Package ‘fastlogitME’

July 22, 2025

Type Package

Title Basic Marginal Effects for Logit Models

Version 0.1.0

Author Mathieu Steijn

Maintainer Mathieu Steijn <m.p.a.steijn@uu.nl>

Description
Calculates marginal effects based on logistic model objects such as 'glm' or 'speedglm' at the average (default) or at given values using finite differences. It also returns confidence intervals for said marginal effects and the p-values, which can easily be used as input in stargazer. The function only returns the essentials and is therefore much faster but not as detailed as other functions available to calculate marginal effects. As a result, it is highly suitable for large datasets for which other packages may require too much time or calculating power.

License GPL

Encoding UTF-8

LazyData true

Imports stats

Enhances speedglm

RoxygenNote 6.1.1

NeedsCompilation no

Repository CRAN

Date/Publication 2020-11-05 09:30:05 UTC

Contents

fastlogitME *Basic Marginal Effects for Logit Models*

Description

Calculates marginal effects based on logistic model objects such as ’glm’ or ’speedglm’ at the average (default) or at given values using finite differences. It also returns confidence intervals for said marginal effects and the p-values, which can easily be used as input in stargazer. The function only returns the essentials and is therefore much faster but not as detailed as other functions available to calculate marginal effects. As a result, it is highly suitable for large datasets for which other packages may require too much time or calculating power.

Usage

```
fastlogitME(model, at = NULL, vars = NULL, conf.band = .95)
```

Arguments

`model` a model object of the type "glm" or "speedglm" where the family argument is specified to be binomial(’logit’)

`at` A list containing names of variables and specific values at which to calculate the marginal effects. The default is to calculate these at the mean of each continuous variable and for the reference category of dummy variables.

`vars` A character string or vector of character strings indicating the names of the variables for which marginal effects should be calculated. The default is all variables in the model object.

`conf.band` The bandwidth of the confidence interval has to be between 0 and 1. The default is .95, which corresponds to a 95% confidence interval

Details

This function is a fast but basic alternative to other packages destinated to calculating marginal effects for logit models. It is particularly helpful when working with large datasets, where other options may take too much time and/or CPU for some uses. It is also compatible with speedglm, which also achieves time gains compared to glm.

The function calculates confidence intervals on the scale of the link function (log odds) before converting it to the response scale (probabilities) rather than estimating standard errors on the response scale because the latter is linear while the link function is actually non-linear. As a result, confidence intervals based on a standard error on the response scale, commonly derived from bootstrapping or the delta method, may exceed the logically possible range of 0% to 100% when estimating the probability of "succes" at certain values in the dataset, see the references for more information.

The results can easily be exported using stargazer. First one needs to use rbind to add a row of zeros as first row to the data.frame resulting from fastlogitME, this replaces the intercept stargazer is expecting to find.Use the logit model stated at the model argument of the function as input for stargazer. Replace the coefficients of this model by those in the column ME using the coef argument of stargazer. Set ci = TRUE to display confidence intervals instead of standard errors. Specify these confidence intervals using the columns Conflower and Confupper and the ci.custom argument of stargazer. Specify omit = "Constant" so the intercept is not reported. 3.

Value

A data.frame containg the name of the variable, the marginal effect, the upper bound of the confidence interval, the lower bound of the confidence interval, and the p-value.

Note

It is advisable to consider if a logistic regression is the best option in your specific case, see the sources under references for more information.

Author(s)

Mathieu Steijn <m.p.a.steijn@uu.nl>

References

Allison, P. (2012). Logistic Regression for Rare Events [Blog post].

Buteikis, A. (2020) Practical Econometrics and Data Science. Vilnius University

Cameron, A. C., & Trivedi, P. K. (2010). Microeconometrics using Stata (revised ed.). Number musr in Stata Press books. StataCorp LP.

Hellevik, O. (2009). Linear versus logistic regression when the dependent variable is a dichotomy. Quality & Quantity, 43(1):59-74.

King, G. and Zeng, L. (2001). Logistic Regression in Rare Events Data. Political Analysis, 9(02):137-163.

Simpson, G. (2018) Confidence intervals for GLMs [Blog post].

Wooldridge, J. M. (2002). Introductory Econometrics: A Modern Approach, 2003. New York: South-Western College Publishing.

Examples

```
# simulate some data
set.seed(12345)
n = 1000
x = rnorm(n)
z = sample(0:1, n, replace = TRUE)
cat = as.factor(sample(c("cat1", "cat2", "cat3"), n, replace = TRUE))
# binary outcome
y = ifelse(pnorm(1 + 0.5*x - 0.5*z + rnorm(n))>0.5, 1, 0)
data = data.frame(y,x,z,cat)
a<-glm(y ~ x + z + cat, data = data, family = binomial('logit'))
fastlogitME(a)
fastlogitME(a, at = list("x" = 1.2, "z" = 1, "catcat2" = 1), vars = c("x"), conf.band = .99)
```

Index