Package ‘glm2’

May 25, 2018

Type Package
Title Fitting Generalized Linear Models
Version 1.2
Depends R (>= 3.5.0), stats
Description Fits generalized linear models using the same model specification as glm in the stats package, but with a modified default fitting method that provides greater stability for models that may fail to converge using glm.
License GPL (>= 2)
LazyData true
NeedsCompilation no
Author Ian Marschner [aut] (using code from glm and glm.fit in the stats package),
Mark W. Donoghoe [cre, ctb]
Maintainer Mark W. Donoghoe <markdonoghoe@gmail.com>
Repository CRAN
Date/Publication 2018-05-25 07:37:53 UTC

R topics documented:

glm2-package .............................................. 2
crabs ....................................................... 2
glm.fit2 ................................................... 3
glm2 ....................................................... 5
heart ....................................................... 9

Index 11
**glm2-package**  
*Fitting Generalized Linear Models*

**Description**

Fits generalized linear models using the same model specification as `glm` in the `stats` package, but with a modified default fitting method. The method provides greater stability for models that may fail to converge using `glm`.

**Details**

- **Package:** glm2
- **Type:** Package
- **Version:** 1.2
- **License:** GPL (>=2)
- **LazyData:** true

There are two functions in the package, `glm2` and `glm.fit2`. The `glm2` function fits generalized linear models using the same model specification as `glm` in the `stats` package. It is identical to `glm` except for minor modifications to change the default fitting method. The `glm.fit2` function provides the default fitting method for `glm2`. It is identical to `glm.fit` in the `stats` package, except for modifications to the computational method that provide more stable convergence. Normally only `glm2` would be called directly, although like `glm.fit`, `glm.fit2` can be called directly. It can also be passed to `glm` as an alternative fitting method, using the `method` argument. See Marschner (2011) for a discussion of the need for a modified fitting method.

**Author(s)**

Ian Marschner (using code from `glm` and `glm.fit` in the `stats` package)

Maintainer: Mark W. Donoghoe <markdonoghoe@gmail.com>

**References**


---

**crabs**  
*Horseshoe Crab Data*

**Description**

This data set is derived from Agresti (2007, Table 3.2, pp.76-77). It gives 4 variables for each of 173 female horseshoe crabs. Also provided are two random samples of the data with replacement, which are useful for illustrating the convergence properties of `glm` and `glm2`.
Usage

data(crabs)

Format

A data frame with 173 observations on the following 6 variables:

- **Satellites**: number of male partners in addition to the female’s primary partner
- **Width**: width of the female in centimeters
- **Dark**: a binary factor indicating whether the female has dark coloring (yes or no)
- **GoodSpine**: a binary factor indicating whether the female has good spine condition (yes or no)
- **Rep1**: a random sample with replacement from 1:173
- **Rep2**: a second random sample with replacement from 1:173

Details

The variables Dark and GoodSpine are derived from the raw data. In the notation of Table 3.2 of Agresti (2007), Dark = yes corresponds to C>3 and GoodSpine = yes corresponds to S<3. The two random samples Rep1 and Rep2 can be used to provide random samples with replacement from the full data set. These two random samples are useful for illustrating the convergence properties of glm and glm2; see examples in the help documentation for glm2.

References


glm.fit2

Generalized Linear Models Fitting Method

Description

glm.fit2 is identical to glm.fit in the stats package, except for a modification to the computational method that provides improved convergence properties. It is the default fitting method for glm2 and can also be used as an alternative fitting method for glm, instead of the default method glm.fit.

