Marginal Effects for Generalized Linear Models: The mfx Package for R

Alan Fernihough
Queen's University Belfast

Abstract

mfx is an R package which provides functions that estimate a number of popular generalized linear models, returning marginal effects as output. This paper briefly describes the method used to compute these marginal effects and their associated standard errors, and demonstrates how this is implemented with **mfx** in R. I also illustrate how the package extends to incorporate the calculation of odds and incidence rate ratios for certain generalized linear models. Finally, I present an example showing how the output produced via **mfx** can be translated into LaTeX.

Keywords: Marginal effects, odds ratio, incidence rate ratio, generalized linear models, R, **mfx**.

1. Introduction

The Generalized Linear Model (GLM) is a modified version of the classic linear regression model typically estimated via Ordinary Least Squares (OLS).[1] Researchers will generally use a GLM approach when the response variable being modeled does not have a normally distributed error term. Since the absence of a normally distributed error term violates the Gauss-Markov assumptions, the use of a GLM is preferable in many scenarios.[2] The GLM works by permitting the regressors to be related to the response variable by means of a link function. For example, in cases where the response variable is binary (takes a value of either zero or one), the probit or logit link functions are commonly used because these functions bound the predicted response between zero and one.

One drawback associated with the GLM is that the estimated model coefficients cannot be directly interpreted as marginal effects (i.e., the change in the response variable predicted after a one unit change in one of the regressors), like in an OLS regression. The estimated coefficients are multiplicative effects, dependent on both the link function chosen for the GLM and other variables alongside their estimated coefficient values. Therefore, it is difficult for one to judge the magnitude of a GLM regression based on the estimated coefficient values.

The open-source R offers a number of functions that facilitate GLM estimation. Furthermore, two R packages are available that contain functions providing platform from which users can interpret an estimated GLM. The package **effects** (Fox *et al.* 2013), described in Fox (2003), contains a comprehensive array of functions that allow users to graphically illustrate a GLM in effect plots. While effect plots are arguably a better representation of the results, these

[1] McCullagh and Nelder (1989) provide a complete overview of the GLM.
[2] Since the error term is non-normal this induces heteroskedasticity.

plots may become unwieldy for researchers trying to display the effects for a large number of variables and/or multiple model specifications. In such cases, a table of marginal effect results may offer a more concise method of displaying results. The **erer** (Sun 2013) package allows users to calculate marginal effects for either a binary logit or probit model.

While the packages **effects** and **erer** host a number of functions aiding the interpretation of the GLM, the package described in this article, **mfx** (Fernihough 2014), contains important additional features that are useful in empirical research. First, **mfx** both estimates the GLM and calculates the associated marginal effects in one function. Second, **mfx** can estimate adjusted standard errors, robust to either heteroskedasticity or clustering. Third, **mfx** provides the user with the ability to estimate marginal effects for a variety of GLM specifications, namely: binary logit, binary probit, count Poisson, count negative binomial, and beta distributed responses. Fourth, since odds ratios or incidence rate ratios are more commonly used in certain academic disciplines, like epidemiology, **mfx** also contains functions that return these values instead of marginal effects. Fifth, **mfx** allows the user to decide if they want to compute either the marginal effects for the average individual in the sample or the "average partial effects" as advocated in Wooldridge (2002). Finally, the output produced in **mfx** can easily be accommodated using the **texreg** (Leifeld 2013), so that publication quality LaTeX tables can be generated with relative simplicity.

The paper proceeds as follows. Section 2 contains a brief overview on the methods by which marginal effects are computed. Section 3 outlines details of the software. Section 4 offers a worked example that demonstrates how to use the software in practice and how the output can be used to generate publication standard LaTeX tables. Finally, Section 5 summarizes the main contributions of the paper, highlights a number of the package's drawbacks, and offers possible areas for future development.

2. Marginal effects

Let $\mathsf{E}(y_i|\boldsymbol{x}_i)$ represent the expected value of a dependent variable y_i given a vector of explanatory variables $\boldsymbol{x}_i$, for an observation unit i. In the case where $\boldsymbol{y}$ is a linear function of $(x_1, \cdots, x_j) = \boldsymbol{X}$ and $\boldsymbol{y}$ is a continuous variable, the following model with k regressors can be estimated via OLS:

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}, \tag{1}$$

where $\boldsymbol{\epsilon}$ represents the error term, or

$$y_i = \beta_0 + \beta_1 x_{1i} + \cdots + \beta_j x_{ki} + \epsilon_i, \tag{2}$$

so the additive vector of predicted coefficients can be obtained from the usual computation: $\hat{\boldsymbol{\beta}} = (\boldsymbol{X}^\top \boldsymbol{X})^{-1} \boldsymbol{X}^\top \boldsymbol{y}$. From (1) and (2) it is straightforward to see that the marginal effect of the variable $\boldsymbol{x}_j$, where $j \in \{1, \cdots, k\}$ on the dependent variable is: $\partial \boldsymbol{y} / \partial \boldsymbol{x}_j = \beta_j$. In other words, a unit increase in the variable $\boldsymbol{x}_j$ increases the variable $\boldsymbol{y}$ by β_j units.

