit*

Description

This function provides plots for diagnostic analysis of a Box-Cox symmetric or a zero-adjusted regression fit.

Usage

```
## S3 method for class 'BCSreg'
plot(
  x,
  which = 1:4,
  ask = prod(graphics::par("mfcol")) < length(which) && grDevices::dev.interactive(),
  pch = "+",
  las = 1,
  cex = 0.8,
  lwd = 2,
  ...
)
```

Arguments

x	an object of class "BCSreg".
which	numeric; if a subset of the plots is required, specify a subset of the numbers 1:7.
ask	logical; if TRUE, the user is asked before each plot.
pch, las, cex, lwd, ...	graphical parameters (see par)

Details

The plot method for BCSreg objects provides seven types of diagnostic plots in the following order:

Residuals vs fitted values a plot of the residuals versus the fitted medians.

Residuals vs observation indices. an index plot of the residuals versus the observation indices.

Density plot a graph that compares the empirical density of the residuals with the density of the standard normal distribution.

Normal probability plot a normal probability plot of the residuals with a confidence region constructed according to Fox (2016) using the qqPlot function.

Case-weight perturbation An index plot of local influence based on the case-weight perturbation scheme.

Fitted vs observed values a dispersion diagram of the fitted values versus the observed values.

Residuals vs v(z) function a dispersion diagram of the $v(z)$ function versus the residuals. For some BCS models, the $v(z)$ function may be interpreted as weights in the estimation process. If family = "NO", the $v(z)$ function is constant.

The which argument can be used to select a subset of the implemented plots. Default is which = 1:4. See residuals.BCSreg for details on the residuals.

Value

plot method for "BCSreg" objects returns seven types of diagnostic plots.

Author(s)

Francisco F. de Queiroz <<felipeq@ime.usp.br>>

Rodrigo M. R. de Medeiros <<rodrigo.matheus@ufrn.br>>

References

Medeiros, R. M. R., and Queiroz, F. F. (2025). Flexible modeling of nonnegative continuous data: Box-Cox symmetric regression and its zero-adjusted extension.

Examples

```
## Data set: raycatch (for description, run ?raycatch)
hist(raycatch$cpue, xlab = "Catch per unit effort")
plot(cpue ~ tide_phase, raycatch, pch = 16,
    xlab = "Tide phase", ylab = "Catch per unit effort")
plot(cpue ~ location, raycatch, pch = 16,
    xlab = "Location", ylab = "Catch per unit effort")
plot(cpue ~ max_temp, raycatch, pch = 16,
    xlab = "Maximum temperature", ylab = "Catch per unit effort")

## Fit a double Box-Cox normal regression model:
fit <- BCSreg(cpue ~ location + tide_phase |
                location + tide_phase + max_temp, raycatch)

## Available plots:

### Residuals vs fitted values (fitted medians)
plot(fit, which = 1)

### Residuals vs observation indices
plot(fit, which = 2)

### Density plot
plot(fit, which = 3)

### Normal probability plot
plot(fit, which = 4)

### Local influence
plot(fit, which = 5)

### Fitted medians vs response
plot(fit, which = 6)

### v(z) function
plot(fit, which = 7)
```

---

raycatch *Fishing Data from Todos-os-Santos Bay*

Description

This dataset contains information on white ray landings using the traditional "grozeira" fishing method in Baía de Todos os Santos, Brazil. Data were collected from January 2012 to January 2013 by Marion (2015). The study aimed to analyze the relationship between catch per unit effort (cpue) and various environmental factors.

Usage

```
data(raycatch)
```

Format

A data frame with 186 rows and 8 variables:

period fishing period, categorized as "Dry" or "Rainy".

location fishing location, categorized as "Area 1", "Area 2", "Area 3", or "Area 4".

wind_speed wind speed (m/s).

tide_phase tide phase, categorized as "Quadrature" or "Spring tide".

max_temp maximum temperature (°C).

min_temp minimum temperature (°C).

sunshine_duration duration of sunshine (hours).

cpue represents the catch per unit effort, calculated as the productivity (in grams) divided by the product of the number of hooks and the soak time (in hours).

