Package ‘detectseparation’

April 19, 2026

Title Detect and Check for Separation and Infinite Maximum Likelihood Estimates

Version 0.4.0

Description Provides pre-fit and post-fit methods for detecting separation and infinite maximum likelihood estimates in generalized linear models with categorical responses. The pre-fit methods apply on binomial-response generalized liner models such as logit, probit and cloglog regression, and can be directly supplied as fitting methods to the glm() function. They solve the linear programming problems for the detection of separation developed in Konis (2007, <https://ora.ox.ac.uk/objects/uuid:8f9ee0d0-d78e-4101-9ab4-f9cbceed2a2a>) using 'ROI' <https://cran.r-project.org/package=ROI> or 'lpSolveAPI' <https://cran.r-project.org/package=lpSolveAPI>. The post-fit methods apply to models with categorical responses, including binomial-response generalized linear models and multinomial-response models, such as baseline category logits and adjacent category logits models; for example, the models implemented in the 'brglm2' <https://cran.r-project.org/package=brglm2> package. The post-fit methods successively refit the model with increasing number of iteratively reweighted least squares iterations, and monitor the ratio of the estimated standard error for each parameter to what it has been in the first iteration. According to the results in Lesaffre & Albert (1989, <https://www.jstor.org/stable/2345845>), divergence of those ratios indicates data separation.

URL https://github.com/ikosmidis/detectseparation

BugReports https://github.com/ikosmidis/detectseparation/issues

Imports ROI, ROI.plugin.lpsolve, lpSolveAPI, pkgload

Depends R (>= 3.3.0)

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 7.3.3

Suggests knitr, rmarkdown, covr, brglm2, AER, tinytest, ROI.plugin.ecos, ROI.plugin.glpk, ROI.plugin.alabama, ROI.plugin.neos

VignetteBuilder knitr

NeedsCompilation no

Author Ioannis Kosmidis [aut, cre] (ORCID: <https://orcid.org/0000-0003-1556-0302>),
Dirk Schumacher [aut],
Florian Schwendinger [aut],
Kjell Konis [ctb]

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

Repository CRAN

Date/Publication 2026-04-19 05:10:24 UTC

Contents

`check_infinite_estimates` *Generic method for checking for infinite estimates*

Description

Generic method for checking for infinite estimates

Usage

```
check_infinite_estimates(object, ...)

checkInfiniteEstimates(object, ...)
```

Arguments

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

See Also

check_infinite_estimates.glm

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 glm() call.
nsteps	starting from maxit = 1, the GLM is refitted for maxit = 2, maxit = 3, ..., maxit = nsteps. Default value is 20.
...	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.

check_infinite_estimates() can also be used to identify the occurrence of infinite estimates in baseline category logit models for nominal responses (see brglm2::brmultinom()), and adjacent category logit models for ordinal responses (see brglm2::bracl()).

Value

An object of class inf_check that has a plot() method.

A matrix inheriting from class inf_check, with nsteps rows and p columns, where p is the number of model parameters. A plot() method is provided for inf_check objects for the easy inspection of the ratios of the standard errors.

Note

For the definition of complete and quasi-complete separation, see Albert and Anderson (1984). Kosmidis and Firth (2021) prove that the reduced-bias estimator that results by the penalization of the logistic regression log-likelihood by Jeffreys prior takes always finite values, even when some of the maximum likelihood estimates are infinite. The reduced-bias estimates can be computed using the **brglm2** R package.

References

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

Kosmidis I. and Firth D. (2021). Jeffreys-prior penalty, finiteness and shrinkage in binomial-response generalized linear models. *Biometrika*, **108**, 71–82

See Also

nnet::multinom(), detect_separation(), brglm2::brmultinom(), brglm2::bracl()

Examples

```
# endometrial data from Heinze \& Schemper (2002) (see ?endometrial)
data("endometrial", package = "detectseparation")
endometrial_ml <- glm(HG ~ NV + PI + EH, data = endometrial,
                      family = binomial("probit"))
# clearly the maximum likelihood estimate for the coefficient of
# NV is infinite
(estimates <- check_infinite_estimates(endometrial_ml))
plot(estimates)



# Aligator data (Agresti, 2002, Table~7.1)
if (requireNamespace("brglm2", quietly = TRUE)) {
    data("alligators", package = "brglm2")
    all_ml <- brglm2::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
    plot(check_infinite_estimates(all_ml))
    # Bias reduction the brglm2 R packages can be used to get finite estimates
    all_br <- brglm2::brmultinom(foodchoice ~ size + lake , weights = round(freq/3),
                         data = alligators, ref = 1)
    plot(check_infinite_estimates(all_br))
}
```

detectseparation — *detectseparation: Methods for Detecting and Checking for Separation and Infinite Maximum Likelihood Estimates*

Description

detectseparation provides pre-fit and post-fit methods for the detection of separation and of infinite maximum likelihood estimates in binomial response generalized linear models.