A GLM takes the following form:

$$g(\boldsymbol{y}) = \boldsymbol{X}\boldsymbol{\beta}, \tag{3}$$

where the link function $g(\cdot)$ transforms the expectation of the response to a linear equation. The function $g(\cdot)$ is invertible, and thus we can rewrite Equation 3:

$$\boldsymbol{y} = g^{-1}(\boldsymbol{X}\boldsymbol{\beta}), \tag{4}$$

so the inverse link function, also known as the mean function, is applied to the product of the regressors ($\boldsymbol{X}$) and the model coefficients ($\boldsymbol{\beta}$). Therefore, the GLM in Equation 4 can be seen as the linear regression model nested within a nonlinear transformation. The choice of $g(\cdot)$ should depend on the distribution of the response $\boldsymbol{y}$.

Since the GLM typically implies that the linear model inside a nonlinear function, one cannot directly infer the marginal effects from the estimated coefficients.[3] Alternatively, based on Equation 4, we can see that:

$$\frac{\partial \boldsymbol{y}}{\partial \boldsymbol{x}_j} = \beta_j \times \frac{\partial g^{-1}(\boldsymbol{X}\boldsymbol{\beta})}{\partial \boldsymbol{x}_j}. \tag{5}$$

Thus, the nonlinearity in the link function means that the marginal effect of $\boldsymbol{x}_j$ now depends on the derivative of the inverse link function, and contained within this function are all of the other regressors and their associated regression coefficient values.

Here we use the probit model as an example, although the calculations for other GLM approaches is similar. The link function for the probit is based on the inverse normal distribution, so:

$$\mathsf{P}(\boldsymbol{y} = 1|\boldsymbol{x}) = \int_{-\infty}^{\boldsymbol{X}\beta} \phi(z)dz = \Phi(\boldsymbol{X}\boldsymbol{\beta}), \tag{6}$$

where $\Phi(\cdot)$ and $\phi(\cdot)$ denote both the normal cumulative and probability density functions respectively. The marginal effect for a continuous variable in a probit model is:

$$\frac{\partial \boldsymbol{y}}{\partial \boldsymbol{x_j}} = \hat{\beta}_j \times \phi(\boldsymbol{X}\hat{\boldsymbol{\beta}}) \tag{7}$$

since $\Phi'(\cdot) = \phi(\cdot)$, so the marginal effect for a continuous variable $\boldsymbol{x}_j$ depends on all of the estimated $\hat{\boldsymbol{\beta}}$ coefficients, which are fixed, and the complete design matrix $\boldsymbol{X}$, the values for which are variable. Because the values for $\boldsymbol{X}$ vary, the marginal effects depend on the procedure one employs. The literature offers two common approaches (Kleiber and Zeileis 2008). The first, and simplest, calculates the marginal effects when each variable in the design matrix is at its average value. Otherwise known as the partial effects for the average individual (Greene 2008), they can be calculated as:

$$\frac{\partial \boldsymbol{y}}{\partial \boldsymbol{x_j}} = \hat{\beta}_j \times \phi(\bar{\boldsymbol{X}}\hat{\boldsymbol{\beta}}). \tag{8}$$

The alternative approach calculates the average partial effects (Wooldridge 2002) or average of the sample marginal effects (Kleiber and Zeileis 2008), by calculating a partial effect for each observation unit (where there are n observations) and then averaging:

$$\frac{\partial \boldsymbol{y}}{\partial \boldsymbol{x_j}} = \hat{\beta}_j \times \frac{\sum_{i=1}^{n} \phi(\boldsymbol{X}_i\hat{\boldsymbol{\beta}})}{n}. \tag{9}$$

Usually, the choice over which method one uses is unimportant as the difference in values returned by both methods is likely to be small (Greene 2008).