Details

In Brazil, marine fish are harvested through various fishing activities, with artisanal fishing playing a particularly prominent role, especially in the state of Bahia, in the northeastern region of the country. In this region, particularly within the Baía de Todos os Santos, most of the marine production comes from small-scale operations. Among the species landed and sold, rays constitute a significant portion of the catch. The data analyzed in this application were collected by Marion (2015) over a 13-month period (January 2012 to January 2013), based on 231 fishing trips that employed the grozeira (a type of bottom longline) as gear. After eliminating missing data, the data contains information on 186 fish landings. The objective is to identify key factors influencing ray catch levels. The response variable is the catch per unit effort (cpue), defined as

$$\text{cpue} = \frac{\text{Productivity (g)}}{\text{Nº of hooks} \times \text{Immersion time (hours)}}.$$

Source

Data collected from a field study on traditional ray fishing in Baía de Todos os Santos, Brazil.

References

Marion, C. (2015). *Função da Baía de Todos os Santos, Bahia, no ciclo de vida de Dasyatis guttata (Chondrichthyes: Dasyatidae)*. Doctoral dissertation, University of São Paulo, Institute of Oceanography. Retrieved from https://teses.usp.br/teses/disponiveis/21/21134/tde-22072015-154346/pt-br.php

Paula, G. A., and Kumagaia, G. H. *Relatório de análise estatística sobre o projeto: "Variabilidade espaço-temporal da captura da raia branca, Dasyattis guttata, na pesca artesanal da Baía de Todos os Santos, Bahia"*. São Paulo, IME-USP, 2014.

Examples

```
data(raycatch)
summary(raycatch)
plot(cpue ~ tide_phase, raycatch,
     xlab = "Tide phase", ylab = "CPUE (kg)",
     main = "Effect of tide phase on CPUE")
```

`renewables2015` *Renewable Electricity Output for 186 Countries (2015)*

Description

This dataset provides information on renewable electricity output and a set of socioeconomic, environmental, and institutional variables for 186 countries in the year 2015. The data were obtained from the World Bank (https://data.worldbank.org/).

Usage

```
data(renewables2015)
```

Format

A data frame with 186 rows and 9 variables:

`country_name` country name.

`adj_sav_edu` adjusted savings: education expenditure (percent of GNI).

`agri_land` natural logarithm of total agricultural land area (in sq. km).

`elec_fossil` electricity production from fossil fuels (percent of total). Indicates the share of electricity generated from oil, gas, and coal, representing reliance on non-renewable energy.

`forest_area` forest area (percent of land area). Reflects natural resource availability.

`gov_effec` government effectiveness index (range: approximately -2.5 to 2.5). Measures public service quality, policy implementation, and government credibility.

`pop_density` natural logarithm of population density (people per sq. km).

`access_elec` access to electricity (percent of population).

`renew_elec_output` renewable electricity output (in TWh).

Details

The response variable is the renewable electricity output, measured in terawatt-hours (TWh), representing the total electricity generated from renewable sources such as wind, solar photovoltaic, solar thermal, hydro, marine, geothermal, solid biofuels, renewable municipal waste, liquid biofuels, and biogas. Hydro pumped storage is excluded.

Source

https://data.worldbank.org/

Examples

```
data(renewables2015)
summary(renewables2015)
pairs(renewables2015[, -1], pch = 16, cex = 0.8, col = rgb(0,0,0, 0.5))
```

`residuals` *Extract Residuals for a Box-Cox Symmetric Regression Fit*

Description

Residuals resulting from fitting a Box-Cox symmetric or a zero-adjusted Box-Cox symmetric regression.