Usage

```r
glm.fit2(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` as for `glm.fit`
- `y` as for `glm.fit`
- `weights` as for `glm.fit`
- `start` as for `glm.fit`
- `etastart` as for `glm.fit`
- `mustart` as for `glm.fit`
- `offset` as for `glm.fit`
- `family` as for `glm.fit`
- `control` as for `glm.fit`
- `intercept` as for `glm.fit`
- `singular.ok` as for `glm.fit`

Details

`glm.fit2` is a modified version of `glm.fit` in the `stats` package. The computational method in `glm.fit2` uses a stricter form of step-halving to deal with numerical instability in the iteratively reweighted least squares algorithm. Whereas `glm.fit` uses step-halving to correct divergence and parameter space violations, `glm.fit2` additionally uses step-halving to force the model deviance to decrease at each iteration, which improves the convergence properties. Like `glm.fit`, `glm.fit2` usually would not be called directly. Instead, it is called by `glm2` as the default fitting method. Like `glm.fit`, it is possible to call `glm.fit2` directly if the response vector and design matrix have already been calculated, in which case it may be more efficient than calling `glm2`. It can also be passed to `glm` in the `stats` package, using the `method` argument, providing an alternative to the default fitting method `glm.fit`. See Marschner (2011) for a discussion of the need for a modified fitting method.

Value

The value returned by `glm.fit2` has exactly the same documentation as the value returned by `glm.fit`.

Author(s)

`glm.fit2` uses the code from `glm.fit`, whose authors are listed in the help documentation for the `stats` package. Modifications to this code were made by Ian Marschner.

References


See Also

`glm.fit`
Examples

library(glm2)

#--- logistic regression null model -------------------#
# (intercept estimate = log(0.75/0.25) = 1.098612)

y <- c(1,1,1,0)
x <- rep(1,4)

#--- divergence of glm.fit to infinite estimate ---#
fit1 <- glm.fit(x,y, family=binomial(link="logit"),start=-1.81)
fit2 <- glm.fit2(x,y, family=binomial(link="logit"),start=-1.81)
print.noquote(c(fit1$coef,fit2$coef))

glm2

Fitting Generalized Linear Models

Description

Fits generalized linear models using the same model specification as glm in the stats package, but with a modified default fitting method. The method provides greater stability for models that may fail to converge using glm.

Usage

glm2(formula, family = gaussian, data, weights, subset, na.action,
     start = NULL, etastart, mustart, offset, control = list(...),
     model = TRUE, method = "glm.fit2", x = FALSE, y = TRUE,
     singular.ok = TRUE, contrasts = NULL, ...)

Arguments

  formula          as for glm
  family           as for glm
  data             as for glm
  weights          as for glm
  subset           as for glm
  na.action        as for glm
  start            as for glm
  etastart         as for glm
  mustart          as for glm
  offset           as for glm
  control          as for glm except by default control is passed to glm.fit2 instead of glm.fit
model as for glm
method the method used in fitting the model. The default method "glm.fit2" uses iteratively reweighted least squares with modified step-halving that forces the deviance to decrease at each iteration; see help documentation for glm.fit2. As in glm, the alternative method "model.frame" returns the model frame and does no fitting.
x as for glm
y as for glm
singular.ok as for glm
contrasts as for glm
... as for glm

Details

glm2 is a modified version of glm in the stats package. It fits generalized linear models using the same model specification as glm. It is identical to glm except for minor modifications to change the default fitting method. The default method uses a stricter form of step-halving to force the deviance to decrease at each iteration and is implemented in glm.fit2. Like glm, user-supplied fitting functions can be used with glm2 by passing a function or a character string naming a function to the method argument. See Marschner (2011) for a discussion of the need for a modified fitting method.

Value

The value returned by glm2 has exactly the same documentation as the value returned by glm, except for:

method the name of the fitter function used, which by default is "glm.fit2".

Author(s)

glm2 uses the code from glm, whose authors are listed in the help documentation for the stats package. Modifications to this code were made by Ian Marschner.

References


See Also

glm
Examples

```r
library(glm2)
data(crabs)
data(heart)

# EXAMPLE 1: logistic regression null model
# (behaviour of glm and glm2 for different starting values)

y <- c(1, 1, 1, 0)
# intercept estimate = log(0.75/0.25) = 1.098612

# identical behaviour ---#
fit1 <- glm(y ~ 1, family = binomial(link = "logit"),
            control = glm.control(trace = TRUE))
fit2 <- glm2(y ~ 1, family = binomial(link = "logit"),
            control = glm.control(trace = TRUE))
print(noquote(c(fit1$coef, fit2$coef)))

# convergence via different paths ---#
fit1 <- glm(y ~ 1, family = binomial(link = "logit"), start = -1.75,
            control = glm.control(trace = TRUE))
fit2 <- glm2(y ~ 1, family = binomial(link = "logit"), start = -1.75,
            control = glm.control(trace = TRUE))
print(noquote(c(fit1$coef, fit2$coef)))

# divergence of glm to infinite estimate ---#
fit1 <- glm(y ~ 1, family = binomial(link = "logit"), start = -1.81)
fit2 <- glm2(y ~ 1, family = binomial(link = "logit"), start = -1.81)
print(noquote(c(fit1$coef, fit2$coef)))