The partial effects calculation in Equation 5 is not applicable in cases where $\boldsymbol{x}_j$ is a binary/dummy variable like gender. This is because the derivative in Equation 5 is with respect

[3] In the case where $g(\cdot)$ is the identity function, the estimated GLM will be identical to the standard linear regression model.

to a infinitesimally small change in $\boldsymbol{x}_j$ not the binary change from zero to one. Fortunately, calculating the marginal effects in such instances is very straightforward. In the probit model where the j-th regressor is a dummy variable the partial effect for the average individual is simply:

$$\frac{\Delta \boldsymbol{y}}{\Delta \boldsymbol{x}_j} = \Phi(\bar{\boldsymbol{X}}^{-j}\hat{\boldsymbol{\beta}}^{-j} + \hat{\beta}_j) - \Phi(\bar{\boldsymbol{X}}^{-j}\hat{\boldsymbol{\beta}}^{-j}), \tag{10}$$

where $\bar{\boldsymbol{X}}^{-j}$ is a vector of the average values of the design matrix $\boldsymbol{X}$ that excludes the j-th variable. The corresponding sample marginal effect is:

$$\frac{\Delta \boldsymbol{y}}{\Delta \boldsymbol{x}_j} = \frac{\sum_{i=1}^{n} \Phi(\boldsymbol{X}_i^{-j}\hat{\boldsymbol{\beta}}_i^{-j} + \hat{\beta}_j) - \Phi(\boldsymbol{X}^{-j}\hat{\boldsymbol{\beta}}^{-j})}{n}. \tag{11}$$

All functions in **mfx** automatically detect dummy regressors and perform the calculation in either Equation 10 or Equation 11, depending on the type of marginal effect the user wants.

We have already seen that the marginal effect for the j-th regressor in a probit GLM, $\hat{\beta}_j \times \phi(\boldsymbol{X}\hat{\boldsymbol{\beta}})$, is a nonlinear function of $\hat{\boldsymbol{\beta}}$. Therefore, the standard errors that correspond to these marginal effects must be calculated via the delta method of finding approximations based on Taylor series expansions to the variance of functions of random variables:

$$\mathsf{VAR}[f(\boldsymbol{X}\hat{\boldsymbol{\beta}})] = \left[\frac{\partial f(\boldsymbol{X}\hat{\boldsymbol{\beta}})}{\partial \hat{\boldsymbol{\beta}}}\right]^{\top} \mathsf{VAR}[\hat{\boldsymbol{\beta}}] \left[\frac{\partial f(\boldsymbol{X}\hat{\boldsymbol{\beta}})}{\partial \hat{\boldsymbol{\beta}}}\right], \tag{12}$$

where f is the nonlinear transformation and $\mathsf{VAR}[\hat{\boldsymbol{\beta}}]$ is the usual variance-covariance of the estimated parameters. With respect to the probit model previously used the variance of the marginal effects (for the average individual) is:

$$\mathsf{VAR}[\hat{\boldsymbol{\beta}} \times \phi(\bar{\boldsymbol{X}}\hat{\boldsymbol{\beta}})] = \left[\frac{\partial[\hat{\boldsymbol{\beta}} \times \phi(\bar{\boldsymbol{X}}\hat{\boldsymbol{\beta}})]}{\partial \hat{\boldsymbol{\beta}}}\right]^{\top} \mathsf{VAR}[\hat{\boldsymbol{\beta}}] \left[\frac{\partial[\boldsymbol{\beta} \times \phi(\bar{\boldsymbol{X}}\hat{\boldsymbol{\beta}})]}{\partial \hat{\boldsymbol{\beta}}}\right], \tag{13}$$

and since

$$\frac{\partial[\hat{\boldsymbol{\beta}} \times \phi(\bar{\boldsymbol{X}}\hat{\boldsymbol{\beta}})]}{\partial \hat{\boldsymbol{\beta}}} = \phi(\bar{\boldsymbol{X}}\hat{\boldsymbol{\beta}}) \times [\boldsymbol{I_k} - \bar{\boldsymbol{X}}\hat{\boldsymbol{\beta}} \times (\hat{\boldsymbol{\beta}}\bar{\boldsymbol{X}})], \tag{14}$$

the probit marginal effect standard errors will be derived from the diagonal elements of the following matrix of derivatives:

$$\mathsf{VAR}[\hat{\boldsymbol{\beta}} \times \phi(\bar{\boldsymbol{X}}\hat{\boldsymbol{\beta}})] = [\phi(\bar{\boldsymbol{X}}\hat{\boldsymbol{\beta}})]^2 \times [\boldsymbol{I_k} - \bar{\boldsymbol{X}}\hat{\boldsymbol{\beta}} \times (\hat{\boldsymbol{\beta}}\bar{\boldsymbol{X}})][\mathsf{VAR}[\hat{\boldsymbol{\beta}}]][\boldsymbol{I_k} - \boldsymbol{X}\hat{\boldsymbol{\beta}} \times (\hat{\boldsymbol{\beta}}\bar{\boldsymbol{X}})] \tag{15}$$

for continuous regressors, and:

$$\mathsf{VAR}\left[\frac{\Delta \boldsymbol{y}}{\Delta \boldsymbol{x}_j}\right] = \phi(\bar{\boldsymbol{X}}^{-j}\hat{\boldsymbol{\beta}}^{-j} + \hat{\beta}_j) \times \mathsf{VAR}[\hat{\beta}_j] \times \phi(\bar{\boldsymbol{X}}^{-j}\hat{\boldsymbol{\beta}}^{-j} + \hat{\beta}_j) \tag{16}$$

for the j-th discrete regressor, when the user is calculating marginal effects for the average individual.[4]

There are several instances where it might be important to adjust the marginal effect standard errors for either heteroskedasticity or clustering. For example, an over-dispersed Poisson regression will underestimate the usual standard errors. Therefore, one could apply a

[4]The average of the sample marginal effects is analogous.

