

# Package ‘brglm2’

February 14, 2019

**Title** Bias Reduction in Generalized Linear Models

**Version** 0.5.1

**Description** Estimation and inference from generalized linear models based on various methods for bias reduction. The 'brglmFit' fitting method can achieve reduction of estimation bias by solving either the mean bias-reducing adjusted score equations in Firth (1993) <doi:10.1093/biomet/80.1.27> and Kosmidis and Firth (2009) <doi:10.1093/biomet/asp055>, or the median bias-reduction adjusted score equations in Kenne et al. (2016) <arXiv:1604.04768>, or through the direct subtraction of an estimate of the bias of the maximum likelihood estimator from the maximum likelihood estimates as in Cordeiro and McCullagh (1991) <http://www.jstor.org/stable/2345592>. Estimation in all cases takes place via a quasi Fisher scoring algorithm, and S3 methods for the construction of confidence intervals for the reduced-bias estimates are provided. In the special case of generalized linear models for binomial and multinomial responses (both ordinal and nominal), the adjusted score approaches return estimates with improved frequentist properties, that are also always finite, even in cases where the maximum likelihood estimates are infinite (e.g. complete and quasi-complete separation). 'brglm2' also provides pre-fit and post-fit methods for detecting separation and infinite maximum likelihood estimates in binomial response generalized linear models.

**URL** <https://github.com/ikosmidis/brglm2>

**BugReports** <https://github.com/ikosmidis/brglm2/issues>

**Depends** R (>= 3.3.0)

**Imports** MASS, stats, Matrix, graphics, nnet, enrichwith, lpSolveAPI,  
numDeriv

**License** GPL-3

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 6.1.1

**Suggests** testthat, knitr, rmarkdown, covr

**VignetteBuilder** knitr

**NeedsCompilation** yes

**Author** Ioannis Kosmidis [aut, cre] (<<https://orcid.org/0000-0003-1556-0302>>),  
 Kjell Konis [ctb],  
 Euloge Clovis Kenne Pagui [ctb],  
 Nicola Sartori [ctb]

**Maintainer** Ioannis Kosmidis <[ioannis.kosmidis@warwick.ac.uk](mailto:ioannis.kosmidis@warwick.ac.uk)>

**Repository** CRAN

**Date/Publication** 2019-02-14 17:02:15 UTC

## R topics documented:

alligators	2
bracl	3
brglm2	5
brglmControl	6
brglmFit	8
brmultinom	12
check_infinite_estimates	14
check_infinite_estimates.glm	15
coalition	16
confint.brglmFit	17
detect_separation	17
detect_separation_control	19
endometrial	20
lizards	21
mis	22
predict.bracl	23
predict.brmultinom	24
residuals.brmultinom	25
stemcell	26
summary.brglmFit	27
vcov.brglmFit	28
<b>Index</b>	<b>29</b>

---

alligators

*Alligator food choice data*

---

### Description

Alligator food choice data

### Usage

alligators

**Format**

A data frame with 80 rows and 5 variables:

**foodchoice** primary food type, in volume, found in an alligator's stomach, with levels fish, invertebrate, reptile, bird, other

**lake** lake of capture with levels Hancock, Oklawaha, Trafford, George

**gender** gender of the alligator with levels Male and Female

**size** size of the alligator with levels  $\leq 2.3$  meters long and  $> 2.3$  meters long

**freq** number of alligators for each foodchoice, lake, gender and size combination

**Source**

The alligators data set is analysed in Agresti (2002, Subsection~7.1.2).

**References**

Agresti, A. (2002). *\*Categorical Data Analysis\**. Wiley Series in Probability and Statistics. Wiley

---

bracl	<i>Bias reduction for adjacent category logit models for ordinal responses using the Poisson trick.</i>
-------	---

---

**Description**

bracl is a wrapper of [brglmFit](#) that fits adjacent category logit models with or without proportional odds using implicit and explicit bias reduction methods. See Kosmidis & Firth (2011) for details.

**Usage**

```
bracl(formula, data, weights, subset, na.action, parallel = FALSE,
      contrasts = NULL, model = TRUE, x = TRUE, control = list(...),
      ...)
```

**Arguments**

formula	a formula expression as for regression models, of the form response ~ predictors. The response should be a factor (preferably an ordered factor), which will be interpreted as an ordinal response, with levels ordered as in the factor. The model must have an intercept: attempts to remove one will lead to a warning and be ignored. An offset may be used. See the documentation of <a href="#">formula</a> for other details.
data	an optional data frame in which to interpret the variables occurring in formula.
weights	optional case weights in fitting. Default to 1.
subset	expression saying which subset of the rows of the data should be used in the fit. All observations are included by default.

na.action	a function to filter missing data.
parallel	if FALSE (default), then a non-proportional odds adjacent category model is fit, assuming different effects per category; if TRUE then a proportional odds adjacent category model is fit. See Details.
contrasts	a list of contrasts to be used for some or all of the factors appearing as variables in the model formula.
model	logical for whether the model matrix should be returned.
x	should the model matrix be included with in the result (default is TRUE).
control	a list of parameters for controlling the fitting process. See <a href="#">brglmControl</a> for details.
...	arguments to be used to form the default 'control' argument if it is not supplied directly.

### Details

The `brac1` function fits adjacent category models, which assume multinomial observations with probabilities with proportional odds of the form

$$\log \frac{\pi_{ij}}{\pi_{ij+1}} = \alpha_j + \beta^T x_i$$

or with non-proportional odds of the form

$$\log \frac{\pi_{ij}}{\pi_{ij+1}} = \alpha_j + \beta_j^T x_i$$

where  $x_i$  is a vector of covariates and  $\pi_{ij}$  is the probability that category  $j$  is observed at the covariate setting  $i$ .