Usage

```
## S3 method for class 'BCSreg'
residuals(object, approach = c("combined", "separated"), ...)
```

Arguments

`object`	an object of class `"BCSreg"`, a result of a call to BCSreg.
`approach`	a character string indicating the approach for calculating residuals when a zero-adjusted regression is fitted. Should be either `"combined"` (default) for combined residuals or `"separated"` for separate residuals. Ignored if the model is not zero-adjusted.
`...`	further arguments passed to or from other methods.

Details

For a Box-Cox symmetric regression fit, the residuals are the quantile residuals (Dunn and Smyth, 1996), defined by $r_i^q = \Phi^{-1}(\widehat{F}(y_i))$, where $\widehat{F}(\cdot)$ is the fitted cumulative distribution function and $\Phi(\cdot)$ is cumulative distribution function of the standard normal distribution.

For zero-adjusted Box-Cox symmetric regressions, two approaches are available:

- **Combined approach**: Returns a single vector of residuals defined as
  $$r_i^q = \begin{cases} \Phi^{-1}(u_i), & y_i = 0, \\ \Phi^{-1}\left[\widehat{F}^{(0)}(y_i)\right], & y_i > 0, \end{cases}$$
  where u_i is a random variable uniformly distributed in $(0, \widehat{\alpha}_i]$ and $F^{(0)}$ is the fitted cumulative distribution function of the mixed response.
- **Separated approach**: Returns a list containing:
  - Quantile residuals for the positive (continuous) component.
  - Standardized Pearson residuals for the discrete component, defined by
    $r_i^p = \frac{\mathbb{I}(y_i=0)-\widehat{\alpha}_i}{\sqrt{\widehat{\alpha}_i(1-\widehat{\alpha}_i)(1-\widehat{h}_{ii})}}$,
    where $\widehat{h}_{ii}$ is the ith diagonal element of the "hat matrix" resulting from a fit of a generalized linear model with a binary response given by $\mathbb{I}(y_i = 0)$, being $\mathbb{I}$ the indicator function.

See more details in Medeiros and Queiroz (2025).

Value

If a Box-Cox symmetric regression is fitted to the data, it returns a numeric vector containing the quantile residuals (Dunn and Smyth, 1996).

If the model is a zero-adjusted Box-Cox symmetric regression:

- For `approach = "combined"`, it returns a numeric vector with "combined" quantile residuals. See details
- For `approach = "separated"`, it returns a list with two components: `continuous` (quantile residuals for strictly positive responses) and `discrete` (standardized Pearson residuals for the discrete component).

Author(s)

Francisco F. de Queiroz <<felipeq@ime.usp.br>>

Rodrigo M. R. de Medeiros <<rodrigo.matheus@ufrn.br>>

References

Dunn, P. K. and Smyth, G. K. (1996). Randomized quantile residuals. *Journal of Computational and Graphical Statistics*, **5**, 236—244.

Medeiros, R. M. R., and Queiroz, F. F. (2025). Flexible modeling of nonnegative continuous data: Box-Cox symmetric regression and its zero-adjusted extension.