White (1980) correction to the estimated variance-covariance matrix to account for this heteroskedasticity.[5] Another example applies in cases where the researcher is estimating models with clustered data. Ignoring the clustered nature of certain data will lead to an underestimate of the standard errors. The **mfx** package allows the user to correct for clustering using either a one-way or two-way correction in the variance-covariance matrix (Cameron *et al.* 2011) using the functionality offered in the **sandwich** package (Zeileis 2004, 2006).

Typically economists use marginal effects to display the output after estimating a GLM. However, other disciplines, particularly the medical sciences, use odds ratios (for example, in a logistic regression) or incidence rate ratios (for count regression models). Both ratios are derived from the fact that the underlining GLM is a log-linear model, so taking the exponent of the coefficient results in a multiplicative effect. Odds ratios are defined as the ratio of the probability of success and the probability of failure and therefore range between zero and infinity. Thus, an explanatory variable in a logistic regression with an odds ratio of 2 indicates that a one unit change in the explanatory variable increases the odds of the event by 2 to 1. Alternatively, an odds ratio of 1 would indicate that the regressor of interest does not influence the response. The incidence rate ratios used in Poisson and negative binomial count regression models are analogous to the aforementioned odds ratios. Once again they are multiplicative effects. For example, an incidence rate ratio of two will indicate that a one unit increase in the explanatory variable of interest doubles the underlying rate by which the count event is occurring. The **mfx** package accommodates odds ratios and incidence rate ratios in the applicable log-linear models.

3. Package details

The **mfx** software is an add-on package to the statistical software R, and is freely available from the Comprehensive R Archive Network (CRAN, http://CRAN.R-project.org/package=mfx). In addition to the base implementation of R, it requires the following packages: **MASS** (Venables and Ripley 2002), **sandwich** (Zeileis 2004, 2006), **lmtest** (Zeileis and Hothorn 2002), and **betareg** (Cribari-Neto and Zeileis 2010; Grün *et al.* 2012). Once R and the required packages have been installed, **mfx** can be loaded using the following code.

```
R> library("mfx")
```

Table 1 summarizes the GLM approaches that are compatible with the functions provided in **mfx**. The functions in **mfx** will first estimate the specified GLM, and after the GLM is fitted, the marginal effects (or odds/incidence rate ratios). These functions all return the requested output in the familiar coefficient table summary.

First, we look at the function that estimates a probit model, and returns its marginal effects as an output. The `probitmfx` function and it's arguments are shown below.

```
probitmfx(formula, data, atmean = TRUE, robust = FALSE, clustervar1 = NULL,
          clustervar2 = NULL, start = NULL, control = list())
```

The function is similar to either the `lm` or `glm` functions. The first argument: `formula` requires an object suitable for the formula class in R. The `formula` argument is identical

[5] The presence of heteroskedasticity in models with a binary response is best handled explicitly using **glmx** (Zeileis *et al.* 2013).

Regression Model	Response Type	Response Range	Marginal Effects	Odds Ratios	Incidence Rate Ratios
Probit	Binary	$\{0, 1\}$	✓	✗	✗
Logit	Binary	$\{0, 1\}$	✓	✓	✗
Poisson	Count	$[0, +\infty)$	✓	✗	✓
Negative Binomial	Count	$[0, +\infty)$	✓	✗	✓
Beta	Rate	$(0, 1)$	✓	✓	✗

Table 1: GLM approaches available in **mfx**.

to that required when estimating a probit model via the `glm` function, and is required by `probitmfx`. The next argument, `data` is for a data frame object. This argument is necessary so users should group their data into a data frame object prior to use. When `atmean = TRUE`, the resulting marginal effects will be for the average observation—as in Equation 8 and Equation 10—while if `atmean = FALSE`, the average of the sample marginal effects will be calculated—as in Equation 9 and Equation 11. In general, average of the sample marginal effects will take longer to be calculated. The `robust` argument allows the users to apply White's correction for the presence of heteroskedasticity in the calculation of marginal effect standard errors. Both of the `clustervar1` and `clustervar2` arguments are reserved for the names of the variables on which the user wishes to calculate either one or two-way clustered standard errors. These cluster names must correspond to a variable contained within the `data` object. The `start` and `control` arguments relate to identical arguments used to fit a model with `glm`.