# EXAMPLE 2: identity link Poisson (successful boundary convergence
# using 4 identical approaches in glm and glm2 with the method argument)

satellites <- crabs$Satellites
width.shifted <- crabs$Width - min(crabs$Width)
dark <- crabs$Dark
goodspine <- crabs$GoodSpine

fit1 <- glm(satellites ~ width.shifted + factor(dark) + factor(goodspine),
            family = poisson(link = "identity"), start = rep(1, 4))

fit2 <- glm2(satellites ~ width.shifted + factor(dark) + factor(goodspine),
             family = poisson(link = "identity"), start = rep(1, 4))

fit1.eq <- glm2(satellites ~ width.shifted + factor(dark) + factor(goodspine),
                family = poisson(link = "identity"), start = rep(1, 4),
                method = "glm.fit")
```
fit2.eq <- glm(satellites ~ width.shifted + factor(dark) + factor(goodspine),
  family = poisson(link="identity"), start = rep(1,4), method = "glm.fit2")

noquote(c("deviances: ",fit1$dev,fit2$dev,fit1.eq$dev,fit2.eq$dev))
noquote(c("converged: ",fit1$conv,fit2$conv,fit1.eq$conv,fit2.eq$conv))
noquote(c("boundary: ",fit1$bound,fit2$bound,fit1.eq$bound,fit2.eq$bound))

# EXAMPLE 3: identity link Poisson (periodic non-convergence in glm)

# example S: identity link poisson (periodic non-Mconvergence in glm)

R1 <- crabs$Rep1
satellites <- crabs$Satellites[R1]
width.shifted <- crabs$Width[R1] - min(crabs$Width)
dark <- crabs$Dark[R1]
goodspine <- crabs$GoodSpine[R1]

fit1 <- glm(satellites ~ width.shifted + factor(dark) + factor(goodspine),
  family = poisson(link="identity"), start = rep(1,4),
  control = glm.control(trace=TRUE))

fit2 <- glm2(satellites ~ width.shifted + factor(dark) + factor(goodspine),
  family = poisson(link="identity"), start = rep(1,4),
  control = glm.control(trace=TRUE))

noquote(c("deviances: ",fit1$dev,fit2$dev))
noquote(c("converged: ",fit1$conv,fit2$conv))

# EXAMPLE 4: log link binomial (periodic non-convergence in glm)

# example T: log link binomial (periodic non-Mconvergence in glm)

patients <- heart$Patients
deaths <- heart$Deaths
agegroup <- heart$AgeGroup
severity <- heart$Severity
delay <- heart$Delay
region <- heart$Region
start.p <- sum(deaths)/sum(patients)

fit1 <- glm(cbind(deaths,patients-deaths) ~ factor(agegroup) + factor(severity)
  + factor(delay) + factor(region), family = binomial(link="log"),
  start = c(log(start.p), rep(0,8)), control = glm.control(trace=TRUE,maxit=100))

fit2 <- glm2(cbind(deaths,patients-deaths) ~ factor(agegroup) + factor(severity)
  + factor(delay) + factor(region), family = binomial(link="log"),
  start = c(log(start.p), rep(0,8)), control = glm.control(trace=TRUE))

noquote(c("deviances: ",fit1$dev,fit2$dev))
noquote(c("converged: ",fit1$conv,fit2$conv))

# EXAMPLE 5: identity link Poisson (aperiodic non-convergence in glm)

heart

Heart Attack Data

Description

This data set is a cross-tabulation of data on 16949 individuals who experienced a heart attack (ASSENT-2 Investigators, 1999). There are 4 categorical factors each at 3 levels, together with the number of patients and the number of deaths for each observed combination of the factors. This data set is useful for illustrating the convergence properties of glm and glm2.

Usage

data(heart)

Format

A data frame with 74 observations on the following 6 variables.

Deaths  number of deaths
Patients number of patients
AgeGroup categorization of the age of the patients
Severity severity of the heart attack
Delay categorization of the time from heart attack to treatment
Region geographical region in which the patients were treated
Details

The variable AgeGroup groups the age of the patients as follows: 1 = <65 years, 2 = 65-75 years, 3 = >75 years. The variable Delay groups the time from heart attack onset to treatment as follows: 1 = <2 hours, 2 = 2-4 hours, 3 = >4 hours. The variable Severity denotes the severity of the heart attack using the Killip class: 1 = least severe (class I), 2 = middle severity (class II), 3 = most severe (class III or IV). The variable Region provides the geographical region in which the patients were treated: 1 = Western countries, 2 = Latin America, 3 = Eastern Europe. This data set is useful for illustrating the convergence properties of glm and glm2; see examples in the help documentation for glm2.

References

Index

*Topic datasets
  crabs, 2
  heart, 9
*Topic package
  glmRMpackage, 2

crabs, 2

glm, 5, 6
glm.fit, 4
glm.fit2, 3
glm2, 5
glm2-package, 2

heart, 9