### Author(s)

Ioannis Kosmidis <ioannis.kosmidis@warwick.ac.uk>

### References

- Kosmidis I, Kenne Pagui EC, Sartori N (2019). Mean and median bias reduction in generalized linear models. *\*arXiv e-prints\**, arXiv:1804.04085. To appear in *Statistics and Computing*, <URL: <https://arxiv.org/abs/1804.04085>>.
- Agresti, A. (2010). *\*Analysis of Ordinal Categorical Data\** (2nd edition). Wiley Series in Probability and Statistics. Wiley.
- Albert A. and Anderson J. A. (1984). On the Existence of Maximum Likelihood Estimates in Logistic Regression Models. *\*Biometrika\**, \*\*71\*\* 1–10.
- Kosmidis I. and Firth D. (2011). Multinomial logit bias reduction via the Poisson log-linear model. *\*Biometrika\**, \*\*98\*\* , 755-759.
- Palmgren, J. (1981). The Fisher Information Matrix for Log Linear Models Arguing Conditionally on Observed Explanatory Variables. *\*Biometrika\**, \*\*68\*\* , 563-566.

**See Also**[multinom](#), [brmultinom](#)**Examples**

```
data("stemcell", package = "brglm2")

# Adjacent category logit (non-proportional odds)
fit_bracl <- bracl(research ~ as.numeric(religion) + gender, weights = frequency,
  data = stemcell, type = "ML")
# Adjacent category logit (proportional odds)
fit_bracl_p <- bracl(research ~ as.numeric(religion) + gender, weights = frequency,
  data = stemcell, type = "ML", parallel = TRUE)
```

---

  
brglm2*brglm2: Bias Reduction in Generalized Linear Models*

---

**Description**

Estimation and inference from generalized linear models based on implicit methods for bias reduction (see Kosmidis, 2014). `brglm2` can achieve reduction of estimation bias by solving either the mean-bias reducing adjusted score equations in Firth (1993) and Kosmidis and Firth (2009), or the median-bias reducing adjusted score equations in Kenne et al. (2016), or through the direct subtraction of an estimate of the bias of the maximum likelihood estimator from the maximum likelihood estimates as in Cordeiro and McCullagh (1991),

**Details**

In the special case of generalized linear models for binomial, Poisson and multinomial responses, the adjusted score equations approaches return estimates with improved frequentist properties, that are also always finite, even in cases where the maximum likelihood estimates are infinite (e.g. complete and quasi-complete separation in multinomial regression; see also [detect\\_separation](#) and [check\\_infinite\\_estimates](#) for pre-fit and post-fit methods for the detection of infinite estimates in binomial response generalized linear models). Estimation in all cases takes place via a modified Fisher scoring algorithm, and S3 methods for the construction of confidence intervals for the reduced-bias estimates are provided.

The similarly named `brglm` R package can only handle generalized linear models with binomial responses. Special care has been taken when developing `brglm2` in order not to have conflicts when the user loads `brglm2` and `brglm` simultaneously. The development and maintenance of the two packages will continue, until `brglm2` incorporates all `brglm` functionality and gets an appropriate wrapper to the `brglm::brglm` function.

**Author(s)**

Ioannis Kosmidis <ioannis.kosmidis@warwick.ac.uk>

## References

- Cordeiro G. M. & McCullagh, P. (1991). Bias correction in generalized linear models. *Journal of the Royal Statistical Society. Series B (Methodological)*, **53**, 629-643
- Firth D. (1993). Bias reduction of maximum likelihood estimates, *Biometrika*, **80**, 27-38
- Kenne Pagui E C, Salvan A and Sartori N (2016). Median bias reduction of maximum likelihood estimates. *arXiv*, **arXiv:1604.04768**
- Kosmidis I and Firth D (2009). Bias reduction in exponential family nonlinear models. *Biometrika*, **96**, 793-804
- Kosmidis I and Firth D (2010). A generic algorithm for reducing bias in parametric estimation. *Electronic Journal of Statistics*, **4**, 1097-1112
- Kosmidis I (2014). Bias in parametric estimation: reduction and useful side-effects. *WIRE Computational Statistics*, **6**, 185-196

---

brglmControl

*Auxiliary function for glm fitting using the brglmFit method.*


---

## Description

Typically only used internally by `brglmFit`, but may be used to construct a control argument.

## Usage

```
brglmControl(epsilon = 1e-08, maxit = 100, trace = FALSE,
             type = c("AS_mixed", "AS_mean", "AS_median", "correction", "ML"),
             transformation = "identity", slowit = 1, max_step_factor = 12)
```

```
brglm_control(epsilon = 1e-08, maxit = 100, trace = FALSE,
              type = c("AS_mixed", "AS_mean", "AS_median", "correction", "ML"),
              transformation = "identity", slowit = 1, max_step_factor = 12)
```

## Arguments

- |                              |   |
|------------------------------|---|
| <code>epsilon</code>         | positive convergence tolerance epsilon.   |
| <code>maxit</code>           | integer giving the maximal number of iterations allowed.  |
| <code>trace</code>           | logical indicating if output should be produced for each iteration.   |
| <code>type</code>            | the type of fitting method to be used. The options are <code>AS_mean</code> (mean-bias reducing adjusted scores), <code>AS_median</code> (median-bias reducing adjusted scores), <code>AS_mixed</code> (bias reduction using mixed score adjustents; default), <code>correction</code> (asymptotic bias correction) and <code>ML</code> (maximum likelihood). |
| <code>transformation</code>  | the transformation of the dispersion to be estimated. Default is <code>identity</code> . See Details.   |
| <code>slowit</code>          | a positive real used as a multiplier for the stepsize. The smaller it is the smaller the steps are.   |
| <code>max_step_factor</code> | the maximum number of step halving steps to consider.   |

## Details

`brglmControl` provides default values and sanity checking for the various constants that control the iteration and generally the behaviour of `brglmFit`.

When `trace` is true, calls to `cat` produce the output for each iteration. Hence, `options(digits = *)` can be used to increase the precision.

`transformation` sets the transformation of the dispersion parameter for which the bias reduced estimates are computed. Can be one of "identity", "sqrt", "inverse", "log" and "inverseSqrt". Custom transformations are accommodated by supplying a list of two expressions (transformation and inverse transformation). See the examples for more details.

`brglm_control` is an alias to `brglmControl`.