Let's take a look at the output produced by `probitmfx` with a simple simulated example.

```
R> set.seed(12345)
R> n = 1000
R> x = rnorm(n)
R> y = ifelse(pnorm(1 + 0.5 * x + rnorm(n)) > 0.5, 1, 0)
R> data = data.frame(y, x)
R> (mod1 = probitmfx(formula = y ~ x, data = data))

Call:
probitmfx(formula = y ~ x, data = data)

Marginal Effects:
     dF/dx Std. Err.      z     P>|z|
x 0.121643  0.012165 9.9997 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

R> names(mod1)

[1] "mfxest" "fit"    "dcvar"  "call"

R> mod1$mfxest
```

```
      dF/dx Std. Err.        z        P>|z|
x 0.1216429 0.0121646 9.999745 1.527906e-23

R> mod1$fit

Call:  glm(formula = formula, family = binomial(link = "probit"), data = data,
    start = start, control = control, x = T)

Coefficients:
(Intercept)            x
     0.9911       0.5102

Degrees of Freedom: 999 Total (i.e. Null);  998 Residual
Null Deviance:            951.8
Residual Deviance: 848.8         AIC: 852.8

R> mod1$dcvar

character(0)

R> mod1$call

probitmfx(formula = y ~ x, data = data)
```

Calling the `probitmfx` object returns a `printCoefmat` object similar to that produced when `summary(glm(...))` is used for a GLM. However, instead of the model coefficients, the `probitmfx` produces the marginal effects: `dF/dx`. The `probitmfx` object contains four objects. The first, `mfxest`, is a table of the marginal effects, their standard errors, a z-test statistic (testing if the marginal effect is equal to zero) and the corresponding p-value associated with the z-test representing a two-tailed test. The name `fit` refers to the stored `glm` object—in this case a probit model. Note that using `summary(probitmfx$fit)` reports uncorrected standard errors, not ones that have been adjusted using the `robust`, `clustervar1`, and `clustervar2` arguments in `probitmfx`. A notifier that signifies for which variables a discrete change marginal effects is captured with `dcvar`. Finally, `call` is the matched call object.

The **mfx** package also contains the following other functions: `betamfx`, `betaor`, `logitmfx`, `logitor`, `negbinirr`, `negbinmfx`, `poissonirr`, `poissonmfx`. Each of these functions is self explanatory, with `mfx`, `or`, or `irr` indicating marginal effects, odds ratios, or incidence rate ratios respectively. The logit and Poisson models are fit with the `glm` function available as a base package in R. The negative binomial is fit using the `glm.nb` function in **MASS**. Finally, the beta regression is fit via the **betareg** package. Both `betamfx` and `betaor` functions use a logit link for the mean function, so it is feasible to calculate both marginal effects and odds ratios for these models.

4. Example analysis

This section illustrates how a simple analysis can be performed in **mfx**. For this analysis, I use the Swiss labor market participation data `SwissLabor` that is included in the **AER** (Kleiber

and Zeileis 2008) package. The code below, clears the workspace and loads the relevant data frame.

```
R> rm(list = ls())
R> library("AER")
Loading required package: car
Loading required package: lmtest
Loading required package: zoo

Attaching package: 'zoo'

The following objects are masked from 'package:base':

    as.Date, as.Date.numeric

Loading required package: sandwich
Loading required package: survival
Loading required package: splines
R> data("SwissLabor")
R> head(SwissLabor)
  participation   income age education youngkids oldkids foreign
1            no 10.78750 3.0         8         1       1      no
2           yes 10.52425 4.5         8         0       1      no
3            no 10.96858 4.6         9         0       0      no
4            no 11.10500 3.1        11         2       0      no
5            no 11.10847 4.4        12         0       2      no
6           yes 11.02825 4.2        12         0       1      no
```

For this example, we want to model labor force participation as a function of covariates. In the next step we load the **mfx** and estimate the baseline probit model returning the marginal effects as an output.

```
R> library("mfx")
Loading required package: MASS
Loading required package: betareg
Loading required package: Formula
R> (mod1 = probitmfx(participation ~ income + age + education +
+                    youngkids + oldkids + foreign,
+                   data = SwissLabor))
Call:
probitmfx(formula = participation ~ income + age + education +
    youngkids + oldkids + foreign, data = SwissLabor)

Marginal Effects:
               dF/dx  Std. Err.       z     P>|z|
income     -0.1992314  0.0485655 -4.1023 4.090e-05 ***
age        -0.1232260  0.0214953 -5.7327 9.885e-09 ***
```

```
education   0.0080889  0.0069485  1.1641    0.2444
youngkids  -0.3110035  0.0409640 -7.5921 3.147e-14 ***
oldkids    -0.0053438  0.0177937 -0.3003    0.7639
foreignyes  0.3112408  0.0429215  7.2514 4.125e-13 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

dF/dx is for discrete change for the following variables:

[1] "foreignyes"
```

