Package ‘marginaleffects’

October 18, 2021

Title Marginal Effects, Marginal Means, Predictions, and Contrasts

Version 0.2.0

Description Compute, summarize, and plot marginal effects, adjusted predictions, contrasts, and marginal means for a wide variety of models.

License GPL (>= 3)

Copyright inst/COPYRIGHTS

LazyData true

Encoding UTF-8

URL https://vincentarelbundock.github.io/marginaleffects/

https://github.com/vincentarelbundock/marginaleffects

BugReports https://github.com/vincentarelbundock/marginaleffects/issues

RoxygenNote 7.1.2

VignetteBuilder knitr

Config/testthat/edition 3

Config/testthat/parallel true

Depends R (>= 3.5.0)

Imports checkmate, generics, insight (>= 0.14.5), methods, numDeriv

Suggests bench, broom, dplyr, emmeans, ggplot2, ggbeeswarm, haven, kableExtra, knitr, magrittr, margins, modelsummary, patchwork, rlang, rmarkdown, testthat (>= 3.0.0), tidyverse, vdg, AER, betareg, bife, brglm2, crch, estimatr, fixest, gam, geepack, glmx, ivreg, lme4, MASS, mgcv, nlme, nnet, ordinal, plm, pscl, quantreg, rms, robustbase, sandwich, speedglm, survey, survival, truncreg, covr, spelling

Collate 'complete_levels.R' 'counterfactual_typical.R' 'get_coef.R'

'get_dydx.R' 'get_dydx_and_se.R' 'get_dydx_se.R'

'get_group_names.R' 'get_predict.R' 'get_vcov.R'

'marginaleffects.R' 'marginalmeans.R' 'mean_or_mode.R'

'set_coef.R' 'methods_MASS.R' 'methods_betareg.R'

'methods_bife.R' 'methods_crch.R' 'methods_glmx.R'
counterfactual

Generate "counterfactual" datasets for use in `marginaleffects`'s `newdata` argument

Description

Generate "counterfactual" datasets for use in `marginaleffects`'s `newdata` argument

Usage

```r
counterfactual(..., model = NULL, newdata = NULL)
```
Arguments

... named arguments with vectors of values for the variables to construct (see Examples below.)
model Model object
newdata data.frame (one and only one of the model and newdata arguments must be true).

Details

If counterfactual is used in a marginaleffects or predictions call as the newdata argument, users do not need to specify the model or newdata argument. The data is extracted automatically from the model.

If users supply a model, the data used to fit that model is retrieved using the insight::get_data function.

If users supply a model, the data used to fit that model is retrieved using the insight::get_data function, and then replicated with different values of the variables in the at list.

Value

A data.frame where each row of the original data is repeated multiple times for each of the values of the variables in the at list. See example below.

Examples

# All rows are repeated twice, with different values of `hp`
cd <- counterfactual(newdata = mtcars, hp = c(100, 110))
cd[cd$rowid %in% 1:3,]

# We get the same result by feeding a model instead of a data.frame
mod <- lm(mpg ~ hp + wt, mtcars)
cd <- counterfactual(model = mod, hp = c(100, 110))
cd[cd$rowid %in% 1:3,]

# Use in `marginaleffects` to compute "Counterfactual Average Marginal Effects"
marginaleffects(mod, newdata = counterfactual(hp = c(100, 110)))

marginaleffects Marginal effects using numerical derivatives

Description

A "marginal effect" is the partial derivative of the regression equation with respect to a variable in the model. This function uses automatic differentiation to compute marginal effects for a vast array of models, including non-linear models with transformations (e.g., polynomials). The list of supported models and of models whose numerical results have been validated against external software (Stata or margins) is held in the supported_models dataset which accompanies this package.
Usage

```
marginaleffects(
  model,
  newdata = NULL,
  variables = NULL,
  vcov = TRUE,
  type = "response",
  ...
)
```

Arguments

- **model**: Model object.
- **newdata**: A dataset over which to compute marginal effects. `NULL` uses the original data used to fit the model.
- **variables**: Variables to consider (character vector). `NULL` calculates marginal effects for all terms in the model object.
- **vcov**: Matrix or boolean
  - `FALSE`: does not compute unit-level standard errors.
  - `TRUE`: computes unit-level standard errors using the default `vcov(model)` variance-covariance matrix.
  - Named square matrix: computes standard errors with a user-supplied variance-covariance matrix. This matrix must be square and have dimensions equal to the number of coefficients in `get_coef(model)`.
- **type**: Type(s) of prediction as string or vector. This can differ based on the model type, but will typically be a string such as: "response", "link", "probs", or "zero".
- **...**: Additional arguments are pushed forward to `predict()`.