## Value

a list with components named as the arguments, including symbolic expressions for the dispersion transformation (`Trans`) and its inverse (`inverseTrans`)

## Author(s)

Ioannis Kosmidis <ioannis.kosmidis@warwick.ac.uk>

## See Also

`brglmFit` and `glm.fit`

## Examples

```
data("coalition", package = "brglm2")
## The maximum likelihood fit with log link
coalitionML <- glm(duration ~ fract + numst2, family = Gamma, data = coalition)

## Bias reduced estimation of the dispersion parameter
coalitionBRi <- glm(duration ~ fract + numst2, family = Gamma, data = coalition,
                    method = "brglmFit")
coef(coalitionBRi, model = "dispersion")

## Bias reduced estimation of log(dispersion)
coalitionBRl <- glm(duration ~ fract + numst2, family = Gamma, data = coalition,
                    method = "brglmFit", transformation = "log")
coef(coalitionBRl, model = "dispersion")

## Just for illustration: Bias reduced estimation of dispersion^0.25
my_transformation <- list(expression(dispersion^0.25), expression(transformed_dispersion^4))
coalitionBRc <- update(coalitionBRi, transformation = my_transformation)
coef(coalitionBRc, model = "dispersion")
```

## Description

`brglmFit` is a fitting function for `glm` that fits generalized linear models using implicit and explicit bias reduction methods (Kosmidis, 2014). Currently supported methods include the mean bias-reducing adjusted scores approach in Firth (1993) and Kosmidis & Firth (2009), the median bias-reduction adjusted scores approach in Kenne et al. (2016), the correction of the asymptotic bias in Cordeiro & McCullagh (1991), and maximum likelihood. Estimation is performed using a quasi Fisher scoring iteration, which, in the case of mean-bias reduction, resembles an iterative correction of the asymptotic bias of the Fisher scoring iterates.

## Usage

```
brglmFit(x, y, weights = rep(1, nobs), start = NULL, etastart = NULL,
         mustart = NULL, offset = rep(0, nobs), family = gaussian(),
         control = list(), intercept = TRUE, fixed_totals = NULL,
         singular.ok = TRUE)
```

```
brglm_fit(x, y, weights = rep(1, nobs), start = NULL,
          etastart = NULL, mustart = NULL, offset = rep(0, nobs),
          family = gaussian(), control = list(), intercept = TRUE,
          fixed_totals = NULL, singular.ok = TRUE)
```

## Arguments

<code>x</code>	<code>x</code> is a design matrix of dimension $n * p$ .
<code>y</code>	<code>y</code> is a vector of observations of length $n$ .
<code>weights</code>	an optional vector of ‘prior weights’ to be used in the fitting process. Should be <code>NULL</code> or a numeric vector.
<code>start</code>	starting values for the parameters in the linear predictor. If <code>NULL</code> (default) then the maximum likelihood estimates are calculated and used as starting values.
<code>etastart</code>	applied only when <code>start</code> is not <code>NULL</code> . Starting values for the linear predictor to be passed to <code>glm.fit</code> when computing starting values using maximum likelihood.
<code>mustart</code>	applied only when <code>start</code> is not <code>NULL</code> . Starting values for the vector of means to be passed to <code>glm.fit</code> when computing starting values using maximum likelihood.
<code>offset</code>	this can be used to specify an <i>a priori</i> known component to be included in the linear predictor during fitting. This should be <code>NULL</code> or a numeric vector of length equal to the number of cases. One or more <code>offset</code> terms can be included in the formula instead or as well, and if more than one is specified their sum is used. See <code>model.offset</code> .
<code>family</code>	a description of the error distribution and link function to be used in the model. For <code>glm</code> this can be a character string naming a family function, a family function or the result of a call to a family function. For <code>glm.fit</code> only the third option is supported. (See <code>family</code> for details of family functions.)



control	a list of parameters controlling the fitting process. See <a href="#">brglmControl</a> for details.
intercept	logical. Should an intercept be included in the <i>null</i> model?
fixed_totals	effective only when family is poisson. Either NULL (no effect) or a vector that indicates which counts must be treated as a group. See Details for more information and <a href="#">brmultinom</a> .
singular.ok	logical. If FALSE, a singular model is an error.
...	arguments to be used to form the default 'control' argument if it is not supplied directly.

## Details

A detailed description of the supported adjustments and the quasi Fisher scoring iteration is given in the iteration vignette (see, Kosmidis et al, 2018). A shorter description of the quasi Fisher scoring iteration is also given in one of the vignettes of the *\*enrichwith\** R package (see, <https://cran.r-project.org/package=enrichwith/vignettes/bias.html>). Kosmidis and Firth (2010) describe a parallel quasi Newton-Raphson iteration with the same stationary point.

In the special case of generalized linear models for binomial, Poisson and multinomial responses, the adjusted score equations approach returns estimates with improved frequentist properties, that are also always finite, even in cases where the maximum likelihood estimates are infinite (e.g. complete and quasi-complete separation in multinomial regression; see also [detect\\_separation](#) and [check\\_infinite\\_estimates](#) for pre-fit and post-fit methods for the detection of infinite estimates in binomial response generalized linear models).

The type of the bias-reducing adjustment to be used is specified through the type argument (see [brglmControl](#) for details). The default is to use the mean bias-reducing adjustments in Firth (1993) and Kosmidis & Firth (2009) (type = "AS\_mean").

The null deviance is evaluated based on the fitted values using the method specified by the type argument (see [brglmControl](#)).

The family argument of the current version of `brglmFit` can accept any combination of [family](#) objects and link functions (including ones with user-specified link functions, [mis](#) links, and [power](#) links), except the [quasi](#), [quasipoisson](#) and [quasibinomial](#) families.

The description of method argument and the Fitting functions section in [glm](#) gives information on supplying fitting methods to [glm](#).

`fixed_totals` can be used to constrain the means of a poisson model to add up to the corresponding observed counts according to

`brglm_fit` is an alias to `brglmFit`.

