

# Package ‘modmarg’

June 5, 2017

**Title** Calculating Marginal Effects and Levels with Errors

**Version** 0.5.0

**Description** Calculate predicted levels and marginal effects from 'glm' objects, using the delta method to calculate standard errors. This is an R-based version of the 'margins' command from Stata.

**URL** <https://github.com/anniejw6/modmarg>

**BugReports** <https://github.com/anniejw6/modmarg/issues>

**Depends** R (>= 2.10.0)

**License** GPL-2

**Encoding** UTF-8

**LazyData** true

**Suggests** knitr, rmarkdown, testthat

**VignetteBuilder** knitr

**RoxygenNote** 5.0.1

**NeedsCompilation** no

**Author** Alex Gold [aut],  
Nat Olin [ctb],  
Annie Wang [aut, cre]

**Maintainer** Annie Wang <[anniejw6@gmail.com](mailto:anniejw6@gmail.com)>

**Repository** CRAN

**Date/Publication** 2017-06-05 17:43:58 UTC

## R topics documented:

cvcov	2
marg	2
margex	4

<b>Index</b>	<b>6</b>
--------------	----------

---

`vcov`*Clustered variance-covariance matrices and T statistic d.o.f.*

---

**Description**

Variance-covariance matrices with robust clustered standard errors and degrees-of-freedom for T statistics, for tests and examples specifying `vcov` (d.o.f. defined as  $g - 1$ , where  $g$  is the number of clusters). Generated with `margex` data in this package.

**Usage**`vcov`**Format**

A list of three lists, from an OLS model, logit model, and OLS with a polynomial interaction with missing data, each containing

**clust** 3-by-3 variance-covariance matrix

**dof** integer, degrees of freedom for the T statistic

**Details**

See `data-raw/make_cluster_vcov.R` for details.

**Source**

[http://cameron.econ.ucdavis.edu/research/Cameron\\_Miller\\_JHR\\_2015\\_February.pdf](http://cameron.econ.ucdavis.edu/research/Cameron_Miller_JHR_2015_February.pdf)

---

`marg`*Estimating predictive margins on a model*

---

**Description**

This function estimates the predictive effects and levels for variables within a model using the delta method.

**Usage**

```
marg(mod, var_interest, type = "levels", vcov_mat = NULL, dof = NULL,
      at = NULL, base_rn = 1, at_var_interest = NULL, data = mod$data,
      cofint = 0.95)
```

**Arguments**

mod	model object, currently only support those of class <code>glm</code>
var_interest	name of the variable of interest, must correspond to a covariate in the model
type	either 'levels' (predicted outcomes) or 'effects' <i>dydx</i> , defaults to 'levels'
vcov_mat	the variance-covariance matrix, defaults to NULL in which case <code>vcov(model)</code> is used.
dof	integer, the degrees of freedom used for the T statistic in an OLS model. Defaults to NULL in which case <code>mod\$df.residual</code> is used.
at	list, should be in the format of <code>list('var_name' = c(values))</code> , defaults to NULL. This calculates the margins of the variable at these particular variables. If all values are needed, suggested syntax is <code>at = list('var' = unique(df\$var))</code> .
base_rn	numeric, if <code>type == 'effects'</code> , the base level (taken as the index of one of the ordered unique values in <code>var_interest</code> ). if <code>type == 'levels'</code> , this parameter is ignored. Defaults to 1.
at_var_interest	vector, if <code>type == 'levels'</code> , the values for the variable of interest at which levels should be calculated. If NULL, indicates all levels for a factor variable, defaults to NULL
data	data.frame that margins should run over, defaults to <code>mod\$data</code>
cofint	numeric, confidence interval (must be less than 1), defaults to 0.95

**Details**

The variable for the predictive margin is specified by `var_interest`. If margins are only needed at particular values of `var_interest`, `at_var_interest` should be used. If margins of `var_interest` are needed at across the levels of a *different* variable in the model, `at` should be used.

If higher-order polynomial terms (e.g.  $y = x + x^2$ ) are added using the R function `poly`, the `raw = TRUE` argument should be used to include the basic polynomial terms instead of orthogonal polynomial terms. If orthogonal polynomials are used, `marg` will fail when the user specifies `at` for a small set of values for the variable in question (e.g. `at = list(x = 10)`), since `poly` needs more data to calculate orthogonal polynomials (e.g. `poly(10, 2)` fails, but `poly(c(10, 8, 3), 2)` will run).

P values are calculated with T tests for gaussian families, and Z tests otherwise. If a new variance-covariance matrix is provided (e.g. for clustering standard errors), the degrees of freedom for the T test / p-value calculation may need to be specified using `dof`. To replicate Stata clustering `vce(cluster var_name)`, `dof` should be set to  $g - 1$ , where  $g$  is the number of unique levels of the clustering variable.

This function currently only supports `glm` objects. If you would like to use `lm` objects, consider running a `glm` with family `gaussian`.

**Value**

list of dataframes with predicted margins/effects, standard errors, p-values, and confidence interval bounds

**Examples**

```

data(mtcars)
mod <- glm(vs ~ as.factor(gear) + mpg, data = mtcars, family = 'binomial')

# Get the level of the outcome variable at different values of `gear`
marg(mod, var_interest = 'gear', type = 'levels')

# Get the effect of `gear` on the outcome value, holding values of `mpg`
# constant
marg(mod, var_interest = 'gear', type = 'effects',
      at = list(mpg = c(15, 21)))

data(margex)
mod <- glm(outcome ~ as.factor(treatment) + distance,
           data = margex, family = 'binomial')
# Get the level of the outcome variable at different values of `treatment`
marg(mod, var_interest = 'treatment', type = 'levels', at = NULL)
# Get the effect of `treatment` on the outcome variable
marg(mod, var_interest = 'treatment', type = 'effects', at = NULL)
# Get the level of the outcome variable at different values of `distance`
marg(mod, var_interest = 'distance', type = 'levels',
      at = NULL, at_var_interest = c(10, 20, 30))

# Using a custom variance-covariance matrix for clustered standard errors
# (also requires custom degrees of freedom for T statistic with OLS model),
# clustering on the "arm" variable

data(margex)
data(cvcov)
?cvcov
v <- cvcov$ols$clust
d <- cvcov$ols$stata_dof
mod <- glm(outcome ~ treatment + distance,
           data = margex, family = 'binomial')
marg(mod, var_interest = 'treatment', type = 'levels',
      vcov_mat = v, dof = d)

```

---

margex

*Artificial data for margins*


---

**Description**

A fictitious dataset outcome, treatment, and demographic variables for 3000 observations.

**Usage**

margex

**Format**

A data frame with 3000 rows and 11 variables:

**y** numeric

**outcome** integer, 0 or 1

**sex** character: "female" or "male"

**group** integer

**age** integer

**distance** numeric

**ycn** numeric

**yc** numeric, 0 or 1

**treatment** integer

**agegroup** character: "20-29", "30-39", or "40+"

**arm** integer

**Source**

<http://www.stata-press.com/data/r14/margex.dta>

# Index

## \*Topic **datasets**

cvcov, [2](#)

margex, [4](#)

cvcov, [2](#)

glm, [3](#)

marg, [2](#)

margex, [4](#)

poly, [3](#)