Details

The key methods are `detect_separation()` and `check_infinite_estimates()`.

Author(s)

Maintainer: Ioannis Kosmidis <ioannis.kosmidis@warwick.ac.uk> (ORCID)

Authors:

- Dirk Schumacher <mail@dirk-schumacher.net>
- Florian Schwendinger <FlorianSchwendinger@gmx.at>

Other contributors:

- Kjell Konis <kjell.konis@me.com> [contributor]

See Also

`detect_separation()`, `check_infinite_estimates()`

detect_infinite_estimates — *Detect Infinite Estimates*

Description

Method for `glm()` that detects infinite components in the maximum likelihood estimates of generalized linear models with binomial responses.

Usage

```
detect_infinite_estimates(
  x,
  y,
  weights = NULL,
  start = NULL,
  etastart = NULL,
  mustart = NULL,
  offset = NULL,
  family = gaussian(),
  control = list(),
  intercept = TRUE,
  singular.ok = TRUE
)

detectInfiniteEstimates(
  x,
  y,
  weights = NULL,
  start = NULL,
  etastart = NULL,
  mustart = NULL,
  offset = NULL,
  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 *a priori* 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 `offset` terms can be included in the formula instead or as well, and if more than one is specified their sum is used. See `model.offset`.
`family`	a description of the error distribution and link function to be used in the model. For `glm` this can be a character string naming a family function, a family function or the result of a call to a family function. For `glm.fit` only the third option is supported. (See `family` for details of family functions.)

control	a list of parameters controlling separation detection. See detect_separation_control() for details.
intercept	logical. Should an intercept be included in the *null* model?
singular.ok	logical. If FALSE, a singular model is an error.

Details

For binomial-response generalized linear models with "log" link, separated data allocations do not necessarily lead to infinite maximum likelihood estimates. For this reason, for models with the "log" link detect_infinite_estimates() relies on an alternative linear optimization model developed in Schwendinger et al. (2021). For all the other supported links detect_infinite_estimates() relies on the linear programming methods developed in Konis (2007). See detect_separation() for definitions and details.

detect_infinite_estimates() is a wrapper to the functions separator_ROI(), separator_lpSolveAPI() (a modified version of the separator() function from the **safeBinaryRegression** R package), and dielb_ROI(). The first two are used for non-log links, while dielb_ROI() is used for the "log" link.

The coefficients() method extracts a vector of values for each of the model parameters under the following convention: 0 if the maximum likelihood estimate of the parameter is finite, and Inf or -Inf if the maximum likelihood estimate of the parameter is plus or minus infinity. This convention makes it easy to adjust the maximum likelihood estimates to their actual values by element-wise addition.

detect_infinite_estimates() can be passed directly as a method to the glm() function. See, examples.

detectInfiniteEstimates() is an alias for detect_infinite_estimates().

Author(s)

Ioannis Kosmidis [aut, cre] <ioannis.kosmidis@warwick.ac.uk>, Florian Schwendinger [aut] <FlorianSchwendinger@gmx.at>, Dirk Schumacher [aut] <mail@dirk-schumacher.net>, Kjell Konis [ctb] <kjell.konis@me.com>

References

Silvapulle, M. J. (1981). On the Existence of Maximum Likelihood Estimators for the Binomial Response Models. Journal of the Royal Statistical Society. Series B (Methodological), 43(3), 310–313. https://www.jstor.org/stable/2984941

Konis 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

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

Kosmidis I. and Firth D. (2021). Jeffreys-prior penalty, finiteness and shrinkage in binomial-response generalized linear models. *Biometrika*, **108**, 71–82. doi:10.1093/biomet/asaa052

Schwendinger, F., Grün, B. & Hornik, K. (2021). A comparison of optimization solvers for log binomial regression including conic programming. *Computational Statistics*, **36**, 1721–1754. doi:10.1007/s00180021010845

See Also

glm(), detect_separation(), check_infinite_estimates(), brglm2::brglm_fit()

Examples

