Title

stata.com

margins — Marginal means, predictive margins, and marginal effects

Description

Margins are statistics calculated from predictions of a previously fit model at fixed values of some covariates and averaging or otherwise integrating over the remaining covariates.

The `margins` command estimates margins of responses for specified values of covariates and presents the results as a table.

Capabilities include estimated marginal means, least-squares means, average and conditional marginal and partial effects (which may be reported as derivatives or as elasticities), average and conditional adjusted predictions, and predictive margins.

Quick start

Estimated marginal means (least-squares means)

Estimated marginal mean of `y` for each level of `a` after `anova y a##b`

```
margins a, asbalanced
```

Estimated marginal mean of `y` for each level of the interaction of `a` and `b` after `anova y a##b##c`

```
margins a#b, asbalanced
```

Estimated marginal means of `y1`, `y2`, and `y3` for each level of `a` after `manova y1 y2 y3 = a##b`

```
margins a, asbalanced
```

Adjusted means and adjusted predictions

Adjusted mean of `y` for each level of `a` when `x` is at its mean after `regress y i.a x`

```
margins a, atmeans
```

Same as above, but set `x` to 10 rather than to its mean

```
margins a, at(x=10)
```

Same as above, and also report adjusted means for $x = 20$, $x = 30$, and $x = 40$

```
margins a, at(x=(10(10)40))
```

Adjusted predicted probability of $y = 1$ for each level of `a` when `x` is at its mean after `logit y a##c.x`

```
margins a, atmeans
```

Adjusted predicted probability for each level of the interaction of `a` and `b`, holding `x` at 25, after `logit y a##b##c.x`

```
margins a#b, at(x=25)
```

Adjusted prediction for each level of `a` when $x = 25$ and $b = 1$

```
margins a, at(x=25 b=1)
```

Predictive margins and potential-outcome means

Overall predictive margin, the average predicted probability of y = 1, after `logit y a##b x1 x2`

```
margins
```

Predictive margins (potential-outcome means) for each level of `a`

```
margins a
```

Predictive margins for `a`, when `x1` is set to 10, 20, 30, and 40

```
margins a, at(x1=(10(10)40))
```

Predictive margins for levels of the interaction of `a` and `b`

```
margins a#b
```

Predictive margins for `a` for all combinations of `x1` = 10, 20, 30 and `x2` = 50, 100, 150

```
margins a, at(x1=(10(10)30) x2=(50(50)150))
```

Predictive margins for `a`, first for `x1` = 10, 20, 30 with `x2` at its observed values, then for `x2` = 50, 100, 150 with `x1` at its observed values

```
margins a, at(x1=(10(10)30))  at(x2=(50(50)150))
```

Predictive margins for `a` after `svy:logit y a##b x1 x2`

```
margins a, vce(unconditional)
```

Average predicted probabilities of y = 1, y = 2, ... after `mlogit y x1 x2 i.a`

```
margins
```

Predictive margins for each level of `a` for each outcome of `y`

```
margins a
```

Average marginal effects and average partial effects

Average marginal effect of `x1` on the predicted probability of y = 1 after `probit y c.x1##c.x2##a` with continuous `x1` and `x2` and binary `a`

```
margins, dydx(x1)
```

Average marginal effect (average partial effect) of binary `a`

```
margins, dydx(a)
```

Average marginal effect of `x1` when `x2` is set to 10, 20, 30, and 40

```
margins, dydx(x1) at(x2=(10(10)40))
```

Average marginal effect of `x1` when `a` is set to 0 and then to 1

```
margins a, dydx(x1)
```

Average marginal effect of each variable in the model

```
margins, dydx(*)
```

Average marginal effect of all variables on the truncated expected value of `y`, `e(0,.)`, after `tobit y x1 x2 x3, ll(0)`

```
margins, dydx(*) predict(e(0,.))
```

Same as above, and report marginal effects for censored expected value of `y`, `ystar(0,.)`, and for the linear prediction, `xb`

```
margins, dydx(*) predict(e(0,.)) predict(ystar(0,.)) predict(xb)
```

Conditional marginal effects and conditional partial effects

Marginal effect of `x1` on the predicted probability of `y` = 1, setting all variables to their means, after `probit y c.x1##c.x2##a` with continuous `x1` and `x2` and binary `a`

```
margins, dydx(x1) atmeans
```

Marginal effect (partial effect) of `a` when all variables are set to their means

```
margins, dydx(a) atmeans
```

Marginal effect of `x1` when `a` = 0 and `x1` and `x2` are set to their means

```
margins, dydx(x1) at(a=0 (mean) x1 x2)
```

Same as above

```
margins, dydx(x1) at(a=0) atmeans
```

Marginal effect of `x1` when for all possible combinations of `a` = 0, 1, `x1` = 50, 100, and `x2` = 10, 20, 30, 40

```
margins a, dydx(x1) at(x1=(50 100) x2=(10(10)40))
```

Marginal effects of `x1`, `x2`, and `a` with all variables set to their means

```
margins, dydx(*) atmeans
```

Menu

Statistics > Postestimation

Syntax

`margins` [*marginlist*] [*if*] [*in*] [*weight*] [`,` *response_options* *options*]

where *marginlist* is a list of factor variables or interactions that appear in the current estimation results. The variables may be typed with or without the `i.` prefix, and you may use any factor-variable syntax:

```
. margins i.sex i.group i.sex#i.group
. margins sex group sex#i.group
. margins sex##group
```

response_options	Description
Main	
`predict(`*pred_opt*`)`	estimate margins for `predict,` *pred_opt*
`expression(`*pnl_exp*`)`	estimate margins for *pnl_exp*
`dydx(`*varlist*`)`	estimate marginal effect of variables in *varlist*
`eyex(`*varlist*`)`	estimate elasticities of variables in *varlist*
`dyex(`*varlist*`)`	estimate semielasticity—$d(\widehat{y})/d(\ln x)$
`eydx(`*varlist*`)`	estimate semielasticity—$d(\ln\widehat{y})/d(x)$
`continuous`	treat factor-level indicators as continuous

options	Description
Main	
`grand`	add the overall margin; default if no *marginlist*
At	
`at(`*atspec*`)`	estimate margins at specified values of covariates
`atmeans`	estimate margins at the means of covariates
`asbalanced`	treat all factor variables as balanced
if/in/over	
`over(`*varlist*`)`	estimate margins at unique values of *varlist*
`subpop(`*subspec*`)`	estimate margins for subpopulation
Within	
`within(`*varlist*`)`	estimate margins at unique values of the nesting factors in *varlist*
Contrast	
contrast_options	any options documented in [R] **margins, contrast**
Pairwise comparisons	
pwcompare_options	any options documented in [R] **margins, pwcompare**
SE	
`vce(delta)`	estimate SEs using delta method; the default
`vce(unconditional)`	estimate SEs allowing for sampling of covariates
`nose`	do not estimate SEs
Advanced	
`noweights`	ignore weights specified in estimation
`noesample`	do not restrict `margins` to the estimation sample
`emptycells(`*empspec*`)`	treatment of empty cells for balanced factors
`estimtolerance(`*tol*`)`	specify numerical tolerance used to determine estimable functions; default is `estimtolerance(1e-5)`
`noestimcheck`	suppress estimability checks
`force`	estimate margins despite potential problems
`chainrule`	use the chain rule when computing derivatives
`nochainrule`	do not use the chain rule
Reporting	
`level(#)`	set confidence level; default is `level(95)`
`mcompare(`*method*`)`	adjust for multiple comparisons; default is `mcompare(noadjust)`
`noatlegend`	suppress legend of fixed covariate values
`post`	post margins and their VCE as estimation results
display_options	control columns and column formats, row spacing, line width and factor-variable labeling
`df(#)`	use t distribution with # degrees of freedom for computing p-values and confidence intervals

method	Description
noadjust	do not adjust for multiple comparisons; the default
bonferroni [adjustall]	Bonferroni's method; adjust across all terms
sidak [adjustall]	Šidák's method; adjust across all terms
scheffe	Scheffé's method

Time-series operators are allowed if they were used in the estimation.

See at() under *Options* for a description of *atspec*.

collect is allowed; see [U] **11.1.10 Prefix commands**.

fweights, aweights, iweights, and pweights are allowed; see [U] **11.1.6 weight**.

df(#) does not appear in the dialog box.

