Package 'svymargins'

July 6, 2026

Title Predictive Margins for Survey Analyses

Version 0.1.0

Author Tommi Härkänen [aut, cre]

Maintainer Tommi Härkänen <tommi.harkanen@thl.fi>

Depends R (>= 4.2.0)

Description Predictive margins like in the 'Stata' procedure 'margins'. This package is based on the survey::svypredmean() function. Supported regression analyses are based on survey::svyglm() and svyVGAM::svy_vglm() functions (for multinomial logistic regression models).

License GPL (>= 3)

Encoding UTF-8

Imports plyr, dplyr, stringr, survey, tidyselect, VGAM

Suggests knitr, rmarkdown, svyVGAM

VignetteBuilder knitr

Config/roxygen2/version 8.0.0

NeedsCompilation no

Repository CRAN

Date/Publication 2026-07-06 14:30:02 UTC

Contents

asymm_ci *Helper function to convert symmetric probability scale confidence intervals into asymmetric*

Description

Helper function to convert symmetric probability scale confidence intervals into asymmetric

Usage

```
asymm_ci(x, ci.level = 0.95)
```

Arguments

x	Matrix with two columns as returned by confint function.
ci.level	Default 0.95 for 95% confidence intervals. The transformation is based on the logit transformation and the delta method.

Value

Matrix with two columns as returned by confint function.

Examples

```
library(survey)
library(dplyr)
library(svyVGAM)
n <- 1000
# Generate data:
set.seed(1234)
d <- data.frame(sex=factor(sample(c("M", "F"), n, replace=TRUE)),
                education=factor(sample(c("low", "middle", "high"), n, replace=TRUE)))
d <- d |>
  mutate(age=runif(n, 0, 40) + as.numeric(education) * 20,
         pr1=1,
         pr2=exp(-3 + 0.5 * as.numeric(education) + 0.05 * age),
         pr3=exp(1 + -0.5 * as.numeric(education) + 0.02 * age)) |>
  rowwise() |>
  mutate(y=which(rmultinom(1, 1, c(pr1, pr2, pr3))[,1] == 1)) |>
  ungroup() |>
  mutate(y=factor(y, levels=1:3, labels=LETTERS[1:3]))
# Create survey design:
my.svy <- svydesign(~ 1, weights=~ 1, data=d)
# Run regression analysis:
res <- svy_vglm(y ~ education + age + sex, family=multinomial(refLevel=1), design=my.svy)
# Define margins as a named list:
target.l <- list(null=list(),
                 educ=list("education"),
                 age=list(age=seq(40,70,10)),
```

```
                  educ_age=list("education", age=seq(40,70,10)))
# Calculate predictive margins:
marg <- svymargins(res, groupfactor=target.l, y.lev=2)
confint(marg)
confint(marg) |> asymm_ci()
```

---

scenarios2group *Specify constraints for selected covariates for predicted margins*

Description

Specify constraints for selected covariates for predicted margins

Usage

```
scenarios2group(scenarios, dat)
```

Arguments

scenarios	A (named) list of scenarios based on factor and/or continuous covariates.
dat	A dataframe containing the covariates.
	Details
	All combinations of the specified covariates are returned as a dataframe. If the list contains more than one element, then the dataframes based on the elements are returned, and binded together. if scenarios is a named list, then the returned dataframe has column group_id containing the list names.

Value

A dataframe containing the combinations of the specified covariates.

Examples

```
n <- 1000
d <- data.frame(sex=factor(sample(c("M", "F"), n, replace=TRUE)),
                age=runif(20, 100, n),
                education=factor(sample(c("low", "middle", "high"), n, replace=TRUE)),
                y=rnorm(n))
target.l <- list(null=list(), educ=list("education"), age=list(age=seq(40,70,10)),
                 educ_age=list("education", age=seq(40,70,10)))
scenarios2group(target.l, d)
```

svymargins *Generic function for* svymargins

Description

Generic function for svymargins

Usage

```
svymargins(adjustmodel, ...)
```

Arguments

adjustmodel	A regression analysis result object.
...	Other arguments See other methods for different regression analyses.

Value

As svypredmeans, an object of class svystat with the predictive marginal means and their covariance matrix. Additional attribute: groups, which contains dataframe with the groupfactor variable values, and the estimated margins and standard errors.

svymargins.default *The default method for* svymargins

Description

The default method for svymargins

Usage

```
## Default S3 method:
svymargins(adjustmodel, ...)
```

Arguments

adjustmodel	A regression analysis result object.
...	Other arguments Executed for regression analyses, which have not been implemented.

Value

An error message.

See Also

svymargins.svyglm, svymargins.svy_vglm

`svymargins.svyglm` *Mimics the margins procedure of Stata for GLM estimated with survey::svyglm*

Description

Based on the survey::svypredmeans function, but is more flexible. No restrictions with interactions. Implements also subsets with the `subs` argument.

