Package 'bellreg'

May 7, 2026

Title Count Regression Models Based on the Bell Distribution

Version 0.0.2.2

Description Bell regression models for count data with overdispersion. The implemented models account for ordinary and zero-inflated regression models under both frequentist and Bayesian approaches. Theoretical details regarding the models implemented in the package can be found in Castellares et al. (2018) <doi:10.1016/j.apm.2017.12.014> and Lemonte et al. (2020) <doi:10.1080/02664763.2019.1636940>.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.2

Biarch true

URL https://github.com/fndemarqui/bellreg, https://fndemarqui.github.io/bellreg/

BugReports https://github.com/fndemarqui/bellreg/issues

Depends R (>= 3.4.0)

Imports dplyr, extraDistr, Formula, magic, MASS, methods, numbers, LambertW, loo, purrr, Rcpp (>= 0.12.0), Rdpack, rstan (>= 2.26.0), rstantools (>= 2.0.0)

RdMacros Rdpack

LinkingTo BH (>= 1.66.0), Rcpp (>= 0.12.0), RcppEigen (>= 0.3.3.3.0), rstan (>= 2.26.0), StanHeaders (>= 2.26.0)

SystemRequirements GNU make

Suggests knitr, rmarkdown, testthat

VignetteBuilder knitr

NeedsCompilation yes

Author Fabio Demarqui [aut, cre, cph] (ORCID: <https://orcid.org/0000-0001-9236-1986>), Marcos Prates [ctb] (ORCID: <https://orcid.org/0000-0001-8077-4898>), Fredy Caceres [ctb], Andrew Johnson [ctb]

Maintainer Fabio Demarqui <fndemarqui@est.ufmg.br>

Repository CRAN

Date/Publication 2024-10-23 08:50:02 UTC

Contents

`bellreg-package` *The 'bellreg' package.*

Description

Bell Regression models for count data with overdispersion. The implemented models account for ordinary and zero-inflated regression models under both frequentist and Bayesian approaches. Theorical details regarding the models implemented in the package can be found in (Castellares et al. 2018) and (Lemonte et al. 2020)

_PACKAGE

References

Stan Development Team (2020). RStan: the R interface to Stan. R package version 2.19.3. https://mc-stan.org

Castellares F, Ferrari SL, Lemonte AJ (2018). "On the Bell distribution and its associated regression model for count data." *Applied Mathematical Modelling*, **56**, 172 - 185. doi:10.1016/j.apm.2017.12.014.

Lemonte AJ, Moreno-Arenas G, Castellares F (2020). "Zero-inflated Bell regression models for count data." *Journal of Applied Statistics*, **47**(2), 265-286. doi:10.1080/02664763.2019.1636940.

`AIC.bellreg` *Akaike information criterion*

Description

Akaike information criterion

Usage

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

Arguments

`object`	an object of the class bellreg.
`...`	further arguments passed to or from other methods.
`k`	numeric, the penalty per parameter to be used; the default k = 2 is the classical AIC.

Value

the Akaike information criterion value when a single model is passed to the function; otherwise, a data.frame with the Akaike information criterion values and the number of parameters is returned.

Examples

```
library(bellreg)
data(faults)
fit1 <- bellreg(nf ~ 1, data = faults, approach = "mle")
fit2 <- bellreg(nf ~ lroll, data = faults, approach = "mle")
AIC(fit1, fit2)
```

AIC.zibellreg *Akaike information criterion for zibellreg objects*

Description

Akaike information criterion for zibellreg objects

Usage

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

Arguments

object	an object of the class zibellreg.
...	further arguments passed to or from other methods.
k	numeric, the penalty per parameter to be used; the default k = 2 is the classical AIC.

Value

the Akaike information criterion value when a single model is passed to the function; otherwise, a data.frame with the Akaike information criterion values and the number of parameters is returned.

Examples

```
library(bellreg)
data(cells)
fit1 <- zibellreg(cells ~ 1|1, data = cells, approach = "mle")
fit2 <- zibellreg(cells ~ 1|smoker+gender, data = cells, approach = "mle")
fit3 <- zibellreg(cells ~ smoker+gender|smoker+gender, data = cells, approach = "mle")
AIC(fit1, fit2, fit3)
```

bell *Family Objects for Models*

Description

Family objects provide a convenient way to specify the details of the models used by functions such as glm. See the documentation for glm for the details on how such model fitting takes place.