Examples

```
# BCS regression for strictly positive response variables

## Data set: raycatch (for description, run ?raycatch)
hist(raycatch$cpue, xlab = "Catch per unit effort")
plot(cpue ~ tide_phase, raycatch, pch = 16,
     xlab = "Tide phase", ylab = "Catch per unit effort")
plot(cpue ~ location, raycatch, pch = 16,
     xlab = "Location", ylab = "Catch per unit effort")
plot(cpue ~ max_temp, raycatch, pch = 16,
     xlab = "Maximum temperature", ylab = "Catch per unit effort")

## BCS fit
fit <- BCSreg(cpue ~ location + tide_phase + max_temp |
                location + tide_phase + max_temp, raycatch)

## Quantile residuals
rq <- residuals(fit)
rq

## Normal probability plot
qqnorm(rq, pch = "+", cex = 0.8)
qqline(rq, col = "dodgerblue", lwd = 2)

# Zero-adjusted BCS (ZABCS) regression for nonnegative response variables
```

```
## Data set: renewables2015 (for description, run ?renewables2015)
plot(ecdf(renewables2015$renew_elec_output), cex = 0.3, main = "Empirical CDF")
abline(h = mean(renewables2015$renew_elec_output == 0), col = "grey", lty = 3)
text(1250, 0.155, paste0("prop. of zeros: ~0.12"), col = "blue")

plot(renew_elec_output ~ adj_sav_edu, renewables2015, pch = 16,
     xlab = "Education expenditure (percent of GNI)",
     ylab = "Renewable electricity output (in TWh)")
plot(renew_elec_output ~ agri_land, renewables2015, pch = 16,
     xlab = "Matural logarithm of total agricultural land area",
     ylab = "Renewable electricity output (in TWh)")

## Zero-adjusted BCS fit
fit0 <- BCSreg(renew_elec_output ~ adj_sav_edu + agri_land |
                 adj_sav_edu + agri_land | adj_sav_edu + agri_land, renewables2015)

## Combined approach (default)
rq <- residuals(fit0)
rq

### Normal probability plot
qqnorm(rq, pch = "+", cex = 0.8)
qqline(rq, col = "dodgerblue", lwd = 2)

## Separated approach
res <- residuals(fit0, approach = "separated")
str(res)

### Normal probability plots

# Continuous part
qqnorm(res$continuous, pch = "+", cex = 0.8)
qqline(res$continuous, col = "dodgerblue", lwd = 2)

# Discrete part (Pearson's standardized residuals do not have a normal distribution.)
qqnorm(res$discrete, pch = "+", cex = 0.8)
qqline(res$discrete, col = "dodgerblue", lwd = 2)
```

---

summary.BCSreg *Summarizing a Box-Cox Symmetric Regression Fit*

Description

summary method for class "BCSreg".

Usage

```
## S3 method for class 'BCSreg'
summary(object, ...)
```

```
## S3 method for class 'summary.BCSreg'
print(x, digits = getOption("digits"), ...)
```

Arguments

`object`	an object of class `"BCSreg"`, a result of a call to BCSreg.
`...`	further arguments passed to or from other methods.
`x`	an object of class `"summary.BCSreg"`, a result of a call to summary.BCSreg.
`digits`	a non-null value for digits specifies the minimum number of significant digits to be printed in values.

Details

An object of class `"summary.BCSreg"` provides additional information from a Box-Cox symmetric or zero-adjusted Box-Cox symmetric regression fit. In addition to summary tables with estimates, standard errors, and individual significance tests of the regression coefficients, it also provides the quantile residuals (Dunn and Smyth, 1996) and goodness-of-fit measures.

The goodness-of-fit measures include:

- A pseudo-R_p^2 defined based on Ferrari and Cribari-Neto (2004) as the square of the sample correlation coefficient between $d_1(y)$ and $d_1(\widehat{y})$, where y is the response variable, $\widehat{y}$ is the adjusted median, and $d_1()$ is the link function (`link`) used in the regression structure for the scale parameter. By definition, $0 \leq R_p^2 \leq 1$, and perfect agreement between the fitted values and the response yields $R_p^2 = 1$.
- A general goodness-of-fit measure based on Vanegas and Paula (2016), defined as

$$\Upsilon_\zeta = \frac{1}{n}\sum_{i=1}^{n}\left|\Phi^{-1}[\widehat{F}(y_{(i)})] - \nu_{(i)}\right|,$$

  where $y_{(i)}$ is the ith order statistic of the response, $\nu_{(i)}$ is the expected value of the ith order statistic from a standard normal sample of size n, $\Phi(\cdot)$ denotes the cumulative distribution function of the standard normal distribution, and $\widehat{F}(\cdot)$ denotes the fitted cumulative distribution function of the assumed distribution for the response.
- A weighting function $v(\cdot)$ that plays a role in the parameter estimation process, providing information about the individual contribution of each observation. It depends on the generating density function and is constant for distributions generated by the normal family (`family = "NO"`).