## Author(s)

Ioannis Kosmidis [aut, cre] <ioannis.kosmidis@warwick.ac.uk>, Euloge Clovis Kenne Pagui [ctb] <kenne@stat.unipd.it>

## References

- Kosmidis I, Kenne Pagui EC, Sartori N (2019). Mean and median bias reduction in generalized linear models. *\*arXiv e-prints\**, arXiv:1804.04085. To appear in *Statistics and Computing*, <URL: <https://arxiv.org/abs/1804.04085>>.
- Cordeiro G. M. & McCullagh, P. (1991). Bias correction in generalized linear models. *\*Journal of the Royal Statistical Society. Series B (Methodological)\**, **53**, 629-643
- Firth D. (1993). Bias reduction of maximum likelihood estimates, *Biometrika*, **80**, 27-38
- Kenne Pagui, E. C., A. Salvan, and N. Sartori (2017). Median bias reduction of maximum likelihood estimates. *\*Biometrika\**, **104**, 923–938
- Kosmidis I and Firth D (2009). Bias reduction in exponential family nonlinear models. *\*Biometrika\**, **96**, 793-804
- Kosmidis I and Firth D (2010). A generic algorithm for reducing bias in parametric estimation. *\*Electronic Journal of Statistics\**, **4**, 1097-1112
- Kosmidis I (2014). Bias in parametric estimation: reduction and useful side-effects. *\*WIRE Computational Statistics\**, **6**, 185-196

## See Also

[glm.fit](#) and [glm](#)

## Examples

```
## The lizards example from ?brglm::brglm
data("lizards")
# Fit the model using maximum likelihood
lizardsML <- glm(cbind(grahami, opalinus) ~ height + diameter +
                light + time, family = binomial(logit), data = lizards,
                method = "glm.fit")
# Mean bias-reduced fit:
lizardsBR_mean <- glm(cbind(grahami, opalinus) ~ height + diameter +
                    light + time, family = binomial(logit), data = lizards,
                    method = "brglmFit")
# Median bias-reduced fit:
lizardsBR_median <- glm(cbind(grahami, opalinus) ~ height + diameter +
                      light + time, family = binomial(logit), data = lizards,
                      method = "brglmFit", type = "AS_median")
summary(lizardsML)
summary(lizardsBR_median)
summary(lizardsBR_mean)

## Another example from
## King, Gary, James E. Alt, Nancy Elizabeth Burns and Michael Laver
## (1990). "A Unified Model of Cabinet Dissolution in Parliamentary
## Democracies", _American Journal of Political Science_, 34, 846-870

## Not run:
data("coalition", package = "brglm2")
# The maximum likelihood fit with log link
```

```

coalitionML <- glm(duration ~ fract + numst2, family = Gamma, data = coalition)
# The mean bias-reduced fit
coalitionBR_mean <- update(coalitionML, method = "brglmFit")
# The bias-corrected fit
coalitionBC <- update(coalitionML, method = "brglmFit", type = "correction")
# The median bias-corrected fit
coalitionBR_median <- update(coalitionML, method = "brglmFit", type = "AS_median")

## End(Not run)

## Not run:
## An example with offsets from Venables & Ripley (2002, p.189)
data("anorexia", package = "MASS")

anorexML <- glm(Postwt ~ Prewt + Treat + offset(Prewt),
  family = gaussian, data = anorexia)
anorexBC <- update(anorexML, method = "brglmFit", type = "correction")
anorexBR_mean <- update(anorexML, method = "brglmFit")
anorexBR_median <- update(anorexML, method = "brglmFit", type = "AS_median")

## All methods return the same estimates for the regression
## parameters because the maximum likelihood estimator is normally
## distributed around the `true` value under the model (hence, both
## mean and component-wise median unbiased). The Wald tests for
## anorexBC and anorexBR_mean differ from anorexML
## because the bias-reduced estimator of the dispersion is the
## unbiased, by degree of freedom adjustment (divide by n - p),
## estimator of the residual variance. The Wald tests from
## anorexBR_median are based on the median bias-reduced estimator
## of the dispersion that results from a different adjustment of the
## degrees of freedom (divide by n - p - 2/3)
summary(anorexML)
summary(anorexBC)
summary(anorexBR_mean)
summary(anorexBR_median)

## End(Not run)

## endometrial data from Heinze & Schemper (2002) (see ?endometrial)
data("endometrial", package = "brglm2")
endometrialML <- glm(HG ~ NV + PI + EH, data = endometrial,
  family = binomial("probit"))
endometrialBR_mean <- update(endometrialML, method = "brglmFit",
  type = "AS_mean")
endometrialBC <- update(endometrialML, method = "brglmFit",
  type = "correction")
endometrialBR_median <- update(endometrialML, method = "brglmFit",
  type = "AS_median")

summary(endometrialML)
summary(endometrialBC)
summary(endometrialBR_mean)
summary(endometrialBR_median)

```

---

brmultinom	<i>Bias reduction for multinomial response models using the Poisson trick.</i>
------------	--

---

## Description

brmultinom is a wrapper of `brglmFit` that fits multinomial regression models using implicit and explicit bias reduction methods. See Kosmidis & Firth (2011) for details.

## Usage

```
brmultinom(formula, data, weights, subset, na.action, contrasts = NULL,
           ref = 1, model = TRUE, x = TRUE, control = list(...), ...)
```

## Arguments

formula	a formula expression as for regression models, of the form <code>response ~ predictors</code> . The response should be a factor or a matrix with <code>K</code> columns, which will be interpreted as counts for each of <code>K</code> classes. A log-linear model is fitted, with coefficients zero for the first class. An offset can be included: it should be a numeric matrix with <code>K</code> columns if the response is either a matrix with <code>K</code> columns or a factor with <code>K &gt;= 2</code> classes, or a numeric vector for a response factor with 2 levels. See the documentation of <code>formula()</code> for other details.
data	an optional data frame in which to interpret the variables occurring in formula.
weights	optional case weights in fitting.
subset	expression saying which subset of the rows of the data should be used in the fit. All observations are included by default.
na.action	a function to filter missing data.
contrasts	a list of contrasts to be used for some or all of the factors appearing as variables in the model formula.
ref	the reference category to use for multinomial regression. Either an integer, in which case <code>levels(response)[ref]</code> is used as a baseline, or a character string. Default is 1.
model	logical. If true, the model frame is saved as component <code>model</code> of the returned object.
x	should the model matrix be included with in the result (default is TRUE).
control	a list of parameters for controlling the fitting process. See <code>brglmControl</code> for details.
...	arguments to be used to form the default 'control' argument if it is not supplied directly.

## Details

The models `brmultinom` handles are also known as baseline-category logit models (see, Agresti, 2002, Section 7.1), because they model the log-odds of every category against a baseline category. The user can control which baseline (or reference) category is used via the `ref`. By default `brmultinom` uses the first category as reference.