Options

Warning: The option descriptions are brief and use jargon. Skip to Remarks and examples if you are reading about margins *for the first time.*

Main

predict(*pred_opt*) and expression(*pnl_exp*) are mutually exclusive; they specify the response. If neither is specified, the response will be the default prediction that would be produced by predict after the underlying estimation command. Some estimation commands, such as mlogit, document a different default prediction for margins than for predict.

predict(*pred_opt*) specifies the option(s) to be specified with the predict command to produce the variable that will be used as the response. After estimation by logistic, you could specify predict(xb) to obtain linear predictions rather than the predict command's default, the probabilities.

Multiple predict() options can be specified to compute margins of multiple predictions simultaneously.

expression(*pnl_exp*) specifies the response as an expression. See [R] **predictnl** for a full description of *pnl_exp*. After estimation by logistic, you might specify expression(exp(predict(xb))) to use relative odds rather than probabilities as the response. For examples, see *Example 12: Margins of a specified expression*.

dydx(*varlist*), eyex(*varlist*), dyex(*varlist*), and eydx(*varlist*) request that margins report derivatives of the response with respect to *varlist* rather than on the response itself. eyex(), dyex(), and eydx() report derivatives as elasticities; see *Expressing derivatives as elasticities*.

continuous is relevant only when one of dydx() or eydx() is also specified. It specifies that the levels of factor variables be treated as continuous; see *Derivatives versus discrete differences*. This option is implied if there is a single-level factor variable specified in dydx() or eydx().

grand specifies that the overall margin be reported. grand is assumed when *marginlist* is empty.

At

at(*atspec*) specifies values for covariates to be treated as fixed.

at(age=20) fixes covariate age to the value specified. at() may be used to fix continuous or factor covariates.

at(age=20 sex=1) simultaneously fixes covariates age and sex at the values specified.

`at(age=(20 30 40 50))` fixes age first at 20, then at 30, ... `margins` produces separate results for each specified value.

`at(age=(20(10)50))` does the same as `at(age=(20 30 40 50))`; that is, you may specify a numlist.

`at((mean) age (median) distance)` fixes the covariates at the summary statistics specified. `at((p25) _all)` fixes all covariates at their 25th percentile values. See *Syntax of at()* for the full list of summary-statistic modifiers.

`at((mean) _all (median) x x2=1.2 z=(1 2 3))` is processed from general to specific, with settings for named covariates overriding general settings specified via `_all`. Thus, all covariates are fixed at their means except for `x` (fixed at its median), `x2` (fixed at 1.2), and `z` (fixed first at 1, then at 2, and finally at 3).

`at((means) _all (asobserved) x2)` is a convenient way to set all covariates except `x2` to the mean.

Multiple `at()` options can be specified, and each will produce a different set of margins.

See *Syntax of at()* for more information.

`atmeans` specifies that covariates be fixed at their means and is shorthand for `at((mean) _all)`. `atmeans` differs from `at((mean) _all)` in that `atmeans` will affect subsequent `at()` options. For instance,

```
. margins ..., atmeans  at((p25) x)  at((p75) x)
```

produces two sets of margins with both sets evaluated at the means of all covariates except `x`.

`asbalanced` is shorthand for `at((asbalanced) _factor)` and specifies that factor covariates be evaluated as though there were an equal number of observations in each level; see *Obtaining margins as though the data were balanced*. `asbalanced` differs from `at((asbalanced) _factor)` in that `asbalanced` will affect subsequent `at()` options in the same way as `atmeans` does.

if/in/over

`over(`*varlist*`)` specifies that separate sets of margins be estimated for the groups defined by *varlist*. The variables in *varlist* must contain nonnegative integer (or missing) values. The variables need not be covariates in your model. When `over()` is combined with the `vce(unconditional)` option, each group is treated as a subpopulation; see [SVY] **Subpopulation estimation**.

`subpop(`[*varname*] [*if*]`)` is intended for use with the `vce(unconditional)` option. It specifies that margins be estimated for the single subpopulation identified by the indicator variable or by the `if` expression or by both. Zero or missing indicates that the observation be excluded; nonzero or nonmissing, that it be included. See [SVY] **Subpopulation estimation** for why `subpop()` is preferred to `if` expressions and `in` ranges when also using `vce(unconditional)`. If `subpop()` is used without `vce(unconditional)`, it is treated merely as an additional `if` qualifier.

Within

`within(`*varlist*`)` allows for nested designs. *varlist* contains the nesting variable(s) over which margins are to be estimated. See *Obtaining margins with nested designs*. As with `over(`*varlist*`)`, when `within(`*varlist*`)` is combined with `vce(unconditional)`, each level of the variables in *varlist* is treated as a subpopulation.

Contrast

contrast_options are any of the options documented in [R] **margins, contrast**.

Pairwise comparisons

pwcompare_options are any of the options documented in [R] **margins, pwcompare**.

SE

`vce(delta)` and `vce(unconditional)` specify how the VCE and, correspondingly, standard errors are calculated.

`vce(delta)` is the default. The delta method is applied to the formula for the response and the VCE of the estimation command. This method assumes that values of the covariates used to calculate the response are given or, if all covariates are not fixed using `at()`, that the data are given.

`vce(unconditional)` specifies that the covariates that are not fixed be treated in a way that accounts for their having been sampled. The VCE is estimated using the linearization method. This method allows for heteroskedasticity or other violations of distributional assumptions and allows for correlation among the observations in the same manner as `vce(robust)` and `vce(cluster ...)`, which may have been specified with the estimation command. This method also accounts for complex survey designs if the data are `svyset`. See *Obtaining margins with survey data and representative samples*. When you use complex survey data, this method requires that the linearized variance estimation method be used for the model. See [SVY] **svy postestimation** for an example of `margins` with replication-based methods.

`nose` suppresses calculation of the VCE and standard errors. See *Requirements for model specification* for an example of the use of this option.

Advanced

`noweights` specifies that any weights specified on the previous estimation command be ignored by `margins`. By default, `margins` uses the weights specified on the estimator to average responses and to compute summary statistics. If weights are specified on the `margins` command, they override previously specified weights, making it unnecessary to specify `noweights`. The `noweights` option is not allowed after `svy:` estimation when the `vce(unconditional)` option is specified.

For multilevel models, such as `meglm`, the default behavior is to construct a single weight value for each observation by multiplying the corresponding multilevel weights within the given observation.

`noesample` specifies that `margins` not restrict its computations to the estimation sample used by the previous estimation command. See *Example 15: Margins evaluated out of sample*.

With the default delta-method VCE, `noesample` margins may be estimated on samples other than the estimation sample; such results are valid under the assumption that the data used are treated as being given.

You can specify `noesample` and `vce(unconditional)` together, but if you do, you should be sure that the data in memory correspond to the original `e(sample)`. To show that you understand that, you must also specify the `force` option. Be aware that making the `vce(unconditional)` calculation on a sample different from the estimation sample would be equivalent to estimating the coefficients on one set of data and computing the scores used by the linearization on another set; see [P] **_robust**.

`emptycells(strict)` and `emptycells(reweight)` are relevant only when the `asbalanced` option is also specified. `emptycells()` specifies how empty cells are handled in interactions involving factor variables that are being treated as balanced; see *Obtaining margins as though the data were balanced*.

`emptycells(strict)` is the default; it specifies that margins involving empty cells be treated as not estimable.

emptycells(reweight) specifies that the effects of the observed cells be increased to accommodate any missing cells. This makes the margin estimable but changes its interpretation. emptycells(reweight) is implied when the within() option is specified.

estimtolerance(*tol*) specifies the numerical tolerance used to determine estimable functions. The default is estimtolerance(1e-5).

A linear combination of the model coefficients z is found to be not estimable if

$$\texttt{mreldif(}z\texttt{, }z \times H\texttt{)} > tol$$

where H is defined in *Methods and formulas*.

noestimcheck specifies that margins not check for estimability. By default, the requested margins are checked and those found not estimable are reported as such. Nonestimability is usually caused by empty cells. If noestimcheck is specified, estimates are computed in the usual way and reported even though the resulting estimates are manipulable, which is to say they can differ across equivalent models having different parameterizations. See *Estimability of margins*.

force instructs margins to proceed in some situations where it would otherwise issue an error message because of apparent violations of assumptions. Do not be casual about specifying force. You need to understand and fully evaluate the statistical issues. For an example of the use of force, see *Using margins after the estimates use command*.

chainrule and nochainrule specify whether margins uses the chain rule when numerically computing derivatives. You need not specify these options when using margins after any official Stata estimator; margins will choose the appropriate method automatically.