Usage

```
bell(link = "log")
```

Arguments

link	a specification for the model link function. This can be a name/expression, a literal character string, a length-one character vector, or an object of class "link-glm" (such as generated by make.link) provided it is not specified *via* one of the standard names given next. The gaussian family accepts the links (as names) identity, log and inverse; the binomial family the links logit, probit, cauchit, (corresponding to logistic, normal and Cauchy CDFs respectively) log and cloglog (complementary log-log); the Gamma family the links inverse, identity and log; the poisson family the links log, identity, and sqrt; and the inverse.gaussian family the links 1/mu^2, inverse, identity and log. The quasi family accepts the links logit, probit, cloglog, identity, inverse, log, 1/mu^2 and sqrt, and the function power can be used to create a power link function.

Details

family is a generic function with methods for classes "glm" and "lm" (the latter returning gaussian()).

For the binomial and quasibinomial families the response can be specified in one of three ways:

1. As a factor: 'success' is interpreted as the factor not having the first level (and hence usually of having the second level).
2. As a numerical vector with values between 0 and 1, interpreted as the proportion of successful cases (with the total number of cases given by the weights).
3. As a two-column integer matrix: the first column gives the number of successes and the second the number of failures.

The quasibinomial and quasipoisson families differ from the binomial and poisson families only in that the dispersion parameter is not fixed at one, so they can model over-dispersion. For the binomial case see McCullagh and Nelder (1989, pp. 124–8). Although they show that there is (under some restrictions) a model with variance proportional to mean as in the quasi-binomial model, note that glm does not compute maximum-likelihood estimates in that model. The behaviour of S is closer to the quasi- variants.

Value

An object of class "family" (which has a concise print method). This is a list with elements

family	character: the family name.
link	character: the link name.
linkfun	function: the link.
linkinv	function: the inverse of the link function.
variance	function: the variance as a function of the mean.
dev.resids	function giving the deviance for each observation as a function of (y, mu, wt), used by the residuals method when computing deviance residuals.

aic	function giving the AIC value if appropriate (but NA for the quasi- families). More precisely, this function returns $-2\ell + 2s$, where ℓ is the log-likelihood and s is the number of estimated scale parameters. Note that the penalty term for the location parameters (typically the "regression coefficients") is added elsewhere, e.g., in glm.fit(), or AIC(), see the AIC example in glm. See logLik for the assumptions made about the dispersion parameter.
mu.eta	function: derivative of the inverse-link function with respect to the linear predictor. If the inverse-link function is $\mu = g^{-1}(\eta)$ where η is the value of the linear predictor, then this function returns $d(g^{-1})/d\eta = d\mu/d\eta$.
initialize	expression. This needs to set up whatever data objects are needed for the family as well as n (needed for AIC in the binomial family) and mustart (see glm).
validmu	logical function. Returns TRUE if a mean vector mu is within the domain of variance.
valideta	logical function. Returns TRUE if a linear predictor eta is within the domain of linkinv.
simulate	(optional) function simulate(object, nsim) to be called by the "lm" method of simulate. It will normally return a matrix with nsim columns and one row for each fitted value, but it can also return a list of length nsim. Clearly this will be missing for 'quasi-' families.
dispersion	(optional since R version 4.3.0) numeric: value of the dispersion parameter, if fixed, or NA_real_ if free.

Note

The link and variance arguments have rather awkward semantics for back-compatibility. The recommended way is to supply them as quoted character strings, but they can also be supplied unquoted (as names or expressions). Additionally, they can be supplied as a length-one character vector giving the name of one of the options, or as a list (for link, of class "link-glm"). The restrictions apply only to links given as names: when given as a character string all the links known to make.link are accepted.

This is potentially ambiguous: supplying link = logit could mean the unquoted name of a link or the value of object logit. It is interpreted if possible as the name of an allowed link, then as an object. (You can force the interpretation to always be the value of an object via logit[1].)

Author(s)

The design was inspired by S functions of the same names described in Hastie & Pregibon (1992) (except quasibinomial and quasipoisson).