Value

The function `summary.BCSreg` returns an object of class `"summary.BCSreg"`, which consists of a list with the following components:

call the original function call, given in `object`.

mu summary statistics for the scale submodel.

sigma summary statistics for the relative dispersion submodel.

lambda summary statistics for the skewness parameter. If this parameter is not statistically different from zero, the fitted Box-Cox symmetric (BCS) distribution can be reduced to a more parsimonious log-symmetric distribution.

alpha summary statistics for the zero-adjustment submodel when a zero-adjusted model is considered; and `NULL`, otherwise.

zeta the specified value for the extra parameter of the fitted BCS distribution, if applicable.

family the generating family of the fitted BCS distribution.

link a list with elements `"mu"` and `"sigma"` with the specified link functions for the `mu` and `sigma` regression structures, respectively. If the model is zero-adjusted, the element `"alpha"` will also be returned with the link function for the regression structure of the zero-adjustment parameter.

converged logical indicating successful convergence of the iterative process.

iterations number of iterations reached until the optimization algorithm converges.

logLik log-likelihood of the fitted model.

df number of model parameters

residuals a vector of quantile residuals.

pseudo.r.squared pseudo R-squared value.

Upsilon.zeta an overall goodness-of-fit measure.

v a vector with the $v(z)$ values for all the observations.

AIC, BIC Akaike and Bayesian information criteria.

Author(s)

Francisco F. de Queiroz <<felipeq@ime.usp.br>>

Rodrigo M. R. de Medeiros <<rodrigo.matheus@ufrn.br>>

References

Dunn, P. K. and Smyth, G. K. (1996). Randomized quantile residuals. *Journal of Computational and Graphical Statistics*, **5**, 236—244.

Ferrari, S., and Cribari-Neto, F. (2004). Beta regression for modelling rates and proportions. *Journal of Applied Statistics*, **31**, 799—815.

Medeiros, R. M. R., and Queiroz, F. F. (2025). Flexible modeling of nonnegative continuous data: Box-Cox symmetric regression and its zero-adjusted extension.

Vanegas, L. H., and Paula, G. A. (2016). Log-symmetric distributions: statistical properties and parameter estimation. *Brazilian Journal of Probability and Statistics*, **30**,196—220