The maximum likelihood estimates for the parameters of baseline-category logit models have infinite components with positive probability, which can result in problems in their estimation and the use of inferential procedures (e.g. Wald tests). Albert and Anderson (1984) have categorised the possible data patterns for such models into the exclusive and exhaustive categories of complete separation, quasi-complete separation and overlap, and showed that infinite maximum likelihood estimates result when complete or quasi-complete separation occurs.

The adjusted score approach to bias reduction that `brmultinom` implements (`type = "AS_mean"`) is an alternative to maximum likelihood that results in estimates with smaller asymptotic bias that are also *\*always\** finite, even in cases of complete or quasi-complete separation.

`brmultinom` is a wrapper of `brglmFit` that fits multinomial logit regression models through the 'Poisson trick' (see, for example, Palmgren, 1981; Kosmidis & Firth, 2011).

The implementation relies on the construction of an 'extended' model matrix for the log-linear model and constraints on the sums of the Poisson means. Specifically, a log-linear model is fitted on a Kronecker product ([https://en.wikipedia.org/wiki/Kronecker\\_product](https://en.wikipedia.org/wiki/Kronecker_product)) of the original model matrix  $X$  implied by the formula, augmented by `nrow(X)` dummy variables.

The extended model matrix is sparse, and the **Matrix** package is used for its effective storage.

While `brmultinom` can be used for analyses using multinomial regression models, the current implementation is more of a 'proof of concept' and is not expected to scale well with either of `nrow(X)`, `ncol(X)` or the number of levels in the categorical response.

## Author(s)

Ioannis Kosmidis <ioannis.kosmidis@warwick.ac.uk>

## References

- Kosmidis I, Kenne Pagui EC, Sartori N (2019). Mean and median bias reduction in generalized linear models. *\*arXiv e-prints\**, arXiv:1804.04085. To appear in *Statistics and Computing*, <URL: <https://arxiv.org/abs/1804.04085>>.
- Agresti A. (2002). *\*Categorical data analysis\** (2nd edition). Wiley Series in Probability and Statistics. Wiley.
- Albert A. and Anderson J. A. (1984). On the Existence of Maximum Likelihood Estimates in Logistic Regression Models. *\*Biometrika\**, **71**, 1–10.
- Kosmidis I. and Firth D. (2011). Multinomial logit bias reduction via the Poisson log-linear model. *\*Biometrika\**, **98**, 755-759.
- Palmgren, J. (1981). The Fisher Information Matrix for Log Linear Models Arguing Conditionally on Observed Explanatory Variables. *\*Biometrika\**, **68**, 563-566.

## See Also

`multinom`

## Examples

```
data("housing", package = "MASS")

# Maximum likelihood using nnet::multinom
houseML1nnet <- nnet::multinom(Sat ~ Infl + Type + Cont, weights = Freq,
                             data = housing)
# Maximum likelihood using brmultinom with baseline category 'Low'
houseML1 <- brmultinom(Sat ~ Infl + Type + Cont, weights = Freq,
                      data = housing, type = "ML", ref = 1)
# The estimates are numerically the same as houseML0
all.equal(coef(houseML1nnet), coef(houseML1), tolerance = 1e-04)

# Maximum likelihood using brmultinom with 'High' as baseline
houseML3 <- brmultinom(Sat ~ Infl + Type + Cont, weights = Freq,
                      data = housing, type = "ML", ref = 3)
# The fitted values are the same as houseML1
all.equal(fitted(houseML3), fitted(houseML1), tolerance = 1e-10)

# Bias reduction
houseBR3 <- update(houseML3, type = "AS_mean")
# Bias correction
houseBC3 <- update(houseML3, type = "correction")
```

---

check\_infinite\_estimates

*Generic method for checking for infinite estimates*

---

## Description

Generic method for checking for infinite estimates

## Usage

```
check_infinite_estimates(object, ...)
```

## Arguments

**object** a fitted model object (e.g. the result of a `glm` call).  
**...** other options to be passed to the method.

---

```
check_infinite_estimates.glm
```

*A simple diagnostic of whether the maximum likelihood estimates are infinite*

---

## Description

A simple diagnostic of whether the maximum likelihood estimates are infinite

## Usage

```
## S3 method for class 'glm'
check_infinite_estimates(object, nsteps = 20, ...)
```

## Arguments

object	the result of a <code>glm</code> call.
nsteps	starting from <code>maxit = 1</code> , the GLM is refitted for <code>maxit = 2</code> , <code>maxit = 3</code> , ..., <code>maxit = nsteps</code> . Default value is 30.
...	currently not used.

## Details

`check_infinite_estimates` attempts to identify the occurrence of infinite estimates in GLMs with binomial responses by successively refitting the model. At each iteration the maximum number of allowed IWLS iterations is fixed starting from 1 to `nsteps` (by setting `control = glm.control(maxit = j)`, where `j` takes values 1, ..., `nsteps` in `glm`). For each value of `maxit`, the estimated asymptotic standard errors are divided to the corresponding ones from `control = glm.control(maxit = 1)`. Then, based on the results in Lesaffre & Albert (1989), if the sequence of ratios in any column of the resultant matrix diverges, then complete or quasi-complete separation occurs and the maximum likelihood estimate for the corresponding parameter has value minus or plus infinity.

## References

Lesaffre, E., & Albert, A. (1989). Partial Separation in Logistic Discrimination. *Journal of the Royal Statistical Society. Series B (Methodological)*, **51**, 109-116

## See Also

[multinom](#)

## Examples

```
## endometrial data from Heinze & Schemper (2002) (see ?endometrial)
data("endometrial", package = "brglm2")
endometrialML <- glm(HG ~ NV + PI + EH, data = endometrial,
```

```

        family = binomial("probit"))
## clearly the maximum likelihood estimate for the coefficient of
## NV is infinite
check_infinite_estimates(endometrialML)

## Not run:
## Alligator data (Agresti, 2002, Table~7.1)
data("alligator", package = "brglm2")
all_ml <- brmultinom(foodchoice ~ size + lake , weights = round(freq/3),
                    data = alligators, type = "ML", ref = 1)
## Clearly some estimated standard errors diverge as the number of
## Fisher scoring iterations increases
matplot(check_infinite_estimates(all_ml), type = "l", lty = 1,
        ylim = c(0.5, 1.5))

## End(Not run)

```

---

 coalition

*Coalition data*


---

### Description

Coalition data

### Usage

```
coalition
```

### Format

An object of class `data.frame` with 314 rows and 7 columns.