Value

A data.frame of marginal effect estimates with one row per observation per marginal effect.

Examples

```
mod <- glm(am ~ hp * wt, data = mtcars, family = binomial)
mfx <- marginaleffects(mod)
summary(mfx)
tidy(mfx)
head(mfx)
plot(mfx)

# typical marginal effects
marginaleffects(mod,
  newdata = typical(hp = c(100, 110)))

# counterfactual average marginal effects
marginaleffects(mod,
  newdata =
```
marginalmeans

```r
newdata = counterfactual(hp = c(100, 110))

# heteroskedasticity robust standard errors
marginaleffects(mod, vcov = sandwich::vcovHC(mod))
```

### marginalmeans

**Marginal Means**

#### Description

Compute estimated marginal means for specified factors.

#### Usage

```r
marginalmeans(
  model,
  variables = NULL,
  variables_grid = NULL,
  vcov = insight::get_varcov(model),
  type = "response"
)
```

#### Arguments

- **model**: Model object
- **variables**: Categorical predictors over which to compute marginal means (character vector). **NULL** calculates marginal means for all logical, character, or factor variables in the dataset used to fit model.
- **variables_grid**: Categorical predictors used to construct the prediction grid over which adjusted predictions are averaged (character vector). **NULL** creates a grid with all combinations of all categorical predictors. This grid can be very large when there are many variables and many response levels, so it is advisable to select a limited number of variables in the **variables** and **variables_grid** arguments.
- **vcov**: Matrix or boolean
  - **FALSE**: does not compute unit-level standard errors.
  - **TRUE**: computes unit-level standard errors using the default `vcov(model)` variance-covariance matrix.
  - Named square matrix: computes standard errors with a user-supplied variance-covariance matrix. This matrix must be square and have dimensions equal to the number of coefficients in `get_coef(model)`.
- **type**: Type(s) of prediction as string or vector. This can differ based on the model type, but will typically be a string such as: "response", "link", "probs", or "zero".
Details

This function begins by calling the `predictions` function to obtain a grid of predictors, and adjusted predictions for each cell. The grid includes all combinations of the categorical variables listed in the `variables` and `variables_grid` arguments, or all combinations of the categorical variables used to fit the model if `variables_grid` is `NULL`. In the prediction grid, numeric variables are held at their means.

After constructing the grid and filling the grid with adjusted predictions, `marginalmeans` computes marginal means for the variables listed in the `variables` argument, by average across all categories in the grid.

`marginalmeans` can only compute standard errors for linear models, or for predictions on the link scale, that is, with the `type` argument set to "link".

The `marginaleffects` website compares the output of this function to the popular `emmeans` package, which provides similar but more advanced functionality: https://vincentarelbundock.github.io/marginaleffects/

Value

Data frame of marginal means with one row per variable-value combination.

Examples

```r
library(marginaleffects)

# Convert numeric variables to categorical before fitting the model
data <- mtcars
data$cyl <- as.factor(data$cyl)
data$am <- as.logical(data$am)
mod <- lm(mpg ~ hp + cyl + am, data = data)

# Compute and summarize marginal means
mm <- marginalmeans(mod)
summary(mm)
```

```
plot.marginaleffects  Point-range plot of average marginal effects
```

Description

Uses the `ggplot2` package to draw a point-range plot of the average marginal effects computed by `tidy`.

Usage

```r
## S3 method for class 'marginaleffects'
plot(x, conf.int = TRUE, conf.level = 0.95, ...)
```
**Arguments**

- **x**  
  An object produced by the `marginaleffects` function.

- **conf.int**  
  Logical indicating whether or not to include a confidence interval.

- **conf.level**  
  The confidence level to use for the confidence interval if `conf.int=TRUE`. Must be strictly greater than 0 and less than 1. Defaults to 0.95, which corresponds to a 95 percent confidence interval.

  ...  
  Additional arguments are pushed forward to `predict()`.