Specify nochainrule after estimation by a community-contributed command. We recommend using nochainrule, even though chainrule is usually safe and is always faster. nochainrule is safer because it makes no assumptions about how the parameters and covariates join to form the response.

nochainrule is implied when the expression() option is specified.

Reporting

level(#) specifies the confidence level, as a percentage, for confidence intervals. The default is level(95) or as set by set level; see [U] **20.8 Specifying the width of confidence intervals**.

mcompare(*method*) specifies the method for computing p-values and confidence intervals that account for multiple comparisons within a factor-variable term.

Most methods adjust the comparisonwise error rate, α_c, to achieve a prespecified experimentwise error rate, α_e.

mcompare(noadjust) is the default; it specifies no adjustment.

$$\alpha_c = \alpha_e$$

mcompare(bonferroni) adjusts the comparisonwise error rate based on the upper limit of the Bonferroni inequality

$$\alpha_e \leq m\alpha_c$$

where m is the number of comparisons within the term.

The adjusted comparisonwise error rate is

$$\alpha_c = \alpha_e/m$$

mcompare(sidak) adjusts the comparisonwise error rate based on the upper limit of the probability inequality

$$\alpha_e \leq 1 - (1 - \alpha_c)^m$$

where m is the number of comparisons within the term.

The adjusted comparisonwise error rate is

$$\alpha_c = 1 - (1 - \alpha_e)^{1/m}$$

This adjustment is exact when the m comparisons are independent.

`mcompare(scheffe)` controls the experimentwise error rate using the F or χ^2 distribution with degrees of freedom equal to the rank of the term.

`mcompare(`*method* `adjustall)` specifies that the multiple-comparison adjustments count all comparisons across all terms rather than performing multiple comparisons term by term. This leads to more conservative adjustments when multiple variables or terms are specified in *marginslist*. This option is compatible only with the `bonferroni` and `sidak` methods.

`noatlegend` specifies that the legend showing the fixed values of covariates be suppressed.

`post` causes `margins` to behave like a Stata estimation (e-class) command. `margins` posts the vector of estimated margins along with the estimated variance–covariance matrix to `e()`, so you can treat the estimated margins just as you would results from any other estimation command. For example, you could use `test` to perform simultaneous tests of hypotheses on the margins, or you could use `lincom` to create linear combinations. See *Example 10: Testing margins—contrasts of margins*.

display_options: `noci`, `nopvalues`, `vsquish`, `nofvlabel`, `fvwrap(#)`, `fvwrapon(`*style*`)`, `cformat(%`*fmt*`)`, `pformat(%`*fmt*`)`, `sformat(%`*fmt*`)`, and `nolstretch`.

`noci` suppresses confidence intervals from being reported in the coefficient table.

`nopvalues` suppresses p-values and their test statistics from being reported in the coefficient table.

`vsquish` specifies that the blank space separating factor-variable terms or time-series–operated variables from other variables in the model be suppressed.

`nofvlabel` displays factor-variable level values rather than attached value labels. This option overrides the `fvlabel` setting; see [R] **set showbaselevels**.

`fvwrap(#)` allows long value labels to wrap the first # lines in the coefficient table. This option overrides the `fvwrap` setting; see [R] **set showbaselevels**.

`fvwrapon(`*style*`)` specifies whether value labels that wrap will break at word boundaries or break based on available space.

`fvwrapon(word)`, the default, specifies that value labels break at word boundaries.

`fvwrapon(width)` specifies that value labels break based on available space.

This option overrides the `fvwrapon` setting; see [R] **set showbaselevels**.

`cformat(%`*fmt*`)` specifies how to format margins, standard errors, and confidence limits in the table of estimated margins.

`pformat(%`*fmt*`)` specifies how to format p-values in the table of estimated margins.

`sformat(%`*fmt*`)` specifies how to format test statistics in the table of estimated margins.

`nolstretch` specifies that the width of the table of estimated margins not be automatically widened to accommodate longer variable names. The default, `lstretch`, is to automatically widen the table of estimated margins up to the width of the Results window. Specifying `lstretch` or `nolstretch` overrides the setting given by `set lstretch`. If `set lstretch` has not been set, the default is `lstretch`. `nolstretch` is not shown in the dialog box.

The following option is available with `margins` but is not shown in the dialog box:

`df(#)` specifies that the t distribution with # degrees of freedom be used for computing p-values and confidence intervals. The default typically is to use the standard normal distribution. However, if the estimation command computes the residual degrees of freedom (`e(df_r)`) and `predict(xb)` is specified with `margins`, the default is to use the t distribution with `e(df_r)` degrees of freedom.

Remarks and examples

stata.com

Remarks are presented under the following headings:

Introduction

`margins` is a postestimation command, a command for use after you have fit a model using an estimation command such as `regress` or `logistic`, or using almost any other estimation command.

`margins` estimates and reports margins of responses and margins of derivatives of responses, also known as marginal effects. A margin is a statistic based on a fitted model in which some of or all the covariates are fixed. Marginal effects are changes in the response for change in a covariate, which can be reported as a derivative, elasticity, or semielasticity.

For a brief overview of `margins`, see Williams (2012).

Obtaining margins of responses

What we call margins of responses are also known as predictive margins, adjusted predictions, and recycled predictions. When applied to balanced data, margins of responses are also called estimated marginal means and least-squares means.

A margin is a statistic based on a fitted model calculated over a dataset in which some of or all the covariates are fixed at values different from what they really are. For instance, after a linear regression fit on males and females, the marginal mean (margin of mean) for males is the predicted mean of the dependent variable, where every observation is treated as if it represents a male; thus, those observations that in fact do represent males are included, as well as those observations that represent females. The marginal mean for females would be similarly obtained by treating all observations as if they represented females.

In making the calculation, sex is treated as male or female everywhere it appears in the model. The model might be

```
. regress y age bp i.sex sex#c.age sex#c.bp
```

and then, in making the marginal calculation of the mean for males and females, `margins` not only accounts for the direct effect of `i.sex` but also for the indirect effects of `sex#c.age` and `sex#c.bp`.

The response being margined can be any statistic produced by [R] **predict**, or any expression of those statistics.

Standard errors are obtained by the delta method, at least by default. The delta method assumes that the values at which the covariates are evaluated to obtain the marginal responses are fixed. When your sample represents a population, whether you are using `svy` or not (see [SVY] **svy**), you can specify `margins`' `vce(unconditional)` option and `margins` will produce standard errors that account for the sampling variability of the covariates. Some researchers reserve the term predictive margins to describe this.

The best way to understand `margins` is to see some examples. You can run the following examples yourself if you type

```
. use https://www.stata-press.com/data/r18/margex
(Artificial data for margins)
```

Example 1: A simple case after regress

```
. regress y i.sex i.group
  (output omitted)
. margins sex
Predictive margins                                      Number of obs = 3,000
Model VCE: OLS
Expression: Linear prediction, predict()
```

	Margin	Delta-method std. err.	t	P>|t|	[95% conf.	interval]
sex						
Male	60.56034	.5781782	104.74	0.000	59.42668	61.69401
Female	78.88236	.5772578	136.65	0.000	77.7505	80.01422

The numbers reported in the `Margin` column are average values of the linear prediction of `y`, as noted above the output table after `Expression:`. Based on a linear regression of `y` on `sex` and `group`, 60.6 would be the average value of the linear prediction of `y` if everyone in the data were treated as if they were male, and 78.9 would be the average value if everyone were treated as if they were female.