### Note

Data is as provided in the Zeilig R package (<https://cran.r-project.org/package=Zelig>)

### References

King, G., Alt, J. E., Burns, N. E. and Laver, M. (1990). A Unified Model of Cabinet Dissolution in Parliamentary Democracies. *American Journal of Political Science*, \*\*34\*\*, 846-870.

King, G., Alt, J. E., Burns, N. E. and Laver, M. ICPSR Publication Related Archive, 1115.



---

confint.brglmFit	<i>Method for computing confidence intervals for one or more regression parameters in a <a href="#">brglmFit</a> object</i>
------------------	---

---

### Description

Method for computing confidence intervals for one or more regression parameters in a [brglmFit](#) object

### Usage

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

### Arguments

object	a fitted model object.
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.
...	additional argument(s) for methods.

---

detect_separation	<i>Method for <a href="#">glm</a> that tests for data separation and finds which parameters have infinite maximum likelihood estimates in generalized linear models with binomial responses</i>
-------------------	---

---

### Description

[detect\\_separation](#) is a method for [glm](#) that tests for the occurrence of complete or quasi-complete separation in datasets for binomial response generalized linear models, and finds which of the parameters will have infinite maximum likelihood estimates. [detect\\_separation](#) relies on the linear programming methods developed in Konis (2007).

### Usage

```
detect_separation(x, y, weights = rep(1, nobs), start = NULL,
  etastart = NULL, mustart = NULL, offset = rep(0, nobs),
  family = gaussian(), control = list(), intercept = TRUE,
  singular.ok = TRUE)
```

```
detectSeparation(x, y, weights = rep(1, nobs), start = NULL,
  etastart = NULL, mustart = NULL, offset = rep(0, nobs),
  family = gaussian(), control = list(), intercept = TRUE,
  singular.ok = TRUE)
```

**Arguments**

x	x is a design matrix of dimension $n * p$ .
y	y is a vector of observations of length n.
weights	an optional vector of 'prior weights' to be used in the fitting process. Should be NULL or a numeric vector.
start	currently not used.
etastart	currently not used.
mustart	currently not used.
offset	this can be used to specify an <i>a priori</i> known component to be included in the linear predictor during fitting. This should be NULL or a numeric vector of length equal to the number of cases. One or more <code>offset</code> terms can be included in the formula instead or as well, and if more than one is specified their sum is used. See <code>model.offset</code> .
family	a description of the error distribution and link function to be used in the model. For <code>glm</code> this can be a character string naming a family function, a family function or the result of a call to a family function. For <code>glm.fit</code> only the third option is supported. (See <code>family</code> for details of family functions.)
control	a list of parameters controlling separation detection. See <code>detect_separation_control</code> for details.
intercept	logical. Should an intercept be included in the <i>null</i> model?
singular.ok	logical. If FALSE, a singular model is an error.
...	arguments to be used to form the default 'control' argument if it is not supplied directly.

**Details**

For the definition of complete and quasi-complete separation, see Albert and Anderson (1984).

`detect_separation` is a wrapper to the `separator` function from the `**safeBinaryRegression**` R package, that can be passed directly as a method to the `glm` function. See, examples.

The interface to `separator` was designed by Ioannis Kosmidis after correspondence with Kjell Konis, and a port of `separator` has been included in `**brglm2**` under the permission of Kjell Konis.

`detectSeparation` is an alias for `detect_separation`.

**Author(s)**

Ioannis Kosmidis [aut, cre] <ioannis.kosmidis@warwick.ac.uk>, Kjell Konis [ctb] <kjell.konis@me.com>

**References**

Kjell K. (2007). \*Linear Programming Algorithms for Detecting Separated Data in Binary Logistic Regression Models\*. DPhil. University of Oxford. <https://ora.ox.ac.uk/objects/uuid:8f9ee0d0-d78e-4101-9ab4-f9cbceed2a2a>

Kjell K. (2013). `safeBinaryRegression`: Safe Binary Regression. R package version 0.1-3. <https://CRAN.R-project.org/package=safeBinaryRegression>

**See Also**

[brglmFit](#), [glm.fit](#) and [glm](#)

**Examples**

```
## endometrial data from Heinze & Schemper (2002) (see ?endometrial)
data("endometrial", package = "brglm2")
endometrial_sep <- glm(HG ~ NV + PI + EH, data = endometrial,
  family = binomial("logit"),
  method = "detect_separation")

endometrial_sep
## The maximum likelihood estimate for NV is infinite
summary(update(endometrial_sep, method = "glm.fit"))

## Not run:
## Example inspired by unpublished microeconometrics lecture notes by
## Achim Zeileis https://eeecon.uibk.ac.at/~zeileis/
## The maximum likelihood estimate of southernness is infinite
data("MurderRates", package = "AER")
murder_sep <- glm(I(executions > 0) ~ time + income +
  noncauc + lfp + southern, data = MurderRates,
  family = binomial(), method = "detect_separation")

murder_sep
## which is also evident by the large estimated standard error for NV
murder_glm <- update(murder_sep, method = "glm.fit")
summary(murder_glm)
## and is also reveal by the divergence of the NV column of the
## result from the more computationally intensive check
check_infinite_estimates(murder_glm)
## Mean bias reduction via adjusted scores results in finite estimates
update(murder_glm, method = "brglm_fit")

## End(Not run)
```

---

detect\_separation\_control

*Auxiliary function for the [glm](#) interface when method is [detect\\_separation](#).*

---

**Description**

Typically only used internally by [detect\\_separation](#) but may be used to construct a control argument.

**Usage**

```
detect_separation_control(linear_program = c("primal", "dual"),
  purpose = c("find", "test"),
```

```

beta_tolerance = sqrt(.Machine$double.eps))

detectSeparationControl(linear_program = c("primal", "dual"),
  purpose = c("find", "test"),
  beta_tolerance = sqrt(.Machine$double.eps))

```

### Arguments

`linear_program` should `detect_separation` solve the "primal" or "dual" linear program for separation detection?

`purpose` should `detect_separation` simply "test" for separation or also "find" which parameters are infinite?

`beta_tolerance` maximum absolute variable value from the linear program, before separation is declared.