References

McCullagh P. and Nelder, J. A. (1989) *Generalized Linear Models.* London: Chapman and Hall.

Dobson, A. J. (1983) *An Introduction to Statistical Modelling.* London: Chapman and Hall.

Cox, D. R. and Snell, E. J. (1981). *Applied Statistics; Principles and Examples.* London: Chapman and Hall.

Hastie, T. J. and Pregibon, D. (1992) *Generalized linear models.* Chapter 6 of *Statistical Models in S* eds J. M. Chambers and T. J. Hastie, Wadsworth & Brooks/Cole.

See Also

glm, power, make.link.

For binomial *coefficients*, choose; the binomial and negative binomial *distributions*, Binomial, and NegBinomial.

Examples

```
library(bellreg)
data(faults)
fit <- glm(nf ~ lroll, data = faults, family = bell("log"))
summary(fit)
```

Belldist *Probability function, distribution function, quantile function and random generation for the Bell distribution with parameter theta.*

Description

Probability function, distribution function, quantile function and random generation for the Bell distribution with parameter theta.

Usage

```
dbell(x, theta, log = FALSE)

pbell(q, theta, lower.tail = TRUE, log.p = FALSE)

qbell(p, theta, log.p = FALSE)

rbell(n, theta)
```

Arguments

x	vector of (non-negative integer) quantiles.
theta	parameter of the Bell distribution (theta > 0).
log, log.p	logical; if TRUE, probabilities p are given as log(p).
q	vector of quantiles.
lower.tail	logical; if TRUE (default), probabilities are $P[X \leq x]$; otherwise, $P[X > x]$.
p	vector of probabilities.
n	number of random values to return.

Details

Probability mass function

$$f(x) = \frac{\theta^x e^{1-e^\theta} B_x}{x!},$$

where B_x is the Bell number, and x = 0, 1, ...

Value

dbell gives the (log) probability function, pbell gives the (log) distribution function, qbell gives the quantile function, and rbell generates random deviates.

`bellreg` *Bell regression model*

Description

Fits the Bell regression model to overdispersed count data.

Usage

```
bellreg(
  formula,
  data = NULL,
  approach = c("mle", "bayes"),
  hessian = TRUE,
  link = c("log", "sqrt", "identity"),
  hyperpars = list(mu_beta = 0, sigma_beta = 10),
  ...
)
```

Arguments

`formula`	an object of class "formula" (or one that can be coerced to that class): a symbolic description of the model to be fitted.
`data`	an optional data frame, list or environment (or object coercible by as.data.frame to a data frame) containing the variables in the model. If not found in data, the variables are taken from environment(formula), typically the environment from which ypbp is called.
`approach`	approach to be used to fit the model (mle: maximum likelihood; bayes: Bayesian approach).
`hessian`	hessian logical; If TRUE (default), the hessian matrix is returned when approach="mle".
`link`	assumed link function (log, sqrt or identiy); default is log.
`hyperpars`	a list containing the hyperparameters associated with the prior distribution of the regression coefficients; if not specified then default choice is hyperpars = c(mu_beta = 0, sigma_beta = 10).
`...`	further arguments passed to either `rstan::optimizing` or `rstan::sampling`.

Value

bellreg returns an object of class "bellreg" containing the fitted model.

Examples

```
data(faults)
# ML approach:
mle <- bellreg(nf ~ lroll, data = faults, approach = "mle")
summary(mle)

# Bayesian approach:
bayes <- bellreg(nf ~ lroll, data = faults, approach = "bayes", refresh = FALSE)
summary(bayes)
```

---

cells *Cells data set*

Description

Data set taken from (Crawley 2012) and posteriorly analyzed by (Lemonte et al. 2020). The data includes the count of infected blood cells per square millimetre on microscope slides prepared from n = 511 randomly selected individuals.

Format

A data frame with 511 rows and 5 variables:

- cells: count of infected blood cells per square millimetre on microscope slides
- smoker: smoking status of the subject (0: smoker; 1: non smoker)
- gender: subject's gender (1: male; 0: female).
- age: subject's age categorized into three levels: young ($\leq$ 20), mid (21 to 59), and old ($\geq$ 60).
- weight: body mass score categorized into three levels: normal, overweight, obese.