**Details**

The `tidy` function calculates average marginal effects by taking the mean of all the unit-level marginal effects computed by the `marginaleffects` function.

**Value**

A `ggplot2` object

**Examples**

```r
mod <- glm(am ~ hp + wt, data = mtcars)
mfx <- marginaleffects(mod)
plot(mfx)
```

---

**plot_cap**  
*Conditional adjusted predictions plot*

**Description**

This function plots the adjusted predictions of the outcome (y-axis) against values of one or more predictors.

**Usage**

```r
plot_cap(
  model,
  condition,
  type = "response",
  conf.int = TRUE,
  conf.level = 0.95,
  draw = TRUE
)
```
plot_cme

Conditional marginal effects plot

Description

In models where two continuous variables are interacted, the marginal effect of one variable is conditional on the value of the other variable. This function draws a plot of the marginal effect of the effect variable for different values of the condition variable.

Usage

```r
plot_cme(
  model,
  effect,
  condition,
  type = "response",
  conf.int = TRUE,
  conf.level = 0.95,
  draw = TRUE
)
```
### predictions

**Adjusted Predictions**

**Description**

Compute model-adjusted predictions (fitted values) for a "grid" of regressor values.

**Usage**

```r
predictions(
  model,  
  variables = NULL,  
  newdata = NULL,  
  conf.level = 0.95,  
  type = "response",  
  ...  
)
```
Arguments

model Model object
variables Character vector. Compute Adjusted Predictions for combinations of each of these variables. Factor levels are considered at each of their levels. Numeric variables are considered at Tukey’s Five-Number Summaries. NULL uses the original data used to fit the model.
newdata A dataset over which to compute adjusted predictions. NULL uses the original data used to fit the model.
conf.level The confidence level to use for the confidence interval. No interval is computed if conf.int=NULL. Must be strictly greater than 0 and less than 1. Defaults to 0.95, which corresponds to a 95 percent confidence interval.
type Type(s) of prediction as string or vector. This can differ based on the model type, but will typically be a string such as: "response", "link", "probs", or "zero".
... Additional arguments are pushed forward to predict().

Value

A data.frame with a predicted column with predictions.

Description

Print a marginaleffects summary

Usage

## S3 method for class 'marginaleffects.summary'
print(x, digits = max(3L,getOption("digits") - 3L), ...)

Arguments

x an object produced by the marginaleffects function.
digits the number of significant digits to use when printing.
... Additional arguments are pushed forward to predict().

Value

Printed summary of a marginalmeans object
print.marginalmeans.summary

Print a marginalmeans summary

Description

Print a marginalmeans summary

Usage

```r
## S3 method for class 'marginalmeans.summary'
print(x, digits = max(3L, getOption("digits") - 3L), ...)
```

Arguments

- `x` an object produced by the `marginaleffects` function.
- `digits` the number of significant digits to use when printing.
- `...` Additional arguments are pushed forward to `predict()`.

Value

Printed summary of a marginalmeans object

summary.marginaleffects

Summarize a marginaleffects object

Description

Summarize a marginaleffects object

Usage

```r
## S3 method for class 'marginaleffects'
summary(object, conf.level = 0.95, ...)
```

Arguments

- `object` An object produced by the `marginaleffects` function
- `conf.level` The confidence level to use for the confidence interval if `conf.int=TRUE`. Must be strictly greater than 0 and less than 1. Defaults to 0.95, which corresponds to a 95 percent confidence interval.
- `...` Additional arguments are pushed forward to `predict()`.

Value

Data frame of summary statistics for an object produced by the `marginaleffects` function
**summary.marginalmeans**  
*Summarize a marginalmeans object*

### Description

Summarize a marginalmeans object

### Usage

```r
## S3 method for class 'marginalmeans'
summary(object, conf.level = 0.95, ...)
```

### Arguments

- **object**  
  An object produced by the `marginalmeans` function

- **conf.level**  
  The confidence level to use for the confidence interval if conf.int=TRUE. Must be strictly greater than 0 and less than 1. Defaults to 0.95, which corresponds to a 95 percent confidence interval.

- **...**  
  Additional arguments are pushed forward to `predict()`

### Value

Data frame of summary statistics for an object produced by the `marginalmeans` function

---

**supported_models**  
*List of model types supported by the marginaleffects package*

### Description

A list of model types supported by the `marginaleffects` package

### Usage

```r
supported_models
```

### Format

- **data.frame**

  - **Package**  
    R package which produces the model object.