```
# The classical example given in Silvapulle (1981) can be utilized
# to show that for the log-binomial model there exist data allocations
# which are separated but produce finite estimates.
data("silvapulle1981", package = "detectseparation")

# Since the data is separated the MLE does not exist for the logit link.
glm(y ~ ghqs, data = silvapulle1981, family = binomial(),
    method = "detect_infinite_estimates")

# However, for the log link all components of the MLE are finite.
glm(y ~ ghqs, data = silvapulle1981, family = binomial("log"),
    method = "detect_infinite_estimates")
# The maximum likelihood fit is
glm(y ~ ghqs, data = silvapulle1981, family = binomial("log"), start = c(-1, 0))
```

---

detect_separation   *Detect Separation*

Description

Method for glm() that tests for data separation and finds which parameters have infinite maximum likelihood estimates in generalized linear models with binomial responses

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 = NULL,
  start = NULL,
  etastart = NULL,
  mustart = NULL,
  offset = NULL,
  family = gaussian(),
  control = list(),
  intercept = TRUE,
  singular.ok = TRUE
```

```
)

detectSeparation(
  x,
  y,
  weights = NULL,
  start = NULL,
  etastart = NULL,
  mustart = NULL,
  offset = NULL,
  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 *a priori* 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 `offset` terms can be included in the formula instead or as well, and if more than one is specified their sum is used. See `model.offset`.
`family`	a description of the error distribution and link function to be used in the model. For `glm` this can be a character string naming a family function, a family function or the result of a call to a family function. For `glm.fit` only the third option is supported. (See `family` for details of family functions.)
`control`	a list of parameters controlling separation detection. See `detect_separation_control()` for details.
`intercept`	logical. Should an intercept be included in the *null* model?
`singular.ok`	logical. If `FALSE`, a singular model is an error.

Details

Following the definitions in Albert and Anderson (1984), the data for a binomial-response generalized linear model with logistic link exhibit quasi-complete separation if there exists a non-zero parameter vector β such that $X^0\beta \leq 0$ and $X^1\beta \geq 0$, where X^0 and X^1 are the matrices formed by the rows of the model matrix X corresponding to zero and non-zero responses, respectively.

The data exhibits complete separation if there exists a parameter vector β such that the aforementioned conditions are satisfied with strict inequalities. If there are no vectors β that can satisfy the conditions, then the data points are said to overlap.

If the inverse link function $G(t)$ of a generalized linear model with binomial responses is such that $\log G(t)$ and $\log(1 - G(t))$ are concave and the model has an intercept parameter, then overlap is a necessary and sufficient condition for the maximum likelihood estimates to be finite (see Silvapulle, 1981 for a proof). Such link functions are, for example, the logit, probit and complementary log-log.

`detect_separation()` determines whether or not the data exhibits (quasi-)complete separation. Then, if separation is detected and the link function $G(t)$ is such that $\log G(t)$ and $\log(1 - G(t))$ are concave, the maximum likelihood estimates has infinite components.

`detect_separation()` is a wrapper to the `detect_infinite_estimates()` method. Separation detection, as separation is defined above, takes place using the linear programming methods in Konis (2007) regardless of the link function. The output of those methods is also used to determine which estimates are infinite, unless the link is `"log"`. In the latter case the linear programming methods in Schwendinger et al. (2021) are called to establish if and which estimates are infinite. If the link function is not one of `"logit"`, `"log"`, `"probit"`, `"cauchit"`, `"cloglog"` then a warning is issued.