Example 2: A simple case after logistic

`margins` may be used after almost any estimation command.

```
. logistic outcome i.sex i.group
  (output omitted)
. margins sex
Predictive margins                                      Number of obs = 3,000
Model VCE: OIM
Expression: Pr(outcome), predict()
```

	Margin	Delta-method std. err.	z	P>|z|	[95% conf.	interval]
sex						
Male	.1286796	.0111424	11.55	0.000	.106841	.1505182
Female	.1905087	.0089719	21.23	0.000	.1729241	.2080933

The numbers reported in the `Margin` column are average predicted probabilities. Based on a logistic regression of `outcome` on `sex` and `group`, 0.13 would be the average probability of `outcome` if everyone in the data were treated as if they were male, and 0.19 would be the average probability if everyone were treated as if they were female.

`margins` reports average values after `regress` and average probabilities after `logistic`. By default, `margins` makes tables of whatever it is that `predict` (see [R] **predict**) predicts by default. Alternatively, `margins` can make tables of anything that `predict` can produce if you use `margins`' `predict()` option; see *Example 11: Margins of a specified prediction*.

Example 3: Average response versus response at average

In example 2, margins reported average probabilities of outcome for sex = 0 and sex = 1. If we instead wanted the predicted probabilities evaluated at the mean of the covariates, we would specify margins' atmeans option. We previously typed

```
. logistic outcome i.sex i.group
  (output omitted)
. margins sex
  (output omitted)
```

and now we type

```
. margins sex, atmeans
Adjusted predictions                                    Number of obs = 3,000
Model VCE: OIM

Expression: Pr(outcome), predict()
At: 0.sex   = .4993333 (mean)
    1.sex   = .5006667 (mean)
    1.group = .3996667 (mean)
    2.group = .3726667 (mean)
    3.group = .2276667 (mean)
```

	Margin	Delta-method std. err.	z	P>|z|	[95% conf.	interval]
sex						
Male	.0966105	.0089561	10.79	0.000	.0790569	.1141641
Female	.1508362	.0118064	12.78	0.000	.127696	.1739764

The prediction at the average of the covariates is different from the average of the predictions. The first is the expected probability of a person with average characteristics, a person who, in another problem, might be 3/4 married and have 1.2 children. The second is the average of the probability among actual persons in the data.

When you specify atmeans or any other at option, margins reports the values used for the covariates in the legend above the table. margins lists the values for all the covariates, including values it may not use, in the results that follow. In this example, margins reported means for sex even though those means were not used. They were not used because we asked for the margins of sex, so sex was fixed first at 0 and then at 1.

If you wish to suppress this legend, specify the nolegend option.

Example 4: Multiple margins from one command

More than one margin can be reported by just one margins command. You can type

```
. margins sex group
```

and doing that is equivalent in terms of the output to typing

```
. margins sex
. margins group
```

When multiple margins are requested on the same command, each is estimated separately. There is, however, a difference when you also specify margins' post option. Then, the variance–covariance matrix for all margins requested is posted, and that is what allows you to test, for example, equality of margins. Testing equality of margins is covered in *Example 10: Testing margins—contrasts of margins*.

In any case, below we request margins for sex and for group.

```
. margins sex group
Predictive margins                                      Number of obs = 3,000
Model VCE: OIM
Expression: Pr(outcome), predict()
```

	Margin	Delta-method std. err.	z	P>|z|	[95% conf.	interval]
sex						
Male	.1286796	.0111424	11.55	0.000	.106841	.1505182
Female	.1905087	.0089719	21.23	0.000	.1729241	.2080933
group						
1	.2826207	.0146234	19.33	0.000	.2539593	.311282
2	.1074814	.0094901	11.33	0.000	.0888812	.1260817
3	.0291065	.0073417	3.96	0.000	.0147169	.043496

Example 5: Margins with interaction terms

The estimation command on which margins bases its calculations may contain interaction terms, such as an interaction of sex and group:

```
. logistic outcome i.sex i.group sex#group
  (output omitted)
. margins sex group
Predictive margins                                      Number of obs = 3,000
Model VCE: OIM
Expression: Pr(outcome), predict()
```

	Margin	Delta-method std. err.	z	P>|z|	[95% conf.	interval]
sex						
Male	.1561738	.0132774	11.76	0.000	.1301506	.182197
Female	.1983749	.0101546	19.54	0.000	.1784723	.2182776
group						
1	.3211001	.0176403	18.20	0.000	.2865257	.3556744
2	.1152127	.0099854	11.54	0.000	.0956417	.1347838
3	.0265018	.0109802	2.41	0.016	.0049811	.0480226

We fit the model by typing logistic outcome i.sex i.group sex#group, but the meaning would have been the same had we typed logistic outcome sex##group.

As mentioned in example 4, the results for sex and the results for group are calculated independently, and we would have obtained the same results had we typed margins sex followed by margins group.

The margin for male (sex = 0) is 0.16. The probability 0.16 is the average probability if everyone in the data were treated as if sex = 0, including sex = 0 in the main effect and sex = 0 in the interaction of sex with group.

Had we specified margins sex, atmeans, we would have obtained not average probabilities but the probabilities evaluated at the average. Rather than obtaining 0.16, we would have obtained 0.10

for sex = 0. The 0.10 is calculated by taking the fitted model, plugging in sex = 0 everywhere, and plugging in the average value of the group indicator variables everywhere they are used. That is, rather than treating the group indicators as being (1, 0, 0), (0, 1, 0), or (0, 0, 1) depending on observation, the group indicators are treated as being (0.40, 0.37, 0.23), which are the average values of group = 1, group = 2, and group = 3.

Example 6: Margins with continuous variables

To the above example, we will add the continuous covariate age to the model and then rerun margins sex group.

```
. logistic outcome i.sex i.group sex#group age
  (output omitted)
. margins sex group
Predictive margins                                  Number of obs = 3,000
Model VCE: OIM
Expression: Pr(outcome), predict()
```

	Margin	Delta-method std. err.	z	P>|z|	[95% conf.	interval]
sex						
Male	.1600644	.0125653	12.74	0.000	.1354368	.184692
Female	.1966902	.0100043	19.66	0.000	.1770821	.2162983
group						
1	.2251302	.0123233	18.27	0.000	.200977	.2492834
2	.150603	.0116505	12.93	0.000	.1277685	.1734376
3	.0736157	.0337256	2.18	0.029	.0075147	.1397167

Compared with the results presented in example 5, results for sex change little, but results for groups 1 and 3 change markedly. The tables differ because now we are adjusting for the continuous covariate age, as well as for sex and group.

We will continue examining interactions in example 8. Because we have added a continuous variable, let's take a detour to explain how to obtain margins for continuous variables and to explain their interpretation.

Example 7: Margins of continuous variables

Continuing with our example of

```
. logistic outcome i.sex i.group sex#group age
```

let's examine the continuous covariate age.

You are not allowed to type `margins age`; doing that will produce an error:

```
. margins age
factor age not found in list of covariates
r(322);
```

The message "`age` not found in list of covariates" is `margins`' way of saying, "Yes, age might be in the model, but if it is, it is not included as a factor variable; it is in as a continuous variable." Sometimes, Stata is overly terse. `margins` might also say that because age is continuous there are an infinite number of values at which it could evaluate the margins. At what value(s) should age be fixed? `margins` requires more guidance with continuous covariates. We can provide that guidance by using the `at()` option and typing

```
. margins, at(age=40)
```

To understand why that yields the desired result, let us tell you that if you were to type

```
. margins
```

`margins` would report the overall margin—the margin that holds nothing constant. Because our model is logistic, the average value of the predicted probabilities would be reported. The `at()` option fixes one or more covariates to the value(s) specified and can be used with both factor and continuous variables. Thus, if you typed `margins, at(age=40)`, then `margins` would average over the data the responses for everybody, setting `age=40`. Here is what happens when you type that:

```
. margins, at(age=40)
Predictive margins                                  Number of obs = 3,000
Model VCE: OIM

Expression: Pr(outcome), predict()
At: age = 40
```

	Margin	Delta-method std. err.	z	P>|z|	[95% conf.	interval]
_cons	.1133603	.0070731	16.03	0.000	.0994972	.1272234

Reported is the margin for `age` = 40, adjusted for the other covariates in our model.