---

endometrial

*Histology grade and risk factors for 79 cases of endometrial cancer*

---

### Description

Histology grade and risk factors for 79 cases of endometrial cancer

### Usage

```
endometrial
```

### Format

A data frame with 79 rows and 4 variables:

**NV** neovascularization with coding 0 for absent and 1 for present

**PI** pulsality index of arteria uterina

**EH** endometrium heigh

**HG** histology grade with coding 0 for low grade and 1 for high grade

### Source

The packaged data set was downloaded in .dat format from <http://www.stat.ufl.edu/~aa/glm/data>. The latter link provides the data sets used in Agresti (2015).

The endometrial data set was first analysed in Heinze and Schemper (2002), and was originally provided by Dr E. Asseryanis from the Medical University of Vienna.

### References

Agresti, A. (2015). *\*Foundations of Linear and Generalized Linear Models\**. Wiley Series in Probability and Statistics. Wiley

Heinze, G., & Schemper, M. (2002). A Solution to the Problem of Separation in Logistic Regression. *\*Statistics in Medicine\**, *\*\*21\*\**, 2409–2419

---

lizards	<i>Habitat preferences of lizards</i>
---------	---------------------------------------

---

### Description

The lizards data frame has 23 rows and 6 columns. Variables `grahami` and `opalinus` are counts of two lizard species at two different perch heights, two different perch diameters, in sun and in shade, at three times of day.

### Usage

```
lizards
```

### Format

An object of class `data.frame` with 23 rows and 6 columns.

### Details

- `grahami`. count of grahami lizards
- `opalinus`. count of opalinus lizards
- `height`. a factor with levels `<5ft`, `>=5ft`
- `diameter`. a factor with levels `<=2in`, `>2in`
- `light`. a factor with levels `sunny`, `shady`
- `time`. a factor with levels `early`, `midday`, `late`

### Source

McCullagh, P. and Nelder, J. A. (1989) *Generalized Linear Models* (2nd Edition). London: Chapman and Hall.

Originally from

Schoener, T. W. (1970) Nonsynchronous spatial overlap of lizards in patchy habitats. *Ecology* **51**, 408-418.

---

mis	A link-glm object for misclassified responses in binomial regression models
-----	---

---

### Description

`mis` is a link-glm object that specifies the link function in Neuhaus (1999, expression~(8)) for handling misclassified responses in binomial regression models using maximum likelihood. A prior specification of the sensitivity and specificity is required.

### Usage

```
mis(link = "logit", sensitivity = 1, specificity = 1)
```

### Arguments

link	the baseline link to be used.
sensitivity	the probability of observing a success given that a success actually took place given any covariate values.
specificity	the probability of observing a failure given that a failure actually took place given any covariate values.

### Details

sensitivity + specificity should be greater or equal to 1, otherwise it is implied that the procedure producing the responses performs worse than chance in terms of misclassification.

### References

Neuhaus J. M. (1999). Bias and efficiency loss due to misclassified responses in binary regression. *Biometrika*, **86**, 843-855

### Examples

```
## Define a few links with some misclassification
logit_mis <- mis(link = "logit", sensitivity = 0.9, specificity = 0.9)

lizards_f <- cbind(ghahami, opalinus) ~ height + diameter + light + time

lizardsML <- glm(lizards_f, family = binomial(logit), data = lizards)

lizardsML_mis <- update(lizardsML, family = binomial(logit_mis),
                       start = coef(lizardsML))

## A notable change in coefficients is noted here compared to when
## specificity and sensitivity are 1
coef(lizardsML)
```

```

coef(lizardsML_mis)

## Bias reduction is also possible
update(lizardsML_mis, method = "brglmFit", type = "AS_mean",
       start = coef(lizardsML))

update(lizardsML_mis, method = "brglmFit", type = "AS_median",
       start = coef(lizardsML))

```

---

predict.bracl	<i>Predict method for bracl fits</i>
---------------	--------------------------------------

---

## Description

Obtain class and probability predictions from a fitted adjacent category logits model.

## Usage

```

## S3 method for class 'bracl'
predict(object, newdata, type = c("class", "probs"), ...)

```

## Arguments

object	a fitted object of class inheriting from "bracl".
newdata	optionally, a data frame in which to look for variables with which to predict. If omitted, the fitted linear predictors are used.
type	the type of prediction required. The default is "class", which produces predictions of the response category at the covariate values supplied in "newdata", selecting the category with the largest probability; the alternative "probs" returns all category probabilities at the covariate values supplied in "newdata".
...	further arguments passed to or from other methods.

## Details

If newdata is omitted the predictions are based on the data used for the fit.

## Value

If type = "class" a vector with the predicted response categories; if type = "probs" a matrix of probabilities for all response categories at newdata.

## Examples

```
data("stemcell", package = "brglm2")

# Adjacent category logit (non-proportional odds)
fit_bracl <- bracl(research ~ as.numeric(religion) + gender, weights = frequency,
                  data = stemcell, type = "ML")
# Adjacent category logit (proportional odds)
fit_bracl_p <- bracl(research ~ as.numeric(religion) + gender, weights = frequency,
                   data = stemcell, type = "ML", parallel = TRUE)

# New data
newdata <- expand.grid(gender = c("male", "female"),
                     religion = c("liberal", "moderate", "fundamentalist"))

# Predictions
sapply(c("class", "probs"), function(what) predict(fit_bracl, newdata, what))
sapply(c("class", "probs"), function(what) predict(fit_bracl_p, newdata, what))
```

---

predict.brmultinom      *Predict method for brmultinom fits*

---

## Description

Obtain class and probability predictions from a fitted baseline category logits model.

## Usage

```
## S3 method for class 'brmultinom'
predict(object, newdata, type = c("class", "probs"),
       ...)
```

## Arguments

object	a fitted object of class inheriting from "brmultinom".
newdata	optionally, a data frame in which to look for variables with which to predict. If omitted, the fitted linear predictors are used.
type	the type of prediction required. The default is "class", which produces predictions of the response category at the covariate values supplied in "newdata", selecting the category with the largest probability; the alternative "probs" returns all category probabilities at the covariate values supplied in "newdata".
...	further arguments passed to or from other methods.

## Details

If newdata is omitted the predictions are based on the data used for the fit.