References

Crawley MJ (2012). *The R Book*, 2nd edition. Wiley Publishing. ISBN 0470973927.

Lemonte AJ, Moreno-Arenas G, Castellares F (2020). "Zero-inflated Bell regression models for count data." *Journal of Applied Statistics*, **47**(2), 265-286. doi:10.1080/02664763.2019.1636940.

coef.bellreg *Estimated regression coefficients for the bellreg model*

Description

Estimated regression coefficients for the bellreg model

Usage

```
## S3 method for class 'bellreg'
coef(object, ...)
```

Arguments

object	an object of the class bellreg.
...	further arguments passed to or from other methods.

Value

a vector with the estimated regression coefficients.

Examples

```
data(faults)
fit <- bellreg(nf ~ lroll, data=faults)
coef(fit)
```

coef.zibellreg *Estimated regression coefficients for zibellreg model*

Description

Estimated regression coefficients for zibellreg model

Usage

```
## S3 method for class 'zibellreg'
coef(object, ...)
```

Arguments

object	an object of the class bellreg
...	further arguments passed to or from other methods

Value

a list containing the the estimated regression coefficients associated with the degenerated and Bell count distributions, respectively.

Examples

```
data(cells)
fit <- zibellreg(cells ~ smoker + gender|smoker + gender, data = cells)
coef(fit)
```

`confint.bellreg` *Confidence intervals for the regression coefficients*

Description

Confidence intervals for the regression coefficients

Usage

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

Arguments

`object`	an object of the class bellreg
`parm`	a specification of which parameters are to be given confidence intervals, either a vector of numbers or a vector of names. If missing, all parameters are considered.
`level`	the confidence level required
`...`	further arguments passed to or from other methods

Value

A matrix (or vector) with columns giving lower and upper confidence limits for each parameter. These will be labelled as (1-level)/2 and 1 - (1-level)/2 in \

Examples

```
data(faults)
fit <- bellreg(nf ~ lroll, data = faults)
confint(fit)
```

confint.zibellreg *Confidence intervals for the regression coefficients*

Description

Confidence intervals for the regression coefficients

Usage

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

Arguments

object	an object of the class zibellreg
parm	a specification of which parameters are to be given confidence intervals, either a vector of numbers or a vector of names. If missing, all parameters are considered.
level	the confidence level required
...	further arguments passed to or from other methods

Value

100(1-alpha)% confidence intervals for the regression coefficients

Examples

```
data(cells)
fit <- zibellreg(cells ~ smoker+gender|smoker+gender, data = cells, approach = "mle")
confint(fit)
```

extract_log_lik *Extract pointwise log-likelihood from a Stan model for a bellreg model*

Description

This function extracts the pointwise log-likelihood for a bellreg model.

Usage

```
extract_log_lik(object, ...)
```

Arguments

object	an object of the class bellreg.
...	further arguments passed to or from other methods.

Value

a matrix with the pointwise extracted log-likelihood associated with a bellreg model.

Examples

```
data(faults)
fit <- bellreg(nf ~ lroll, data = faults, approach = "bayes")
loglik <- extract_log_lik(fit)

data(cells)
fit <- zibellreg(cells ~ 1|smoker+gender, data = cells, approach = "bayes", chains = 1, iter = 100)
loglik <- extract_log_lik(fit)
```

---

faults *Faults data set*

Description

Data set taken from ( ) and posteriorly analyzed by (Castellares et al. 2018). The data contains the number of faults in rolls of fabric of different lengths.

Format

A data frame with 32 rows and 2 variables:

- nf: number of faults in rolls of fabric of different lengths.
- lroll: length of the roll.

References

Castellares F, Ferrari SL, Lemonte AJ (2018). "On the Bell distribution and its associated regression model for count data." *Applied Mathematical Modelling*, **56**, 172 - 185. doi:10.1016/j.apm.2017.12.014.

Hind J (ed.) (1982). *Compound Poisson Regression Models*, volume 14 of *Lecture Notes in Statistics*. ISBN 978-0-387-90777-2, doi:10.1007/9781461257714_11.

fitted.bellreg *Extract Model Fitted Values*

Description

This function returns the fitted values.