If we wanted to obtain the margins for age 30, 35, 40, 45, and 50, we could type

```
. margins, at(age=(30 35 40 45 50))
```

or, equivalently,

```
. margins, at(age=(30(5)50))
```

Example 8: Margins of interactions

Our model is

```
. logistic outcome i.sex i.group sex#group age
```

We can obtain the margins of all possible combinations of the levels of sex and the levels of group by typing

```
. margins sex#group
Predictive margins                                      Number of obs = 3,000
Model VCE: OIM
Expression: Pr(outcome), predict()
```

	Margin	Delta-method std. err.	z	P>|z|	[95% conf.	interval]
sex#group						
Male#1	.2379605	.0237178	10.03	0.000	.1914745	.2844465
Male#2	.0658294	.0105278	6.25	0.000	.0451953	.0864636
Male#3	.0538001	.0136561	3.94	0.000	.0270347	.0805656
Female#1	.2158632	.0112968	19.11	0.000	.1937218	.2380045
Female#2	.2054406	.0183486	11.20	0.000	.1694781	.2414032
Female#3	.085448	.0533914	1.60	0.110	-.0191973	.1900932

The first line in the table reports the marginal probability for sex = 0 (male) and group = 1. That is, it reports the estimated probability if everyone in the data were treated as if they were sex = 0 and group = 1.

Also reported are all the other combinations of sex and group.

By the way, we could have typed margins sex#group even if our fitted model did not include sex#group. Estimation is one thing, and asking questions about the nature of the estimates is another. margins does, however, require that i.sex and i.group appear somewhere in the model, because fixing a value outside the model would just produce the grand margin, and you can separately ask for that if you want it by typing margins without arguments.

Example 9: Decomposing margins

We have the model

```
. logistic outcome i.sex i.group sex#group age
```

In example 6, we typed margins sex and obtained 0.160 for males and 0.197 for females. We are going to decompose each of those numbers. Let us explain:

1. The margin for males, 0.160, treats everyone as if they were male, and that amounts to simultaneously
   1a. treating males as males and
   1b. treating females as males.
2. The margin for females, 0.197, treats everyone as if they were female, and that amounts to simultaneously
   2a. treating males as females and
   2b. treating females as females.

The margins 1a and 1b are the decomposition of 1, and the margins 2a and 2b are the decomposition of 2.

We could obtain 1a and 2a by typing

```
. margins if sex==0, at(sex=(0 1))
```

because the qualifier `if sex==0` would restrict `margins` to running on only the males. Similarly, we could obtain 1b and 2b by typing

```
. margins if sex==1, at(sex=(0 1))
```

We run these examples below:

```
. margins if sex==0, at(sex=(0 1))
Predictive margins                                      Number of obs = 1,498
Model VCE: OIM

Expression: Pr(outcome), predict()
1._at: sex = 0
2._at: sex = 1
```

	Margin	Delta-method std. err.	z	P>|z|	[95% conf.	interval]
_at						
1	.0794393	.0062147	12.78	0.000	.0672586	.0916199
2	.1335584	.0127351	10.49	0.000	.1085981	.1585187

```
. margins if sex==1, at(sex=(0 1))
Predictive margins                                      Number of obs = 1,502
Model VCE: OIM

Expression: Pr(outcome), predict()
1._at: sex = 0
2._at: sex = 1
```

	Margin	Delta-method std. err.	z	P>|z|	[95% conf.	interval]
_at						
1	.2404749	.0199709	12.04	0.000	.2013326	.2796171
2	.2596538	.0104756	24.79	0.000	.2391219	.2801857

Putting together the results from example 6 and the results above, we have

Margin treating individuals as themselves	0.170
Margin treating individuals as male	0.160
Margin treating male as male	0.079
Margin treating female as male	0.240
Margin treating individuals as female	0.197
Margin treating male as female	0.134
Margin treating female as female	0.260

Example 10: Testing margins—contrasts of margins

Continuing with the previous example, it would be interesting to test the equality of 2b and 1b, to test whether the average probability of a positive outcome for females treated as females is equal to that for females treated as males. That test would be different from testing the overall significance of `sex` in our model. The test performed on our model would be a test of whether the probability of a positive outcome differs between males and females when they have equal values of the other covariates. The test of equality of margins is a test of whether the *average* probabilities differ given the different pattern of values of the other covariates that the two sexes have in our data.

We can also perform such tests by treating the results from margins as estimation results. There are three steps required to perform tests on margins. First, you must arrange it so that all the margins of interest are reported by just one margins command. Second, you must specify margins' post option. Third, you perform the test with the test command.

Such tests and comparisons can be readily performed by contrasting margins; see [R] **margins, contrast**. Also see *Contrasts of margins—effects (discrete marginal effects)* in [R] **marginsplot**.

In the previous example, we used two commands to obtain our results, namely,

```
. margins if sex==0, at(sex=(0 1))
. margins if sex==1, at(sex=(0 1))
```

We could, however, have obtained the same results by typing just one command:

```
. margins, over(sex) at(sex=(0 1))
```

Performing margins, over(sex) first restricts the sample to sex==0 and then restricts it to sex==1, and that is equivalent to the two different if conditions that we specified before.

To test whether females treated as females is equal to females treated as males, we will need to type

```
. margins, over(sex) at(sex=(0 1)) post
. test _b[2._at#1.sex] = _b[1._at#1.sex]
```

We admit that the second command may seem to have come out of nowhere. When we specify post on the margins command, margins behaves as if it were an estimation command, which means that 1) it posts its estimates and full VCE to e(), 2) it gains the ability to replay results just as any estimation command can, and 3) it gains access to the standard postestimation commands. Item 3 explains why we could use test. We learned that we wanted to test _b[2._at#1.sex] and _b[1._at#1.sex] by replaying the estimation results, but this time with the standard estimation command coeflegend option. So, what we typed was

```
. margins, over(sex) at(sex=(0 1)) post
. margins, coeflegend
. test _b[2._at#1.sex] = _b[1._at#1.sex]
```

We will let you try `margins, coeflegend` for yourself. The results of running the other two commands are

```
. margins, over(sex) at(sex=(0 1)) post
Predictive margins                                      Number of obs = 3,000
Model VCE: OIM

Expression: Pr(outcome), predict()
Over:       sex
1._at: 0.sex
           sex = 0
       1.sex
           sex = 0
2._at: 0.sex
           sex = 1
       1.sex
           sex = 1
```

	Margin	Delta-method std. err.	z	P>|z|	[95% conf.	interval]
_at#sex						
1#Male	.0794393	.0062147	12.78	0.000	.0672586	.0916199
1#Female	.2404749	.0199709	12.04	0.000	.2013326	.2796171
2#Male	.1335584	.0127351	10.49	0.000	.1085981	.1585187
2#Female	.2596538	.0104756	24.79	0.000	.2391219	.2801857

```
. test _b[2._at#1.sex] = _b[1._at#1.sex]
 ( 1)  - 1bn._at#1.sex + 2._at#1.sex = 0

           chi2(  1) =    0.72
         Prob > chi2 =    0.3951
```

We can perform the same test in one command using contrasts of margins:

```
. logistic outcome i.sex i.group sex#group age
  (output omitted)
. margins, over(sex) at(sex=(0 1)) contrast(atcontrast(r._at) wald)
Contrasts of predictive margins                         Number of obs = 3,000
Model VCE: OIM

Expression: Pr(outcome), predict()
Over:       sex
1._at: 0.sex
           sex = 0
       1.sex
           sex = 0
2._at: 0.sex
           sex = 1
       1.sex
           sex = 1
```

	df	chi2	P>chi2
_at@sex			
(2 vs 1) Male	1	14.59	0.0001
(2 vs 1) Female	1	0.72	0.3951
Joint	2	16.13	0.0003

	Contrast	Delta-method std. err.	[95% conf.	interval]
_at@sex				
(2 vs 1) Male	.0541192	.0141706	.0263453	.081893
(2 vs 1) Female	.0191789	.0225516	-.0250215	.0633793

We refit our logistic model because its estimation results were replaced when we posted our margins. The syntax to perform the contrast we want is admittedly not obvious. Contrasting (testing) across `at()` groups is more difficult than contrasting across the margins themselves or across `over()` groups, because we have no natural place for the contrast operators (`r.`, in our case). We also explicitly requested Wald tests of the contrasts, which are not provided by default. Nevertheless, the χ^2 statistic and its p-value for (`2 vs 1`) for `female` matches the results of our `test` command. We also obtain the test of whether the response of males treated as males is equal to the response of males treated as females.