If `separation_type = TRUE` in `detect_separation_control()`, then, whenever separation is detected, `detect_separation()` attempts to distinguish between complete and quasi-complete separation by solving an additional linear program that maximizes the minimum transformed margin. If x_i is the ith row of the model matrix and $\tilde{y}_i = -1 + 2y_i$, where y_i is the ith Bernoulli response (the representation to which any binomial data supplied by the user are transformed internally), then $\bar{x}_i = \tilde{y}_i x_i$. The additional linear program that is solved maximizes t subject to $\bar{X}\beta \geq t\mathbf{1}$ and $-1 \leq \beta_j \leq 1$. This is equivalent to solving $\max_\beta \min_i(\bar{x}_i^\top \beta)$ subject to $-1 \leq \beta_j \leq 1$ for all j. A positive optimal value for t implies complete separation, while an optimal value of zero implies quasi-complete separation provided that separation has already been detected. See Konis (2007, Section 1.3 and Chapter 4) for the definitions of separation and the transformed linear programming formulation.

The `coefficients()` method extracts a vector of values for each of the model parameters under the following convention: `0` if the maximum likelihood estimate of the parameter is finite, and `Inf` or `-Inf` if the maximum likelihood estimate of the parameter is plus or minus infinity. This convention makes it easy to adjust the maximum likelihood estimates to their actual values by element-wise addition.

`detect_separation()` can be passed directly as a method to the `glm()` function. See, examples.

`detectSeparation()` is an alias for `detect_separation()`.

Value

A list that inherits from class `detect_separation`, `glm` and `lm`. A `print` method is provided for `detect_separation` objects. If `detect_separation_control(separation_type = TRUE)` is used and separation is detected, then the returned object has a `complete` component, which is `TRUE` for complete separation and `FALSE` for quasi-complete separation. Otherwise, the `complete` component is `NULL`.

Note

For the definition of complete and quasi-complete separation, see Albert and Anderson (1984). Kosmidis and Firth (2021) prove that the reduced-bias estimator that results by the penalization of the logistic regression log-likelihood by Jeffreys prior takes always finite values, even when some of the maximum likelihood estimates are infinite. The reduced-bias estimates can be computed using the **brglm2** R package.

`detect_separation()` was designed in 2017 by Ioannis Kosmidis for the **brglm2** R package, after correspondence with Kjell Konis, and a port of the `separator()` function had been included in **brglm2** under the permission of Kjell Konis. In 2020, `detect_separation()` and `check_infinite_estimates()` were moved outside **brglm2** into the dedicated **detectseparation** package. Dirk Schumacher authored the `separator_ROI()` function, which depends on the **ROI** R package and is now the default implementation used for detecting separation. In 2022, Florian Schwendinger authored the `dielb_ROI()` function for detecting infinite estimates in log-binomial regression, and, with Ioannis Kosmidis, they refactored the codebase to properly accommodate for the support of log-binomial regression.

Author(s)

Ioannis Kosmidis [aut, cre] <ioannis.kosmidis@warwick.ac.uk>, Dirk Schumacher [aut] <mail@dirk-schumacher.net> Florian Schwendinger [aut] <FlorianSchwendinger@gmx.at>, Kjell Konis [ctb] <kjell.konis@me.com>

References

Konis 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

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

Kosmidis I. and Firth D. (2021). Jeffreys-prior penalty, finiteness and shrinkage in binomial-response generalized linear models. *Biometrika*, **108**, 71–82. doi:10.1093/biomet/asaa052

Silvapulle, M. J. (1981). On the Existence of Maximum Likelihood Estimators for the Binomial Response Models. *Journal of the Royal Statistical Society. Series B (Methodological)*, **43**, 310–313. https://www.jstor.org/stable/2984941

Schwendinger, F., Grün, B. & Hornik, K. (2021). A comparison of optimization solvers for log binomial regression including conic programming. *Computational Statistics*, **36**, 1721–1754. doi:10.1007/s00180021010845

See Also

`glm()`, `detect_infinite_estimates()`, `check_infinite_estimates()`, `brglm2::brglm_fit()`

Examples