Examples

```
# BCS regression for strictly positive response variables

## Data set: raycatch (for description, run ?raycatch)
hist(raycatch$cpue, xlab = "Catch per unit effort")
plot(cpue ~ tide_phase, raycatch, pch = 16,
```

```
     xlab = "Tide phase", ylab = "Catch per unit effort")
plot(cpue ~ location, raycatch, pch = 16,
     xlab = "Location", ylab = "Catch per unit effort")
plot(cpue ~ max_temp, raycatch, pch = 16,
     xlab = "Maximum temperature", ylab = "Catch per unit effort")

## Fit examples

### Fit a single Box-Cox normal regression model:
fit_bcno1 <- BCSreg(cpue ~ location + tide_phase + max_temp, raycatch)
summary(fit_bcno1)

# Other quantities can be obtained from a summary.BCSreg object:
aux <- summary(fit_bcno1)
class(aux)
str(aux)

### Fit a double Box-Cox normal regression model:
fit_bcno2 <- BCSreg(cpue ~ location + tide_phase |
                      location + tide_phase + max_temp, raycatch)
summary(fit_bcno2)


### Fit a double Box-Cox power exponential regression model (family = "PE"):
fit_bcpe <- BCSreg(cpue ~ location + tide_phase + max_temp |
                     location + tide_phase + max_temp, raycatch, family = "PE", zeta = 4)
summary(fit_bcpe)

### Fit a double log-power exponential regression model (lambda = 0):
fit_lpe <- BCSreg(cpue ~ location + tide_phase + max_temp |
                    location + tide_phase + max_temp, raycatch, family = "PE",
                  zeta = 4, lambda = 0)
summary(fit_lpe)

# Zero-adjusted BCS (ZABCS) regression for nonnegative response variables

## Data set: renewables2015 (for description, run ?renewables2015)
plot(ecdf(renewables2015$renew_elec_output), cex = 0.3, main = "Empirical CDF")
abline(h = mean(renewables2015$renew_elec_output == 0), col = "grey", lty = 3)
text(1250, 0.155, paste0("prop. of zeros: ~0.12"), col = "blue")

plot(renew_elec_output ~ adj_sav_edu, renewables2015, pch = 16,
     xlab = "Education expenditure (percent of GNI)",
     ylab = "Renewable electricity output (in TWh)")
plot(renew_elec_output ~ agri_land, renewables2015, pch = 16,
     xlab = "Matural logarithm of total agricultural land area",
     ylab = "Renewable electricity output (in TWh)")

## Fit examples

### Fit a single zero-adjusted Box-Cox normal regression model:
fit_zabcno1 <- BCSreg(renew_elec_output ~ adj_sav_edu + agri_land, renewables2015)
summary(fit_zabcno1)
```

```
# Other quantities obtained from a zero-adjusted fit:
aux <- summary(fit_zabcno1)
str(aux)

### Fit a double zero-adjusted Box-Cox normal regression model:
fit_zabcno2 <- BCSreg(renew_elec_output ~ adj_sav_edu + agri_land |
                        adj_sav_edu + agri_land, renewables2015)
summary(fit_zabcno2)

### Fit a triple zero-adjusted Box-Cox normal regression model:
fit_zabcno3 <- BCSreg(renew_elec_output ~ adj_sav_edu + agri_land |
                        adj_sav_edu + agri_land |
                        adj_sav_edu + agri_land, renewables2015)
summary(fit_zabcno3)


### Fit a triple zero-adjusted Box-Cox power exponential regression model (family = "PE"):
fit_zabcpe <- BCSreg(renew_elec_output ~ adj_sav_edu + agri_land |
                       adj_sav_edu + agri_land |
                       adj_sav_edu + agri_land, renewables2015, family = "PE", zeta = 4)
summary(fit_zabcpe)

### Fit a triple zero-adjusted log-power exponential regression model (lambda = 0):
fit_zalpe <- BCSreg(renew_elec_output ~ adj_sav_edu + agri_land |
                      adj_sav_edu + agri_land |
                      adj_sav_edu + agri_land, renewables2015, family = "PE",
                    zeta = 4, lambda = 0)
summary(fit_zalpe)
summary(fit_lpe)
```

ZABCS *The Zero-Adjusted Box-Cox Symmetric Distributions*

Description

Density function, distribution function, quantile function, and random generation for the class of the zero-adjusted Box-Cox symmetric (ZABCS) distributions.

Usage

```
dZABCS(x, alpha, mu, sigma, lambda, zeta, family = "NO", log = FALSE)

pZABCS(
  q,
  alpha,
  mu,
  sigma,
  lambda,
```

```
  zeta,
  family = "NO",
  lower.tail = TRUE,
  log.p = FALSE
)

qZABCS(
  p,
  alpha,
  mu,
  sigma,
  lambda,
  zeta,
  family = "NO",
  lower.tail = TRUE,
  log.p = FALSE
)

rZABCS(n, alpha, mu, sigma, lambda, zeta, family = "NO")
```