For a gentler introduction to contrasts of margins, see [R] **margins, contrast**.

Example 11: Margins of a specified prediction

We will fit the model

```
. use https://www.stata-press.com/data/r18/margex
. tobit ycn i.sex i.group sex#group age, ul(90)
```

and we will tell the following story about the variables: We run a peach orchard where we allow people to pick their own peaches. A person receives one empty basket in exchange for $20, along with the right to enter the orchard. There is no official limit on how many peaches a person can pick, but only 90 peaches will fit into a basket. The dependent variable in the above tobit model, `ycn`, is the number of peaches picked. We use tobit, a special case of censored-normal regression, because `ycn` is censored at 90.

After fitting this model, if we typed

```
. margins sex
```

we would obtain the margins for males and for females of the uncensored number of peaches picked. We would obtain that because `predict` after `tobit` produces the uncensored number by default. To obtain the censored prediction, we would have to specify `predict`'s `ystar(.,90)` option. If we want the margins based on that response, we type

```
. margins sex, predict(ystar(.,90))
```

The results of typing that are

```
. tobit ycn i.sex i.group sex#group age, ul(90)
  (output omitted)
. margins sex, predict(ystar(.,90))
Predictive margins                                      Number of obs = 3,000
Model VCE: OIM
Expression: E(ycn*|ycn<90), predict(ystar(.,90))
```

	Margin	Delta-method std. err.	z	P>|z|	[95% conf.	interval]
sex						
Male	62.21804	.5996952	103.75	0.000	61.04266	63.39342
Female	78.34272	.4555278	171.98	0.000	77.4499	79.23553

In our previous examples, `sex` = 1 has designated females, so evidently the females visiting our orchard are better at filling baskets than the men.

Example 12: Margins of a specified expression

Continuing with our peach orchard example and the previously fit model

```
. use https://www.stata-press.com/data/r18/margex
. tobit ycn i.sex i.group sex#group age, ul(90)
```

let's examine how well our baskets are working for us. What is the proportion of the number of peaches actually picked to the number that would have been picked were the baskets larger? As mentioned in example 11, `predict, ystar(.,90)` produces the expected number picked given the limit of basket size. `predict, xb` would predict the expected number without a limit. We want the ratio of those two predictions. That ratio will measure as a proportion how well the baskets work. Thus, we could type

```
. margins sex, expression(predict(ystar(.,90))/predict(xb))
```

That would give us the proportion for everyone treated as male and everyone treated as female, but what we want to know is how well baskets work for true males and true females, so we will type

```
. margins, over(sex) expression(predict(ystar(0,90))/predict(xb))
Predictive margins                                      Number of obs = 3,000
Model VCE: OIM
Expression: predict(ystar(0,90))/predict(xb)
Over:       sex
```

	Margin	Delta-method std. err.	z	P>|z|	[95% conf.	interval]
sex						
Male	.9811785	.0013037	752.60	0.000	.9786233	.9837338
Female	.9419962	.0026175	359.88	0.000	.9368659	.9471265

By the way, we could count the number of peaches saved by the limited basket size during the period of data collection by typing

```
. count
  3,000
. margins, expression(3000*(predict(xb)-predict(ystar(.,90))))
  (output omitted)
```

The number of peaches saved turns out to be 9,183.

Example 13: Margins with multiple outcomes (responses)

Estimation commands such as `mlogit` and `mprobit` (see [R] **mlogit** and [R] **mprobit**) calculate multiple responses, and those multiple responses are reflected in the options available with `predict` after estimation. Obtaining margins for such estimators is thus the same as obtaining margins of a specified prediction, which was demonstrated in example 11. The solution is to include the *predict_opt* that selects the desired response in `margins`' `predict(`*predict_opt*`)` option.

If we fit the multinomial logistic model

```
. mlogit group i.sex age
```

then to obtain the margins for the probability that `group` = 1, we would type

```
. margins sex, predict(outcome(1))
```

and to obtain the margins for the probability that `group` = 3, we would type

```
. margins sex, predict(outcome(3))
```

To obtain the margins for each of these outcomes simultaneously, type

```
. margins sex, predict(outcome(1)) predict(outcome(3))
```

We learned about the `outcome(1)` and `outcome(3)` options by looking in [R] **mlogit postestimation**. For an example using `margins` with a multiple-outcome estimator, see example 4 in [R] **mlogit postestimation**.

Example 14: Margins with multiple equations

Estimation commands such as `mvreg`, `manova`, `sureg`, and `reg3` (see [MV] **mvreg**, [MV] **manova**, [R] **sureg**, and [R] **reg3**) fit multiple equations. Obtaining margins for such estimators is the same as obtaining margins with multiple outcomes (see example 13), which in turn is the same as obtaining margins of a specified prediction (see example 11). You place the relevant option from the estimator's `predict` command into `margins`' `predict(`*predict_opt*`)` option.

If we fit the seemingly unrelated regression model

```
. sureg (y = i.sex age) (distance = i.sex i.group)
```

we can obtain the marginal means of `y` for males and females by typing

```
. margins sex, predict(equation(y))
```

and we can obtain the marginal means of `distance` by typing

```
. margins sex, predict(equation(distance))
```

We could obtain the difference between the margins of `y` and `distance` by typing

```
. margins sex, expression(predict(equation(y)) -
> predict(equation(distance)))
```

More examples can be found in [MV] **manova** and [MV] **manova postestimation**.

Example 15: Margins evaluated out of sample

You can fit your model on one dataset and use `margins` on another if you specify `margins`' `noesample` option. Remember that `margins` reports estimated average responses, and, unless you lock all the covariates at fixed values by using the `at()` option, the remaining variables are allowed to vary as they are observed to vary in the data. That is indeed the point of using `margins`. The fitted model provides the basis for adjusting for the remaining variables, and the data provide their values. The predictions produced by `margins` are of interest assuming the data used by `margins` are in some sense interesting or representative. In some cases, you might need to fit your model on one set of data and perform `margins` on another.

In example 11, we fit the model

```
. tobit ycn i.sex i.group sex#group age, ul(90)
```

and we told a story about our peach orchard in which we charged people $20 to collect a basket of peaches, where baskets could hold at most 90 peaches. Let us now tell you that we believe the data on which we estimated those margins were unrepresentative, or at least, we have a more representative sample stored in another `.dta` file. That dataset includes the demographics of our customers but does not include counts of peaches picked. It is a lot of work counting those peaches.

Thus, we will fit our model just as we did previously using the detailed data, but we will bring the other, more representative dataset into memory before issuing the `margins sex, predict(ystar(.,90))` command, and we will add `noesample` to it.

```
. use https://www.stata-press.com/data/r18/margex
(Artificial data for margins)
. tobit ycn i.sex i.group sex#group age, ul(90)
  (output omitted)
. use https://www.stata-press.com/data/r18/peach
. margins sex, predict(ystar(.,90)) noesample
Predictive margins                                      Number of obs = 2,727
Model VCE: OIM
Expression: E(ycn*|ycn<90), predict(ystar(.,90))
```

	Margin	Delta-method std. err.	z	P>|z|	[95% conf.	interval]
sex						
0	56.79774	1.003731	56.59	0.000	54.83046	58.76502
1	75.02146	.6437446	116.54	0.000	73.75974	76.28317

In example 12, we produced an estimate of the number of peaches saved by the limited-size baskets. We can update that estimate using the new demographic data by typing

```
. count
  2,727
. margins, exp(2727*(predict(xb)-predict(ystar(.,90)))) noesample
  (output omitted)
```

By running the above, we find that the updated number of peaches saved is 6,408.

Obtaining margins of derivatives of responses (a.k.a. marginal effects)

Derivatives of responses are themselves responses, so everything said above in *Obtaining margins of responses* is equally true of derivatives of responses, and every example above could be repeated here substituting the derivative of the response for the response.