Creates a `probitmfx` object called `mod1`. The printed object shows the function call, a table of the marginal effects, and a notification that the `foreign` variable represents a discrete change and the marginal effects for this variable have been calculated accordingly. The marginal effect values appear sensible. For example, a one-unit change in the number of young children associated with an observation reduces the probability of labor force participation by $\approx 31\%$. We must keep in mind that these marginal effects refer to the average individual. However, we can calculate the average of the sample marginal effects.

```
R> (mod2 = probitmfx(participation ~ income + age + education +
+                      youngkids + oldkids + foreign,
+                    data = SwissLabor, atmean = FALSE))
Call:
probitmfx(formula = participation ~ income + age + education +
    youngkids + oldkids + foreign, data = SwissLabor, atmean = FALSE)

Marginal Effects:
                dF/dx  Std. Err.       z     P>|z|
income     -0.1729131  0.0409901 -4.2184 2.460e-05 ***
age        -0.1069480  0.0176141 -6.0717 1.265e-09 ***
education   0.0070203  0.0060165  1.1668    0.2433
youngkids  -0.2699201  0.0321591 -8.3933 < 2.2e-16 ***
oldkids    -0.0046379  0.0154409 -0.3004    0.7639
foreignyes  0.2856119  0.0397184  7.1909 6.436e-13 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

dF/dx is for discrete change for the following variables:

[1] "foreignyes"
```

Which leads to comparable results. Given that the response variable is binary, we can also calculate the odds ratios obtained after fitting a logit regression. The code below demonstrates this.

```
R> (mod3 = logitor(participation ~ income + age + education +
+                    youngkids + oldkids + foreign,
```

```
+                 data = SwissLabor))
Call:
logitor(formula = participation ~ income + age + education +
    youngkids + oldkids + foreign, data = SwissLabor)

Odds Ratio:
           OddsRatio Std. Err.       z     P>|z|
income      0.442621  0.090959 -3.9661 7.305e-05 ***
age         0.600298  0.054338 -5.6379 1.721e-08 ***
education   1.032237  0.029972  1.0927    0.2745
youngkids   0.264286  0.047616 -7.3859 1.514e-13 ***
oldkids     0.978254  0.072162 -0.2980    0.7657
foreignyes  3.707675  0.740637  6.5600 5.382e-11 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The output for the odds ratios is as we would expect. The negative marginal effects have odds ratios below one, and the positive marginal effects, above one. The next **mfx** feature that is worth highlighting is the ability of the functions in the package to compute clustered standard errors. This functionality is displayed in the example below.

```
R> SwissLabor$id = 1:dim(SwissLabor)[1]
R> SwissLabor3 = rbind(SwissLabor, SwissLabor, SwissLabor)
R> (mod4 = probitmfx(participation ~ income + age + education +
+                     youngkids + oldkids + foreign,
+                   data = SwissLabor3))
Call:
probitmfx(formula = participation ~ income + age + education +
    youngkids + oldkids + foreign, data = SwissLabor3)

Marginal Effects:
                dF/dx  Std. Err.         z     P>|z|
income     -0.1992314  0.0280393  -7.1054 1.199e-12 ***
age        -0.1232260  0.0124103  -9.9293 < 2.2e-16 ***
education   0.0080889  0.0040117   2.0163   0.04377 *
youngkids  -0.3110035  0.0236506 -13.1499 < 2.2e-16 ***
oldkids    -0.0053438  0.0102732  -0.5202   0.60294
foreignyes  0.3112408  0.0247808  12.5598 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

dF/dx is for discrete change for the following variables:

[1] "foreignyes"
R> (mod5 = probitmfx(participation ~ income + age + education +
+                     youngkids + oldkids + foreign,
+                   data = SwissLabor3, clustervar1 = "id"))
```

```
Call:
probitmfx(formula = participation ~ income + age + education +
    youngkids + oldkids + foreign, data = SwissLabor3, clustervar1 = "id")

Marginal Effects:
               dF/dx  Std. Err.       z     P>|z|
income     -0.1992314  0.0453073 -4.3973 1.096e-05 ***
age        -0.1232260  0.0210013 -5.8675 4.423e-09 ***
education   0.0080889  0.0069325  1.1668    0.2433
youngkids  -0.3110035  0.0467831 -6.6478 2.976e-11 ***
oldkids    -0.0053438  0.0174540 -0.3062    0.7595
foreignyes  0.3112408  0.0437153  7.1197 1.081e-12 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

dF/dx is for discrete change for the following variables:

[1] "foreignyes"
```

In this example the data frame is duplicated twice and added to the existing data frame. Replicating the first `probitmfx` command on the enlarged data frame results in much lower standard errors. However, this can be corrected if we cluster the standard errors on the observation unit. As we can see this "fixes" the problem created by duplicating the data frame.

The final part of this example analysis illustrates how the **mfx** objects estimated in the above can be used to create a table with LaTeX. This involves coercing the **mfx** objects so that they are compatible with the **texreg** package. Since the **mfx** objects return the fitted `glm` object, we can use this as an input in the `texreg` function and override the model coefficients with the estimated marginal effect/odds ratios. Since the estimated `glm` models contain intercept values (and this will nearly always be the case) we take care to create some an additional value in our marginal effect/odds ratios and then remove this output using the `omit.coef` argument. Finally, some aesthetic alterations are made to the `texreg` object before saving the object. Table 2 shows what the table looks like when compiled in LaTeX.

```
R> library("texreg")
Version:  1.29.6
Date:     2013-09-27
Author:   Philip Leifeld (University of Konstanz)
R> mods = list(mod1$fit, mod2$fit, mod3$fit, mod4$fit, mod5$fit)
R> coefs = list(c(0, mod1$mfxest[, 1]), c(0, mod2$mfxest[, 1]),
+              c(0, mod3$oddsratio[, 1]), c(0, mod4$mfxest[, 1]),
+              c(0, mod5$mfxest[, 1]))
R> ses = list(c(0, mod1$mfxest[, 2]), c(0, mod2$mfxest[, 2]),
+            c(0, mod3$oddsratio[, 2]), c(0, mod4$mfxest[, 2]),
+            c(0, mod5$mfxest[, 2]))
R> pvals = list(c(0, mod1$mfxest[, 4]), c(0,mod2$mfxest [, 4]),
+              c(0, mod3$oddsratio[, 4]), c(0,mod4$mfxest [, 4]),
```