Arguments

x, q	vector of positive quantiles.
alpha	vector of zero-adjusted parameters, with values on (0, 1).
mu	vector of strictly positive scale parameters.
sigma	vector of strictly positive relative dispersion parameters.
lambda	vector of real-valued skewness parameters. If `lambda = 0`, the BCS distribution reduces to the corresponding log-symmetric distribution with parameters `mu` and `sigma` (and a possible extra parameter `zeta`).
zeta	strictly positive extra parameter. It must be specified with only one value in cases where the BCS distribution has an extra parameter. See "Details" below.
family	a character that specifies the symmetric generating family of the BCS distribution. Available options are: `"NO"` (default), `"ST"`, `"LOI"`, `"LOII"`, `"PE"`, `"SN"`, `"HP"`, and `"SL"`, corresponding to the normal, Student-*t*, type I logistic, type II logistic, power exponential, sinh-normal, hyperbolic, and slash distributions, respectively.
log, log.p	logical; if `TRUE`, probabilities `p` are given as `log(p)`. Default is `FALSE`.
lower.tail	logical; if `TRUE` (default), probabilities are $P(X \leq x)$ otherwise, $P(X > x)$.
p	vector of probabilities.
n	number of observations. If 'n' is a vector, its length is used as the number of required observations.

Details

The class of the ZABCS distributions was introduced by Medeiros and Queiroz (2025) as an extension of the Box-Cox symmetric (BCS) distributions (Ferrari and Fumes, 2017). The models consists of a broad class of probability distributions for positive continuous data which may include zeros.

Let Y be a positive continuous random variable with a ZABCS distribution with parameters $\alpha \in (0,1)$, $\mu > 0$, $\sigma > 0$, and $\lambda \in \mathbb{R}$ and density generating function r. The probability density function of Y is given by

$$f^{(0)}(y;\alpha,\mu,\sigma,\lambda) = \begin{cases} \alpha, & y = 0, \\ (1-\alpha) f(y;\alpha,\mu,\sigma,\lambda), & y > 0, \end{cases}$$

where

$$f(y;\mu,\sigma,\lambda) = \begin{cases} \dfrac{y^{\lambda-1}}{\mu^{\lambda}\sigma} \dfrac{r(z^2)}{R\left(\frac{1}{\sigma|\lambda|}\right)}, & \text{if } \lambda \neq 0, \\ \dfrac{1}{y\sigma} r(z^2), & \text{if } \lambda = 0, \end{cases} \quad , \quad y > 0,$$

with

$$z = \begin{cases} \dfrac{1}{\sigma\lambda}\left\{\left(\frac{y}{\mu}\right)^{\lambda} - 1\right\}, & \text{if } \lambda \neq 0, \\ \dfrac{1}{\sigma}\log\left(\frac{y}{\mu}\right), & \text{if } \lambda = 0, \end{cases}$$

$r : [0,\infty) \longrightarrow [0,\infty)$ satisfies $\int_0^\infty u^{-1/2} r(u)\mathrm{d}u = 1$, and $R(x) = \int_{-\infty}^{x} r(u^2)\mathrm{d}u, x \in \mathbb{R}$.

The function r is called density generating function, and it specifies the generating symmetric family of Y within the class of the ZABCS probability models. This function can also depend on extra parameters, such as the zero-adjusted Box-Cox t and zero-adjusted Box-Cox power exponential distributions. We call these extra parameters `zeta`. The currently available ZABCS distributions in the `BCSreg` package are listed below:

Distribution	**Family abbreviation**	**N. of extra parameters**
Zero-adjusted Box-Cox Hyperbolic	`"HP"`	1
Zero-adjusted Box-Cox Type I Logistic	`"LOI"`	0
Zero-adjusted Box-Cox Type II Logistic	`"LOII"`	0
Zero-adjusted Box-Cox Normal	`"NO"`	0
Zero-adjusted Box-Cox Power Exponential	`"PE"`	1
Zero-adjusted Box-Cox Sinh-Normal	`"SN"`	1
Zero-adjusted Box-Cox Slash	`"SL"`	1
Zero-adjusted Box-Cox *t*	`"ST"`	1

Value

`dZABCS` returns the density function, `pZABCS` gives the cumulative distribution function, `qZABCS` provides the quantile function, and `rZABCS` generates random variables.