Derivatives are of interest because they are an informative way of summarizing fitted results. The change in a response for a change in the covariate is easy to understand and to explain. In simple models, one hardly needs `margins` to assist in obtaining such margins. Consider the simple linear regression

$$y = \beta_0 + \beta_1 \times \text{sex} + \beta_2 \times \text{age} + \epsilon$$

The derivatives of the responses are

$$dy/d(\text{sex}) = \beta_1$$

$$dy/d(\text{age}) = \beta_2$$

`margins` computes these derivatives using $\widehat{\beta}_1$ and $\widehat{\beta}_2$, the coefficients from the linear predictor of y. How does the linear predictor $\widehat{y}$ change between males and females? It changes by $\widehat{\beta}_1$. How does $\widehat{y}$ change with age? It changes by $\widehat{\beta}_2$ per year.

If you make the model a little more complicated, however, the need for margins arises. Consider the model

$$y = \beta_0 + \beta_1 \times \text{sex} + \beta_2 \times \text{age} + \beta_3 \times \text{age}^2 + \epsilon$$

Now, the derivative with respect to age is

$$dy/d(\text{age}) = \beta_2 + 2 \times \beta_3 \times \text{age}$$

The change in y for a change in age itself changes with age, and so to better understand the fitted results, you might want to make a table of the change in y for a change in age for age $= 30$, age $= 40$, and age $= 50$. `margins` can do that.

Consider an even more complicated model, such as

$$\begin{aligned} y = {} & \beta_0 + \beta_1 \times \text{sex} + \beta_2 \times \text{age} + \beta_3 \times \text{age}^2 + \beta_4 \times \text{bp} + \beta_5 \times \text{sex} \times \text{bp} + \beta_6 \times \text{tmt} \\ & + \beta_7 \times \text{tmt} \times \text{age} + \beta_8 \times \text{tmt} \times \text{age}^2 + \epsilon \end{aligned} \tag{1}$$

The derivatives are

$$dy/d(\text{sex}) = \beta_1 + \beta_5 \times \text{bp}$$

$$dy/d(\text{age}) = \beta_2 + 2 \times \beta_3 \times \text{age} + \beta_7 \times \text{tmt} + 2 \times \beta_8 \times \text{tmt} \times \text{age}$$

$$dy/d(\text{bp}) = \beta_4 + \beta_5 \times \text{sex}$$

$$dy/d(\text{tmt}) = \beta_6 + \beta_7 \times \text{age} + \beta_8 \times \text{age}^2$$

At this point, `margins` becomes indispensable.

Use at() freely, especially with continuous variables

An option one tends to use frequently with derivatives of responses is `at()`. Such use is often to better understand or to communicate how the response varies, or, in technical jargon, to explore the nature of the response surface.

For instance, the effect $dy/d(\text{tmt})$ in (1) is equal to $\beta_6 + \beta_7 \times \text{age} + \beta_8 \times \text{age}^2$, and so simply to understand how treatment varies with age, we may want to fix age at various values. We might type

```
. margins, dydx(tmt) at(age=(30 40 50))
```

Expressing derivatives as elasticities

You specify the `dydx(`*varname*`)` option on the `margins` command to use $dy/d(varname)$ as the response variable. If you want that derivative expressed as an elasticity, you can specify `eyex(`*varname*`)`, `eydx(`*varname*`)`, or `dyex(`*varname*`)`. You substitute `e` for `d` where you want an elasticity. In the discussion below, when we talk about y, we are referring to the linear predictions from the model.

The formulas are

$$\begin{aligned}
\texttt{dydx()} &= dy/dx \\
\texttt{eyex()} &= dy/dx \times (x/y) \\
\texttt{eydx()} &= dy/dx \times (1/y) = d\ln(y)/dx \\
\texttt{dyex()} &= dy/dx \times (x) \quad\ = dy/d\ln(x)
\end{aligned}$$

and the interpretations are

`dydx()`:		change in y for a		change in x
`eyex()`:	proportional	change in y for a	proportional	change in x
`eydx()`:	proportional	change in y for a		change in x
`dyex()`:		change in y for a	proportional	change in x

As `margins` always does with response functions, calculations are made at the observational level and are then averaged. Let's assume that in observation 5, $dy/dx = 0.5$, $y = 15$, and $x = 30$; then

$$\begin{aligned}
\texttt{dydx()} &= 0.5 \\
\texttt{eyex()} &= 1.0 \\
\texttt{eydx()} &= 0.03 \\
\texttt{dyex()} &= 15.0
\end{aligned}$$

Many social scientists would informally explain the meaning of `eyex()` $= 1$ as "y increases 100% when x increases 100%" or as "y doubles when x doubles", although neither statement is literally true. `eyex()`, `eydx()`, and `dyex()` are rates evaluated at a point, just as `dydx()` is a rate, and all such interpretations are valid only for small (infinitesimal) changes in x. It is true that `eyex()` $= 1$ means y increases with x at a rate such that, if the rate were constant, y would double if x doubled. This issue of casual interpretation is no different from casually interpreting `dydx()` as if it represents the response to a unit change. It is not necessarily true that `dydx()` $= 0.5$ means that "y increases by 0.5 if x increases by 1". It is true that "y increases with x at a rate such that, if the rate were constant, y would increase by 0.5 if x increased by 1".

dydx(), eyex(), eydx(), and dyex() may be used with continuous x variables. dydx() and eydx() may also be used with factor variables. For eydx(), effects are based on $d\ln(y)/dx$ for continuous covariates and on differences of $\ln(y)$ for discrete covariates. Estimates based on logarithms have better computational properties and allow us to think of semielasticities as coefficients in a regression with log dependent variables. As we discuss next, when we are analyzing marginal effects of categorical variables, we think in terms of discrete differences instead of derivatives.

Derivatives versus discrete differences

In (1),

$$y = \beta_0 + \beta_1 \times \text{sex} + \beta_2 \times \text{age} + \beta_3 \times \text{age}^2 + \beta_4 \times \text{bp} + \beta_5 \times \text{sex} \times \text{bp} + \beta_6 \times \text{tmt} + \beta_7 \times \text{tmt} \times \text{age} + \beta_8 \times \text{tmt} \times \text{age}^2 + \epsilon$$

Let us call your attention to the derivatives of y with respect to age and sex:

$$dy/d(\text{age}) = \beta_2 + 2 \times \beta_3 \times \text{age} + \beta_7 \times \text{tmt} + 2 \times \beta_8 \times \text{tmt} \times \text{age} \tag{2}$$

$$dy/d(\text{sex}) = \beta_1 + \beta_5 \times \text{bp} \tag{3}$$

age is presumably a continuous variable, and (2) is precisely how margins calculates its derivatives when you type margins, dydx(age). sex, however, is presumably a factor variable, and margins does not necessarily make the calculation using (3) were you to type margins, dydx(sex). We will explain, but let us first clarify what we mean by a continuous and a factor variable. Say that you fit (1) by typing

```
. regress y i.sex age c.age#c.age i.bp bp#sex
> i.tmt tmt#c.age tmt#c.age#c.age
```

It is important that sex entered the model as a factor variable. It would not do to type regress y sex ... because then sex would be a continuous variable, or at least it would be a continuous variable from Stata's point of view. The model estimates would be the same, but margins' understanding of those estimates would be a little different. With the model fit using i.sex, margins understands that either sex is 0 or sex is 1. With the model fit using sex, margins thinks sex is continuous and, for instance, sex = 1.5 is a possibility.

margins calculates dydx() differently for continuous and for factor variables. For continuous variables, margins calculates dy/dx using the fitted values from regress. For factor variables, margins calculates the discrete first difference from the base category. To obtain that for sex, write down the model and then subtract from it the model evaluated at the base category for sex, which is sex = 0. If you do that, you will get the same formula as we obtained for the derivative, namely,

$$\text{discrete difference}\{(\text{sex} = 1) - (\text{sex} = 0)\} = \beta_1 + \beta_5 \times \text{bp}$$

We obtain the same formula because our model is linear regression. Outside of linear regression, and outside of linear response functions generally, the discrete difference is not equal to the derivative. The discrete difference is not equal to the derivative for logistic regression, probit, etc. The discrete difference calculation is generally viewed as better for factor variables than the derivative calculation because the discrete difference is what would actually be observed.

If you want the derivative calculation for your factor variables, specify the continuous option on the margins command.