```
+              c(0, mod5$mfxest[, 4]))
>
R> tr1 = texreg(mods,
+              override.coef = coefs,
+              override.se = ses,
+              override.pval = pvals,
+              omit.coef = "(Intercept)",
+              caption.above = TRUE,
+              caption = "Models Explaining Labor Participation. Marginal Effects
                          and Odds Ratio Example",
+              dcolumn = TRUE,
+              custom.note = "\%stars.",
+              custom.model.names = c("(1)","(2)","(3)","(4)","(5)"),
+              return.string = TRUE)
. . . output omitted . . .
R> tr1 = unlist(strsplit(as.character(tr1), "\n"))
R> tr1 = c(tr1[1:6],
+         "\\\\[-1.8ex]\\hline", "\\hline \\\\[-1.8ex]",
+         " & \\multicolumn{1}{c}{Probit MFX}
+           & \\multicolumn{1}{c}{Probit MFX}
+           & \\multicolumn{1}{c}{Logit OR}
+           & \\multicolumn{1}{c}{Probit MFX}
+           & \\multicolumn{1}{c}{Probit MFX} \\\\",
+         tr1[8:length(tr1)])
> tr1[c(11, 24)] = "\\hline \\\\[-1.8ex]"
> tr1[31:33] = gsub("textsuperscript", "\\\\textsuperscript", tr1[31:33])
R> write(tr1, "table.tex")
```

5. Summary

This article introduces the **mfx** package for R. The package hosts a number of useful functions that should be of interest to those who conduct empirical research. Similarities between the functions provided in **mfx** and the well-known `glm` function mean that using **mfx** should be trivial for existing R users. There are a number of areas upon which the package could be improved. One such area would be to extend the number of models available. Examples of models that could be added include: ordered probit, multinomial logit, heteroskedastic probit, and instrumental variables probit. Another area for future expansion would be to improve the manner in which **mfx** handles nonlinear and interaction terms. For example, the current version of **mfx** calculates the marginal effect for each regressor separately, even if the same variable is included twice—albeit in two different forms, e.g., as a linear value and it's squared term. In instances like this, it may be preferable to have one marginal effect for each unique regressor and therefore **mfx** users should exercise caution before interpreting such values.

References

	Probit MFX (1)	Probit MFX (2)	Logit OR (3)	Probit MFX (4)	Probit MFX (5)
income	-0.20^{***}	-0.17^{***}	0.44^{***}	-0.20^{***}	-0.20^{***}
	(0.05)	(0.04)	(0.09)	(0.03)	(0.05)
age	-0.12^{***}	-0.11^{***}	0.60^{***}	-0.12^{***}	-0.12^{***}
	(0.02)	(0.02)	(0.05)	(0.01)	(0.02)
education	0.01	0.01	1.03	0.01^{*}	0.01
	(0.01)	(0.01)	(0.03)	(0.00)	(0.01)
youngkids	-0.31^{***}	-0.27^{***}	0.26^{***}	-0.31^{***}	-0.31^{***}
	(0.04)	(0.03)	(0.05)	(0.02)	(0.05)
oldkids	-0.01	0.00	0.98	-0.01	-0.01
	(0.02)	(0.02)	(0.07)	(0.01)	(0.02)
foreignyes	0.31^{***}	0.29^{***}	3.71^{***}	0.31^{***}	0.31^{***}
	(0.04)	(0.04)	(0.74)	(0.02)	(0.04)
AIC	1066.98	1066.98	1066.80	3172.95	3172.95
BIC	1100.38	1100.38	1100.19	3214.03	3214.03
Log Likelihood	-526.49	-526.49	-526.40	-1579.47	-1579.47
Deviance	1052.98	1052.98	1052.80	3158.95	3158.95
Num. obs.	872	872	872	2616	2616

$^{***}p < 0.001$, $^{**}p < 0.01$, $^{*}p < 0.05$.

Table 2: Models explaining labor force participation, marginal effects and odds ratio example.

Cameron AC, Gelbach J, Miller D (2011). "Robust Inference with Multi-way Clustering." *Journal of Business and Economic Statistics*, **29**(2), 238–249.

Cribari-Neto F, Zeileis A (2010). "Beta Regression in R." *Journal of Statistical Software*, **34**(2), 1–24. URL http://www.jstatsoft.org/v34/i02/.

Fernihough A (2014). ***mfx**: Marginal Effects, Odds Ratios and Incidence Rate Ratios for GLMs*. R package version 1.1, URL http://cran.r-project.org/web/packages/mfx.

Fox J (2003). "Effect Displays in R for Generalised Linear Models." *Journal of Statistical Software*, **8**(15), 1–27. URL http://www.jstatsoft.org/v08/i15/.

Fox J, Weisberg S, Friendly M, Hong J, Andersen R, Firth D, Taylor S (2013). ***effects**: Effect Displays for Linear, Generalized Linear, Multinomial-Logit, Proportional-Odds Logit Models and Mixed-Effects Models*. R package version 2.3-0, URL http://cran.r-project.org/web/packages/effects.

Greene WH (2008). *Econometric Analysis*. 6th edition. Prentice Hall, New York.

Grün B, Kosmidis I, Zeileis A (2012). "Extended Beta Regression in R: Shaken, Stirred, Mixed, and Partitioned." *Journal of Statistical Software*, **48**(11), 1–25. URL http://www.jstatsoft.org/v48/i11/.

Kleiber C, Zeileis A (2008). *Applied Econometrics with R*. Springer-Verlag, New York.

Leifeld P (2013). "**texreg**: Conversion of Statistical Model Output in R to LaTeXand HTML Tables." *Journal of Statistical Software*, **55**(8), 1–24. URL http://www.jstatsoft.org/v55/i08/.

McCullagh P, Nelder JA (1989). *Generalized Linear Models*. 2nd edition. Chapman & Hall, London.

Sun C (2013). ***erer**: Empirical Research in Economics with R*. R package version 1.4, URL http://cran.r-project.org/web/packages/erer.

Venables WN, Ripley BD (2002). *Modern Applied Statistics with S*. 4th edition. Springer-Verlag, New York.

White H (1980). "A Heteroskedasticity-Consistent Covariance Matrix Estimator and a Direct Test for Heteroskedasticity." *Econometrica*, **48**(8), 817–838.

Wooldridge JM (2002). *Econometric Analysis of Cross Section and Panel Data*. MIT Press, Cambridge.

Zeileis A (2004). "Econometric Computing with HC and HAC Covariance Matrix Estimators." *Journal of Statistical Software*, **11**(10), 1–17. URL http://www.jstatsoft.org/v11/i10/.

Zeileis A (2006). "Object-Oriented Computation of Sandwich Estimators." *Journal of Statistical Software*, **16**(9), 1–16. URL http://www.jstatsoft.org/v16/i09/.

Zeileis A, Hothorn T (2002). "Diagnostic Checking in Regression Relationships." *R News*, **2**(3), 7–10. URL http://CRAN.R-project.org/doc/Rnews/.

Zeileis A, Koenker R, Doebler P (2013). ***glmx**: Generalized Linear Models Extended*. R package version 0.1-0, URL http://cran.r-project.org/web/packages/glmx.

Affiliation:

Alan Fernihough
Queen's University Management School
Queen's University Belfast
185 Stranmillis Road
Belfast
BT9 5EE, United Kingdom
E-mail: alan.fernihough@gmail.com