

# Package ‘marginme’

June 7, 2026

**Type** Package

**Title** Estimation of Relative Risks, Risk Differences, and Marginal Effects from Mixed Models Using Marginal Standardization

**Version** 0.1.2

**Author** Sam Watson [aut, cre]

**Description** Estimation of relative risks, risk differences, and partial effects from mixed model. Marginalisation over random effect terms is accomplished using Markov Chain Monte Carlo.

**Imports** methods, glmmrBase (>= 1.4.0)

**Suggests** glmmTMB (>= 1.1.10)

**License** GPL (>= 2)

**Encoding** UTF-8

**LazyData** true

**URL** <https://github.com/samuel-watson/marginme>

**BugReports** <https://github.com/samuel-watson/marginme/issues>

**RoxygenNote** 7.3.3

**Depends** R (>= 2.10)

**NeedsCompilation** no

**Maintainer** Sam Watson <S.I.Watson@bham.ac.uk>

**Repository** CRAN

**Date/Publication** 2026-06-07 05:50:12 UTC

## Contents

confint.margin . . . . .	2
glmm_marginal . . . . .	3
margin . . . . .	4
print.margin . . . . .	6
summary.margin . . . . .	7
trial_data . . . . .	7

---

confint.margin	<i>Confidence interval for marginal effect</i>
----------------	--

---

### Description

Confidence interval method for class "margin"

### Usage

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

### Arguments

object	An object of class "margin" resulting from a call to margin
parm	No effect.
level	the confidence level required
...	additional argument(s) for methods

### Details

Computes confidence intervals using a standard Wald test for the marginal effect. If argument 'df' is used, then a t-statistic is used to construct the interval with 'df' degrees of freedom, otherwise a z-statistic is used.

### Value

A named vector giving lower and upper confidence limits for the marginal effect. They will be labelled as (1-level)/2 and 1-(1-level)/2. An argument df can be provided to use a t-test to construct the confidence interval.

### Examples

```
## fit a model using glmmTMB
fit <- glmmTMB::glmmTMB(y ~ Treatment + x1 + x2 + x3 + x4 + (1|Cluster),
  data = trial_data, family = binomial(link="logit"), REML = TRUE)
## relative risk, average over random effects and fixed effects
m1 <- margin(fit,
  x = "Treatment",
  type = "ratio",
  average = c("x1", "x2", "x3", "x4"),
  re = "average")
confint(m1)
```

glmm\_marginal

*Marginal effect from a binomial-logit or Poisson-log GLMM***Description**

Marginal effect from a binomial-logit or Poisson-log GLMM

**Usage**

```
glmm_marginal(
  X,
  beta,
  M,
  family,
  x,
  at = character(0),
  atvals = numeric(0),
  atmeans = character(0),
  average = character(0),
  re_type = c("AtZero", "AtEstimated", "At", "Average"),
  type = c("DyDx", "Diff", "Ratio"),
  xvals = c(0, 1),
  Zu = NULL,
  zu_samples = NULL,
  has_intercept = TRUE
)
```

**Arguments**

X	n x P design matrix. Must have column names.
beta	length-P fitted coefficient vector.
M	P x P variance-covariance matrix of beta (caller chooses model-based / sandwich / KR etc.).
family	"binomial" (logit) or "poisson" (log).
x	Name (character) or index (integer) of the column whose marginal effect we want.
at	Character vector of column names to fix at user values.
atvals	Numeric vector matching 'at'.
atmeans	Character vector of column names to fix at column mean.
average	Character vector of column names to average over (their empirical distribution in X is preserved).
re_type	Random-effect treatment: "At" - fix Zu at user value(s) via 'Zu' "AtEstimated" - posterior-mean Zu = rowMeans(zu_samples) "AtZero" - Zu = 0 "Average" - integrate over MCMC draws in zu_samples

type	"DyDx", "Diff", or "Ratio".
xvals	Length-2 numeric. DyDx is evaluated at xvals[1]; Diff and Ratio compare xvals[1] vs xvals[2].
Zu	Used only when re_type = "At". Either a single scalar (when no covariates are averaged) or a length-n vector of pre-computed Z u values (when 'average' is non-empty or only x is named with re_type Average/AtEstimated).
zu_samples	n x iter matrix of MCMC draws of Zu. Required for re_type = "Average" or "AtEstimated".
has_intercept	TRUE if the first / an intercept column is in X and should not be counted among "named" variables. Used only for the 'at' + 'atmeans' + 'average' + 1 == P - int consistency check.

### Value

list with elements 'estimate', 'se', plus the input choices.

---

margin	<i>Estimates Relative Risks, Risk Differences, and Marginal Effects from Mixed Models</i>
--------	---

---

### Description

Calculates the marginal effect of variable  $x$ . There are several options for marginal effect and several types of conditioning or averaging. The type of marginal effect can be the derivative of the mean with respect to  $x$  ('dydx'), the expected difference  $E(y|x=a)-E(y|x=b)$  ('diff'), or the expected log ratio  $\log(E(y|x=a)/E(y|x=b))$  ('ratio'). Other fixed effect variables can be set at specific values ('at'), set at their mean values ('atmeans'), or averaged over ('average'). Averaging over a fixed effects variable here means using all observed values of the variable in the relevant calculation. The random effects can similarly be set at their estimated value ('re="estimated"'), set to zero ('re="zero"'), set to a specific value ('re="at"'), or averaged over ('re="average"'). The standard errors are calculated using the delta method with one of several options for the variance matrix of the fixed effect parameters. Several of the arguments require the names of the variables as given to the model object. Most variables are as specified in the formula, factor variables are specified as the name of the 'variable\_value', e.g. 't\_1'. To see the names of the stored parameters and data variables see the member function 'names()'.