```
# endometrial data from Heinze \& Schemper (2002) (see ?endometrial)
data("endometrial", package = "detectseparation")
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"))

# If we want to further check for the type of separation (complete
# or quasi-complete) we can do
endometrial_sep_type <- update(endometrial_sep, separation_type = TRUE)
endometrial_sep
endometrial_sep_type


# Example inspired by unpublished microeconometrics lecture notes by
# Achim Zeileis https://eeecon.uibk.ac.at/~zeileis/
# The maximum likelihood estimate of sourhernyes is infinite
if (requireNamespace("AER", quietly = TRUE)) {
    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 revealed by the divergence of the NV column of the
    # result from the more computationally intensive check
    plot(check_infinite_estimates(murder_glm))
    # Mean bias reduction via adjusted scores results in finite estimates
    if (requireNamespace("brglm2", quietly = TRUE))
        update(murder_glm, method = brglm2::brglm_fit)
}
```

---

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(
  implementation = c("ROI", "lpSolveAPI"),
  solver = "lpsolve",
  linear_program = c("primal", "dual"),
  purpose = c("find", "test"),
```

```
  tolerance = 1e-04,
  solver_control = list(),
  separation_type = FALSE
)

detectSeparationControl(
  implementation = c("ROI", "lpSolveAPI"),
  solver = "lpsolve",
  linear_program = c("primal", "dual"),
  purpose = c("find", "test"),
  tolerance = 1e-04,
  solver_control = list(),
  separation_type = FALSE
)
```

Arguments

`implementation` should the implementation using `ROI` or the implementation using `lpSolveAPI` be used? Default is `ROI`.

`solver` should the linear program be solved using the `"lpsolve"` (using the `ROI.plugin.lpsolve` package; default) or another solver? Alternative solvers are `"glpk"`, `"cbc"`, `"clp"`, `"cplex"`, `"ecos"`, `"gurobi"`, `"scs"`, `"symphony"`. If the corresponding package `ROI.plugin.[solver]` is not installed then the user will be prompted to install it before continuing.

`linear_program` should `detect_separation()` solve the `"primal"` (default) or `"dual"` linear program for separation detection? Only relevant if `implementation = "lpSolveAPI"`.

`purpose` should `detect_separation()` simply `"test"` for separation or also `"find"` (default) which parameters are infinite? Only relevant if `implementation = "lpSolveAPI"`.

`tolerance` maximum absolute variable value from the linear program, before separation is declared. Default is `1e-04`.

`solver_control` a list with additional control parameters for the `"solver"`. This is solver specific, so consult the corresponding documentation. Default is `list()` unless `solver` is `"alabama"` when the default is `list(start = rep(0, p))`, where `p` is the number of parameters.

`separation_type` logical. Should `detect_separation()` attempt to distinguish complete from quasi-complete separation after separation has been detected? If `TRUE` and separation is detected, then an additional linear program is solved. Default is `FALSE`.

Value

A list with the supplied `linear_program`, `solver`, `solver_control`, `purpose`, `tolerance`, `separation_type`, and `implementation`. The returned list is intended to be passed to the `control` argument of `detect_separation()` and `detect_infinite_estimates()`.

References

Konis 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

Examples

```
data("endometrial", package = "detectseparation")
ctrl <- detect_separation_control(separation_type = TRUE)
glm(HG ~ NV + PI + EH, data = endometrial,
    family = binomial("logit"),
    method = "detect_separation",
    control = ctrl)
```

---

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`: neovasculization with coding 0 for absent and 1 for present
- `PI`: pulsality index of arteria uterina
- `EH`: endometrium height
- `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 https://users.stat.ufl.edu/~aa/glm/data/. The latter link provides the data sets used in Agresti (2015).

The endometrial data set was first analyzed 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. doi:10.1002/sim.1047.

Kosmidis I, Firth D (2021). Jeffreys-prior penalty, finiteness and shrinkage in binomial-response generalized linear models. *Biometrika*, **108**, 71-82. doi:10.1093/biomet/asaa052.

See Also

`detect_separation()`

`lizards` *Habitat preferences of lizards*

Description

Habitat preferences of lizards

Usage

```
lizards
```

Format

A data frame with 23 rows and 6 columns:

- `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`

The 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.

Source

McCullagh P, 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.

See Also

`detect_separation()`

`silvapulle1981` *Separation Example Presented in Silvapulle (1981)*

Description

Separation example presented in Silvapulle (1981).

Usage

```
silvapulle1981
```

Format

A data frame with 35 rows and 2 variables:

- `y`: a factor with the levels `case` and `none-case`, giving the outcome of a standardized psychiatric interview
- `ghqs`: an integer giving the general health questionnaire score.

References

Silvapulle, M. J. (1981). On the Existence of Maximum Likelihood Estimators for the Binomial Response Models. Journal of the Royal Statistical Society. Series B (Methodological), 43(3), 310–313. https://www.jstor.org/stable/2984941

See Also

`detect_infinite_estimates()`

Index