Author(s)

Francisco F. de Queiroz <<felipeq@ime.usp.br>>

Rodrigo M. R. de Medeiros <<rodrigo.matheus@ufrn.br>>

References

Ferrari, S. L. P., and Fumes, G. (2017). Box-Cox symmetric distributions and applications to nutritional data. *AStA Advances in Statistical Analysis*, 101, 321-344.

Medeiros, R. M. R., and Queiroz, F. F. (2025). Flexible modeling of nonnegative continuous data: Box-Cox symmetric regression and its zero-adjusted extension.

See Also

BCS to access the density function, distribution function, quantile function, and a random number generator for the BCS distributions. BCSreg for estimating the parameters of a ZABCS regression model.

Examples

```
# Probability density function

## Right-skewed distributions
curve(dZABCS(x, 0.4, 3, 0.3, -1.5, family = "NO"), from = 0.001, to = 7,
      xlim = c(0, 7), ylim = c(0, 0.5), ylab = "Density")
curve(dZABCS(x, 0.4, 3, 0.3, -1.5, 4, family = "ST"), add = TRUE, col = 2, from = 0.001)
curve(dZABCS(x, 0.4, 3, 0.3, -1.5, 5, family = "PE"), add = TRUE, col = 4, from = 0.001)
points(0, 0.4, type = "h", lty = 2)
points(0, 0.4, pch = 16, lty = 2)
legend("topright", legend = c("BCNO", "BCT", "BCPE"), lty = 1, col = c(1, 2, 4))

## Truncated symmetric distributions (with support on (0, Inf))
curve(dZABCS(x, 0.4, 3, 0.3, 1, family = "NO"), from = 0.001, to = 7,
      xlim = c(0, 7), ylim = c(0, 0.5), ylab = "Density")
curve(dZABCS(x, 0.4, 3, 0.3, 1, 4, family = "ST"), add = TRUE, col = 2, from = 0.001)
curve(dZABCS(x, 0.4, 3, 0.3, 1, 5, family = "PE"), add = TRUE, col = 4, from = 0.001)
points(0, 0.4, type = "h", lty = 2)
points(0, 0.4, pch = 16, lty = 2)
legend("topright", legend = c("BCNO", "BCT", "BCPE"), lty = 1, col = c(1, 2, 4))

## Left-skewed distributions
curve(dZABCS(x, 0.4, 3, 0.3, 3, family = "NO"), from = 0.001, to = 7,
      xlim = c(0, 7), ylim = c(0, 0.5), ylab = "Density")
curve(dZABCS(x, 0.4, 3, 0.3, 3, 4, family = "ST"), add = TRUE, col = 2, from = 0.001)
curve(dZABCS(x, 0.4, 3, 0.3, 3, 5, family = "PE"), add = TRUE, col = 4, from = 0.001)
points(0, 0.4, type = "h", lty = 2)
points(0, 0.4, pch = 16, lty = 2)
legend("topright", legend = c("BCNO", "BCT", "BCPE"), lty = 1, col = c(1, 2, 4))


# Random generation

## Parameter setting
alpha <- 0.2   # zero-adjustment parameter
mu <- 5        # scale parameter
sigma <- 0.2   # relative dispersion parameter
lambda <- -0.2 # skewness parameter

## Generating family
family <- "NO"
```

```
## Visualization
x <- rZABCS(10000, alpha, mu, sigma, lambda, family = family)

hist(x, prob = TRUE, col = "white", main = "")

points(0, mean(x == 0), type = "h", lty = 2)
points(0, mean(x == 0), pch = 16, lty = 2)
curve(dZABCS(x, alpha, mu, sigma, lambda, zeta, family = family), col = "blue", add = TRUE)

plot(ecdf(x), main = "")
curve(pZABCS(x, alpha, mu, sigma, lambda, zeta, family = family), col = "blue", add = TRUE)
```

Index