### Usage

```
margin(
  fit,
  x,
  type,
  re,
  at = c(),
  atmeans = c(),
  average = c(),
```

```

xvals = c(1, 0),
atvals = c(),
revals = c(),
oim = FALSE,
sampling = 250
)

```

## Arguments

<code>fit</code>	Either a <code>lme4</code> , <code>glmmTMB</code> , or <code>glmmrBase</code> model fit.
<code>x</code>	String. Name of the variable to calculate the marginal effect for.
<code>type</code>	String. Either <code>'dydx'</code> for derivative, <code>'diff'</code> for difference, or <code>'ratio'</code> for log ratio. See description.
<code>re</code>	String. Either <code>'estimated'</code> to condition on estimated values, <code>'zero'</code> to set to zero, <code>'at'</code> to provide specific values, or <code>'average'</code> to average over the random effects.
<code>at</code>	Optional. A vector of strings naming the fixed effects for which a specified value is given.
<code>atmeans</code>	Optional. A vector of strings naming the fixed effects that will be set at their mean value.
<code>average</code>	Optional. A vector of strings naming the fixed effects which will be averaged over.
<code>xvals</code>	Optional. A vector specifying the values of <code>'a'</code> and <code>'b'</code> for <code>'diff'</code> and <code>'ratio'</code> . The default is (1,0).
<code>atvals</code>	Optional. A vector specifying the values of fixed effects specified in <code>'at'</code> (in the same order).
<code>revals</code>	Optional. If <code>'re="at"'</code> then this argument provides a vector of values for the random effects.
<code>oim</code>	Logical. If <code>TRUE</code> use the observed information matrix, otherwise use the expected information matrix for standard error and related calculations.
<code>sampling</code>	Integer. Number of MCMC samples to use.

## Value

A named vector with elements `'margin'` specifying the point estimate and `'se'` giving the standard error.

## Examples

```

## fit a model using glmmTMB
fit <- glmmTMB::glmmTMB(y ~ Treatment + x1 + x2 + x3 + x4 + (1|Cluster),
  data = trial_data, family = binomial(link="logit"),REML = TRUE)
## relative risk, average over random effects and fixed effects
m1 <- margin(fit,
  x = "Treatment",
  type = "ratio",
  average = c("x1", "x2", "x3", "x4"),

```

```

      re = "average")
summary(m1)
## stata default for margins command is to set random effects to zero
m2 <- margin(fit,
  x = "Treatment",
  type = "ratio",
  average = c("x1", "x2", "x3", "x4"),
  re = "zero")
summary(m2)
## finally estimate a risk difference, with random effects at zero and fixed effects
## at mean values
m3 <- margin(fit,
  x = "Treatment",
  type = "diff",
  atmeans = c("x1", "x2", "x3", "x4"),
  re = "zero")
summary(m3)

```

---

print.margin

*Prints the marginal output*


---

### Description

Print method for class "margin"

### Usage

```
## S3 method for class 'margin'
print(x, ...)
```

### Arguments

x                    An object of class "margin" resulting from a call to margin  
 ...                  Further arguments passed from other methods

### Value

No return, called for effects

### Examples

```
## fit a model using glmmTMB
fit <- glmmTMB::glmmTMB(y ~ Treatment + x1 + x2 + x3 + x4 + (1|Cluster),
  data = trial_data, family = binomial(link="logit"), REML = TRUE)
## relative risk, average over random effects and fixed effects
margin(fit,
  x = "Treatment",
  type = "ratio",
  average = c("x1", "x2", "x3", "x4"),
  re = "average")

```

---

summary.margin	<i>Summarises the marginal output</i>
----------------	---------------------------------------

---

**Description**

Summary method for "'margin'" class

**Usage**

```
## S3 method for class 'margin'  
summary(object, ...)
```

**Arguments**

object	An object of class "'margin'" resulting from a call to margin
...	Further arguments passed from other methods

**Value**

No return, called for effects

**Examples**

```
## fit a model using glmmTMB  
fit <- glmmTMB::glmmTMB(y ~ Treatment + x1 + x2 + x3 + x4 + (1|Cluster),  
  data = trial_data, family = binomial(link="logit"),REML = TRUE)  
## relative risk, average over random effects and fixed effects  
m1 <- margin(fit,  
  x = "Treatment",  
  type = "ratio",  
  average = c("x1", "x2", "x3", "x4"),  
  re = "average")  
summary(m1)
```

---

trial_data	<i>Simulated trial data</i>
------------	-----------------------------

---

**Description**

Simulated trial data used to demonstrate the estimation of relative risk from an adjusted mixed logistic regression model. See [margin](#).

# Index

`confint.margin`, 2

`glmm_marginal`, 3

`margin`, 4, 7

`print.margin`, 6

`summary.margin`, 7

`trial_data`, 7