  - **Function**  
    R function which produces the model object.

  - **dydx**  
    Can `marginaleffects` retrieve marginal effects?

  - **se**  
    Can `marginaleffects` retrieve standard errors?

  - **stata_dydx**  
    Has the validity of numerical results for marginal effects been checked against Stata’s `margins` command?
Has the validity of numerical results for standard errors been checked against Stata’s margins command?

Has the validity of numerical results for marginal effects been checked against the margins package for R?

Has the validity of numerical results for standard errors been checked against the margins package for R?

---

**tidy.marginaleffects**  
*Tidy a marginaleffects object*

**Description**

Tidy a marginaleffects object

**Usage**

```r
## S3 method for class 'marginaleffects'
tidy(x, conf.int = TRUE, conf.level = 0.95, ...)
```

**Arguments**

- `x`: An object produced by the `marginaleffects` function.
- `conf.int`: Logical indicating whether or not to include a confidence interval.
- `conf.level`: The confidence level to use for the confidence interval if `conf.int` = TRUE. Must be strictly greater than 0 and less than 1. Defaults to 0.95, which corresponds to a 95 percent confidence interval.
- `...`: Additional arguments are pushed forward to `predict()`.

**Details**

The `tidy` function calculates average marginal effects by taking the mean of all the unit-level marginal effects computed by the `marginaleffects` function.

**Value**

A "tidy" data.frame of summary statistics which conforms to the broom package specification.

**Examples**

```r
mod <- lm(mpg ~ hp * wt + factor(gear), data = mtcars)
mfx <- marginaleffects(mod)
tidy(mfx)
```
**tidy.marginalmeans**  
*Tidy a marginalmeans object*

**Description**

Tidy a marginalmeans object

**Usage**

```r
## S3 method for class 'marginalmeans'
tidy(x, conf.int = TRUE, conf.level = 0.95, ...)
```

**Arguments**

- `x` An object produced by the `marginalmeans` function.
- `conf.int` Logical indicating whether or not to include a confidence interval.
- `conf.level` The confidence level to use for the confidence interval if `conf.int=TRUE`. Must be strictly greater than 0 and less than 1. Defaults to 0.95, which corresponds to a 95 percent confidence interval.
- `...` Additional arguments are pushed forward to `predict()`.

**Value**

A "tidy" data.frame of summary statistics which conforms to the broom package specification.

**typical**  
*Generate "typical" datasets for use in marginaleffects's newdata argument*

**Description**

Generate "typical" datasets for use in marginaleffects's newdata argument

**Usage**

```r
typical(..., model = NULL, newdata = NULL)
```

**Arguments**

- `...` named arguments with vectors of values for the typical variables to construct (see Examples below.) The typical data will include combinations of unique values from these vectors
- `model` Model object
- `newdata` data.frame (one and only one of the model and newdata arguments must be true).
**Details**

If `typical` is used in a `marginaleffects` or `predictions` call as the `newdata` argument, users do not need to specify the `model` or `newdata` argument. The data is extracted automatically from the model.

If users supply a model, the data used to fit that model is retrieved using the `insight::get_data` function.

**Value**

A `data.frame` in which each row corresponds to one combination of the named predictors supplied by the user via the `...` dots. Variables which are not explicitly defined are held at their mean or mode.

**Examples**

```r
# The output only has 2 rows, and all the variables except 'hp' are at their
# mean or mode.
typical(newdata = mtcars, hp = c(100, 110))

# We get the same result by feeding a model instead of a data.frame
mod <- lm(mpg ~ hp, mtcars)
typical(model = mod, hp = c(100, 110))

# Use in `marginaleffects` to compute "Typical Marginal Effects"
marginaleffects(mod, newdata = typical(hp = c(100, 110)))
```
Index

* datasets
  supported_models, 12

counterfactual, 2

marginaleffects, 3
marginalmeans, 5

plot.marginaleffects, 6
plot_cap, 7
plot_cme, 8
predictions, 9
print.marginaleffects.summary, 10
print.marginalmeans.summary, 11

summary.marginaleffects, 11
summary.marginalmeans, 12
supported_models, 12

tidy.marginaleffects, 13
tidy.marginalmeans, 14
typical, 14