Usage

```
## S3 method for class 'svyglm'
svymargins(adjustmodel, ..., groupfactor, y.lev = NULL, subs = NULL)
```

Arguments

`adjustmodel`	Result of regression analysis object from the survey::svyglm function.
`...`	Currently unused.
`groupfactor`	Specification of the margins: Character vector of factor variable names or a list. See details and examples.
`y.lev`	Not currently used in `svyglm` models.
`subs`	Character string specifying a subset, e.g. `"age < 50 & year == 2015"`.
	Details
	`groupfactor` is a list specifying several margins specifications. The list can contain one or more lists. A list can contain the names of factor covariates (as character strings) and/or continuous covariates as named vectors containing the desired values at which to calculate the margins.

Value

As `svypredmeans`, an object of class svystat with the predictive marginal means and their covariance matrix. Additional attribute: `groups`, which contains dataframe with the `groupfactor` variable values, and the estimated margins and standard errors. The result of `svymargins` is named based on the `groupfactor`, but if there are duplicates, then the result is unnamed.

Examples

```
library(survey)
library(dplyr)
n <- 1000
# Generate data:
set.seed(1234)
d <- data.frame(sex=factor(sample(c("M", "F"), n, replace=TRUE)),
```

```
                education=factor(sample(c("low", "middle", "high"), n, replace=TRUE)))
d <- d |> mutate(age=runif(n, 0, 40) + as.numeric(education) * 20,
                 y=rnorm(n, sd=5) + as.numeric(education) + 0.05 * age)
# Create survey design:
my.svy <- svydesign(~ 1, weights=~ 1, data=d)
# Run regression analysis:
res <- svyglm(y ~ education + age + sex, design=my.svy)
# Define margins as a named list:
target.l <- list(null=list(),
                 educ=list("education"),
                 age=list(age=seq(40,70,10)),
                 educ_age=list("education", age=seq(40,70,10)))
# Calculate predictive margins:
svymargins(res, groupfactor=target.l)
# Get the output table containing the covariate information from "groups" attribute:
attr(svymargins(res, groupfactor=target.l), "groups")
```

`svymargins.svy_vglm` *Predictive margins for multinomial logistic regression model*

Description

Predictive margins for multinomial logistic regression model

Usage

```
## S3 method for class 'svy_vglm'
svymargins(adjustmodel, ..., groupfactor, y.lev = NULL, subs = NULL)
```

Arguments

`adjustmodel`	Result of regression analysis object from the survey::svyglm function.
`...`	Currently unused.
`groupfactor`	Specification of the margins: Character vector of factor variable names or a list. See details and examples.
`y.lev`	Output category (name or number of level) for which margins are calculated.
`subs`	Character string specifying a subset, e.g. `"age < 50 & year == 2015"`.
	Details
	`groupfactor` is a list specifying several margins specifications. The list can contain one or more lists. A list can contain the names of factor covariates (as character strings) and/or continuous covariates as named vectors containing the desired values at which to calculate the margins.

Value

As `svypredmeans`, an object of class svystat with the predictive marginal means and their covariance matrix. Additional attribute: `groups`, which contains dataframe with the `groupfactor` variable values, and the estimated margins and standard errors. The result of `svymargins` is named based on the `groupfactor`, but if there are duplicates, then the result is unnamed.

Examples

```
library(survey)
library(dplyr)
library(svyVGAM)
n <- 1000
# Generate data:
set.seed(1234)
d <- data.frame(sex=factor(sample(c("M", "F"), n, replace=TRUE)),
                education=factor(sample(c("low", "middle", "high"), n, replace=TRUE)))
d <- d |>
  mutate(age=runif(n, 0, 40) + as.numeric(education) * 20,
         pr1=1,
         pr2=exp(-3 + 0.5 * as.numeric(education) + 0.05 * age),
         pr3=exp(1 + -0.5 * as.numeric(education) + 0.02 * age)) |>
  rowwise() |>
  mutate(y=which(rmultinom(1, 1, c(pr1, pr2, pr3))[,1] == 1)) |>
  ungroup() |>
  mutate(y=factor(y, levels=1:3, labels=LETTERS[1:3]))
# Create survey design:
my.svy <- svydesign(~ 1, weights=~ 1, data=d)
# Run regression analysis:
res <- svy_vglm(y ~ education + age + sex, family=multinomial(refLevel=1), design=my.svy)
# Define margins as a named list:
target.l <- list(null=list(),
                 educ=list("education"),
                 age=list(age=seq(40,70,10)),
                 educ_age=list("education", age=seq(40,70,10)))
# Calculate predictive margins:
svymargins(res, groupfactor=target.l, y.lev=2)
# Get the output table containing the covariate information from "groups" attribute:
attr(svymargins(res, groupfactor=target.l, y.lev="B"), "groups")
```

Index