Usage

```
## S3 method for class 'bellreg'
fitted(object, ...)
```

Arguments

object	an object of the class bellreg.
...	further arguments passed to or from other methods.

Value

a vector with the fitted values (for MLE approach) or a matrix containing the posterior sample of the fitted values.

Examples

```
data(faults)
fit <- bellreg(nf ~ lroll, data = faults)
fitted.values(fit)
```

print.summary.bellreg *Print the summary.bellreg output*

Description

Print the summary.bellreg output

Usage

```
## S3 method for class 'summary.bellreg'
print(x, ...)
```

Arguments

x	an object of the class summary.bellreg.
...	further arguments passed to or from other methods.

Value

a summary of the fitted model.

`print.summary.zibellreg` *Print the summary.zibellreg output*

Description

Print the summary.zibellreg output

Usage

```
## S3 method for class 'summary.zibellreg'
print(x, ...)
```

Arguments

`x`	an object of the class summary.zibellreg.
`...`	further arguments passed to or from other methods.

Value

a summary of the fitted model.

`summary.bellreg` *Summary for the bellreg model*

Description

Summary for the bellreg model

Usage

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

Arguments

`object`	an objecto of the class 'bellreg'.
`...`	further arguments passed to or from other methods.

summary.zibellreg *Summary for the zibellreg model*

Description

Summary for the zibellreg model

Usage

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

Arguments

object	an objecto of the class 'zibellreg'.
...	further arguments passed to or from other methods.

vcov.bellreg *Variance-covariance matrix for a bellreg model*

Description

This function extracts and returns the variance-covariance matrix associated with the regression coefficients when the maximum likelihood estimation approach is used in the model fitting.

Usage

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

Arguments

object	an object of the class bellreg.
...	further arguments passed to or from other methods.

Value

the variance-covariance matrix associated with the regression coefficients.

Examples

```
data(faults)
fit <- bellreg(nf ~ lroll, data = faults)
vcov(fit)
```

vcov.zibellreg *Covariance of the regression coefficients*

Description

Covariance of the regression coefficients

Usage

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

Arguments

object	an object of the class bellreg
...	further arguments passed to or from other methods.

Value

the variance-covariance matrix associated with the regression coefficients.

Examples

```
data(cells)
fit <- zibellreg(cells ~ smoker + gender|smoker + gender, data = cells)
vcov(fit)
```

zibellreg *ZiBell regression model*

Description

Fits the Bell regression model to overdispersed count data.

Usage

```
zibellreg(
  formula,
  data,
  approach = c("mle", "bayes"),
  hessian = TRUE,
  link1 = c("logit", "probit", "cloglog", "cauchy"),
  link2 = c("log", "sqrt", "identity"),
  hyperpars = list(mu_psi = 0, sigma_psi = 10, mu_beta = 0, sigma_beta = 10),
  ...
)
```

Arguments

formula	an object of class "formula" (or one that can be coerced to that class): a symbolic description of the model to be fitted.
data	an optional data frame, list or environment (or object coercible by as.data.frame to a data frame) containing the variables in the model. If not found in data, the variables are taken from environment(formula), typically the environment from which ypbp is called.
approach	approach to be used to fit the model (mle: maximum likelihood; bayes: Bayesian approach).
hessian	hessian logical; If TRUE (default), the hessian matrix is returned when approach="mle".
link1	assumed link function for degenerate distribution (logit, probit, cloglog, cauchy); default is logit.
link2	assumed link function for count distribution (log, sqrt or identiy); default is log.
hyperpars	a list containing the hyperparameters associated with the prior distribution of the regression coefficients; if not specified then default choice is hyperpars = c(mu_psi = 0, sigma_psi = 10, mu_beta = 0, sigma_beta = 10).
...	further arguments passed to either `rstan::optimizing` or `rstan::sampling`.

Value

zibellreg returns an object of class "zibellreg" containing the fitted model.

Examples

```
# ML approach:
data(cells)
mle <- zibellreg(cells ~ smoker+gender|smoker+gender, data = cells, approach = "mle")
summary(mle)

# Bayesian approach:
bayes <- zibellreg(cells ~ 1|smoker+gender, data = cells, approach = "bayes", refresh = FALSE)
summary(bayes)
```

Index