**Value**

If `type = "class"` a vector with the predicted response categories; if `type = "probs"` a matrix of probabilities for all response categories at `newdata`.

**Examples**

```
data("housing", package = "MASS")

# Maximum likelihood using brmultinom with baseline category 'Low'
houseML1 <- brmultinom(Sat ~ Infl + Type + Cont, weights = Freq,
                      data = housing, type = "ML", ref = 1)

# New data
newdata <- expand.grid(Infl = c("Low", "Medium"),
                      Type = c("Tower", "Atrium", "Terrace"),
                      Cont = c("Low", NA, "High"))

## Predictions
sapply(c("class", "probs"), function(what) predict(houseML1, newdata, what))
```

---

residuals.brmultinom *Residuals for multinomial logistic regression and adjacent category logit models*

---

**Description**

Residuals for multinomial logistic regression and adjacent category logit models

**Usage**

```
## S3 method for class 'brmultinom'
residuals(object, type = c("pearson", "response",
                          "deviance", "working"), ...)

## S3 method for class 'bracl'
residuals(object, type = c("pearson", "response",
                          "deviance", "working"), ...)
```

**Arguments**

<code>object</code>	the object coming out of <code>bracl</code> and <code>brmultinom</code> .
<code>type</code>	the type of residuals which should be returned. The options are: "pearson" (default), "response", "deviance", "working". See Details.
<code>...</code>	Currently not used.

**Details**

The residuals computed are the residuals from the equivalent Poisson log-linear model fit, organised in a form that matches the output of `fitted(object, type = "probs")`. As a result, the output is residuals defined in terms of the object and expected multinomial counts.

**See Also**

`brmultinom` `brac1`

---

stemcell

*Opinion on Stem Cell Research and Religious Fundamentalism*

---

**Description**

A data set from the 2006 General Social Survey that shows the relationship in the United States between opinion about funding stem cell research and the fundamentalism/liberalism of one's religious beliefs, stratified by gender.

**Usage**

stemcell

**Format**

A data frame with 24 rows and 4 variables:

**research** opinion about funding stem cell research with levels definitely, probably, probably not, definitely not

**gender** the gender of the respondent with levels female and male

**religion** the fundamentalism/liberalism of one's religious beliefs with levels fundamentalist, moderate, liberal

**frequency** the number of times a respondent fell in each of the combinations of levels for research, religion and gender

**Source**

The stemcell data set is analysed in Agresti (2010, Subsection~4.1.5).

**References**

Agresti, A. (2010). *\*Analysis of Ordinal Categorical Data\** (2nd edition). Wiley Series in Probability and Statistics. Wiley

---

summary.brglmFit      summary method for [brglmFit](#) objects

---

## Description

summary method for [brglmFit](#) objects

## Usage

```
## S3 method for class 'brglmFit'  
summary(object, dispersion = NULL,  
        correlation = FALSE, symbolic.cor = FALSE, ...)
```

## Arguments

object	an object of class "glm", usually, a result of a call to <a href="#">glm</a> .
dispersion	the dispersion parameter for the family used. Either a single numerical value or NULL (the default), when it is inferred from object (see 'Details').
correlation	logical; if TRUE, the correlation matrix of the estimated parameters is returned and printed.
symbolic.cor	logical. If TRUE, print the correlations in a symbolic form (see <a href="#">symnum</a> ) rather than as numbers.
...	further arguments passed to or from other methods.

## Details

The interface of the summary method for [brglmFit](#) objects is identical to that of [glm](#) objects. The summary method for [brglmFit](#) objects computes the p-values of the individual Wald statistics based on the standard normal distribution, unless the family is Gaussian, in which case a t distribution with appropriate degrees of freedom is used.

## See Also

[summary.glm](#) and [glm](#)

## Examples

```
## For examples see examples(brglmFit)
```

---

vcov.brglmFit	<i>Return the variance-covariance matrix for the regression parameters in a <a href="#">brglmFit</a> object</i>
---------------	---

---

**Description**

Return the variance-covariance matrix for the regression parameters in a [brglmFit](#) object

**Usage**

```
## S3 method for class 'brglmFit'  
vcov(object, model = c("mean", "full", "dispersion"),  
      ...)
```

**Arguments**

object	a fitted model object, typically. Sometimes also a <a href="#">summary()</a> object of such a fitted model.
model	character specifying for which component of the model coefficients should be extracted.
...	additional arguments for method functions. For the <a href="#">glm</a> method this can be used to pass a dispersion parameter.

# Index

## \*Topic **datasets**

- alligators, 2
  - coalition, 16
  - endometrial, 20
  - lizards, 21
  - stemcell, 26
- alligators, 2
- bracl, 3, 25
- brglm2, 5
- brglm2-package (brglm2), 5
- brglm\_control (brglmControl), 6
- brglm\_fit (brglmFit), 8
- brglmControl, 4, 6, 7, 9, 12
- brglmFit, 3, 6–8, 8, 12, 13, 17, 19, 27, 28
- brmultinom, 5, 9, 12, 13, 25
- check\_infinite\_estimates, 5, 9, 14
- check\_infinite\_estimates.glm, 15
- checkInfiniteEstimates  
(check\_infinite\_estimates.glm),  
15
- coalition, 16
- confint.brglmFit, 17
- detect\_separation, 5, 9, 17, 17, 18–20
- detect\_separation\_control, 18, 19
- detectSeparation (detect\_separation), 17
- detectSeparationControl  
(detect\_separation\_control), 19
- endometrial, 20
- family, 8, 9, 18
- formula, 3, 12
- glm, 6, 8–10, 14, 15, 17–19, 27, 28
- glm.fit, 7, 8, 10, 19
- lizards, 21
- mis, 9, 22, 22
- model.offset, 8, 18
- multinom, 5, 13, 15
- offset, 8, 18
- power, 9
- predict.bracl, 23
- predict.brmultinom, 24
- print.detect\_separation  
(detect\_separation), 17
- quasi, 9
- quasibinomial, 9
- quasipoisson, 9
- residuals.bracl (residuals.brmultinom),  
25
- residuals.brmultinom, 25
- stemcell, 26
- summary, 28
- summary.brglmFit, 27
- summary.glm, 27
- symnum, 27
- vcov.brglmFit, 28