Example 16: Average marginal effect (partial effects)

Concerning the title of this example, the way we use the term marginal effect, the effects of factor variables are calculated using discrete first differences. If you wanted the continuous calculation, you would specify `margins`' `continuous` option in what follows.

```
. use https://www.stata-press.com/data/r18/margex
(Artificial data for margins)
. logistic outcome treatment##group age c.age#c.age treatment#c.age
  (output omitted)
. margins, dydx(treatment)
Average marginal effects                                  Number of obs = 3,000
Model VCE: OIM
Expression: Pr(outcome), predict()
dy/dx wrt:  1.treatment
```

	dy/dx	Delta-method std. err.	z	P>|z|	[95% conf.	interval]
1.treatment	.0385625	.0162848	2.37	0.018	.0066449	.0704801

Note: dy/dx for factor levels is the discrete change from the base level.

The average marginal effect of treatment on the probability of a positive outcome is 0.039.

Example 17: Average marginal effect of all covariates

We will continue with the model

```
. logistic outcome treatment##group age c.age#c.age treatment#c.age
```

if we wanted the average marginal effects for all covariates, we would type `margins, dydx(*)` or `margins, dydx(_all)`; they mean the same thing. This is probably the most common way `margins, dydx()` is used.

```
. margins, dydx(*)
Average marginal effects                                  Number of obs = 3,000
Model VCE: OIM
Expression: Pr(outcome), predict()
dy/dx wrt:  1.treatment 2.group 3.group age
```

	dy/dx	Delta-method std. err.	z	P>|z|	[95% conf.	interval]
1.treatment	.0385625	.0162848	2.37	0.018	.0066449	.0704801
group						
2	-.0776906	.0181584	-4.28	0.000	-.1132805	-.0421007
3	-.1505652	.0400882	-3.76	0.000	-.2291366	-.0719937
age	.0095868	.0007796	12.30	0.000	.0080589	.0111148

Note: dy/dx for factor levels is the discrete change from the base level.

Example 18: Evaluating marginal effects over the response surface

Continuing with the model

```
. logistic outcome treatment##group age c.age#c.age treatment#c.age
```

What follows maps out the entire response surface of our fitted model. We report the marginal effect of treatment evaluated at age $= 20, 30, \ldots, 60$, by each level of group.

```
. margins group, dydx(treatment) at(age=(20(10)60))
Conditional marginal effects                              Number of obs = 3,000
Model VCE: OIM
Expression: Pr(outcome), predict()
dy/dx wrt:  1.treatment
1._at: age = 20
2._at: age = 30
3._at: age = 40
4._at: age = 50
5._at: age = 60
```

	dy/dx	Delta-method std. err.	z	P>|z|	[95% conf.	interval]
0.treatment	(base outcome)					
1.treatment						
_at#group						
1 1	-.0208409	.0152862	-1.36	0.173	-.0508013	.0091196
1 2	.009324	.0059896	1.56	0.120	-.0024155	.0210635
1 3	.0006558	.0048682	0.13	0.893	-.0088856	.0101972
2 1	-.0436964	.0279271	-1.56	0.118	-.0984325	.0110397
2 2	.0382959	.0120405	3.18	0.001	.014697	.0618949
2 3	.0064564	.0166581	0.39	0.698	-.0261929	.0391057
3 1	-.055676	.0363191	-1.53	0.125	-.1268601	.015508
3 2	.1152235	.0209858	5.49	0.000	.074092	.156355
3 3	.0284808	.0471293	0.60	0.546	-.0638908	.1208524
4 1	-.027101	.0395501	-0.69	0.493	-.1046177	.0504158
4 2	.2447682	.0362623	6.75	0.000	.1736954	.315841
4 3	.0824401	.1025028	0.80	0.421	-.1184616	.2833418
5 1	.0292732	.0587751	0.50	0.618	-.0859239	.1444703
5 2	.3757777	.0578106	6.50	0.000	.2624709	.4890844
5 3	.1688268	.1642191	1.03	0.304	-.1530368	.4906904

Note: dy/dx for factor levels is the discrete change from the base level.

Obtaining margins with survey data and representative samples

The standard errors and confidence intervals produced by `margins` are based by default on the delta method applied to the VCE of the current estimates. Delta-method standard errors treat the covariates at which the response is evaluated as given or fixed. Such standard errors are appropriate if you specify `at()` to fix the covariates, and they are appropriate when you are making inferences about groups exactly like your sample whether you specify `at()` or not.

On the other hand, if you have a representative sample of the population or if you have complex survey data and if you want to make inferences about the underlying population, you need to account for the variation in the covariates that would arise in repeated sampling. You do that using `vce(unconditional)`, which invokes a different standard error calculation based on Korn and Graubard (1999). Syntactically, there are three cases. They all involve specifying the `vce(unconditional)` option on the `margins` command:

1. *You have a representative random sample, and you have not* `svyset` *your data.*
   When you fit the model, you need to specify the `vce(robust)` or `vce(cluster` *clustvar*`)` option. When you issue the `margins` command, you need to specify the `vce(unconditional)` option.
2. *You have a weighted sample, and you have not* `svyset` *your data.*
   You need to specify `[pw=weight]` when you fit the model and, of course, specify the `vce(unconditional)` option on the `margins` command. You do not need to specify the weights on the `margins` command because `margins` will obtain them from the estimation results.
3. *You have* `svyset` *your data, whether it be a simple random sample or something more complex including weights, strata, sampling units, or poststratification, and you are using the linearized variance estimator.*
   You need to use the `svy` prefix when you fit the model. You need to specify `vce(unconditional)` when you issue the `margins` command. You do not need to respecify the weights.

   Even though the data are `svyset`, and even though the estimation was `svy` estimation, `margins` does not default to `vce(unconditional)`. It does not default to `vce(unconditional)` because there are valid reasons to want the data-specific, `vce(delta)` standard-error estimates. Whether you specify `vce(unconditional)` or not, `margins` uses the weights, so you do not need to respecify them even if you are using `vce(unconditional)`.

`vce(unconditional)` is allowed only after estimation with `vce(robust)`, `vce(cluster ...)`, or the `svy` prefix with the linearized variance estimator. If the VCE of the current estimates was specified as clustered, so will be the VCE estimates of `margins`. If the estimates were from a survey estimation, the survey settings in the dataset will be used by `margins`.

When you use `vce(unconditional)`, never specify `if` *exp* or `in` *range* on the `margins` command; instead, specify the `subpop(if` *exp*`)` option. You do that for the usual reasons; see [SVY] **Subpopulation estimation**. If you specify `over(`*varlist*`)` to examine subgroups, the subgroups will automatically be treated as subpopulations.

If you are using a replication-based variance estimator, you may want to use this method to estimate the variance of your margins; see [SVY] **svy postestimation**.

Example 19: Inferences for populations, margins of response

In example 6, we fit the model

```
. logistic outcome i.sex i.group sex#group age
```

and we obtained margins by sex and margins by group,

```
. margins sex group
```

If our data were randomly drawn from the population of interest and we wanted to account for this, we would have typed

```
. logistic outcome i.sex i.group sex#group age, vce(robust)
. margins sex group, vce(unconditional)
```

We do that below:

```
. logistic outcome i.sex i.group sex#group age, vce(robust)
  (output omitted)
. margins sex group, vce(unconditional)
Predictive margins                                      Number of obs = 3,000
Expression: Pr(outcome), predict()
```

	Unconditional					
	Margin	std. err.	z	P>|z|	[95% conf.	interval]
sex						
Male	.1600644	.0131685	12.16	0.000	.1342546	.1858743
Female	.1966902	.0104563	18.81	0.000	.1761963	.2171841
group						
1	.2251302	.0127069	17.72	0.000	.200225	.2500354
2	.150603	.0118399	12.72	0.000	.1273972	.1738088
3	.0736157	.0343188	2.15	0.032	.0063522	.1408793

The estimated margins are the same as they were in example 6, but the standard errors and confidence intervals differ, although not by much. Given that we have 3,000 observations in our randomly drawn sample, we should expect this.

Example 20: Inferences for populations, marginal effects

In example 17, we fit a logistic model and then obtained the average marginal effects for all covariates by typing

```
. logistic outcome treatment##group age c.age#c.age treatment#c.age
. margins, dydx(*)
```

To repeat that and also obtain standard errors for our population, we would type

```
. logistic outcome treatment##group age c.age#c.age treatment#c.age,
> vce(robust)
. margins, dydx(*) vce(unconditional)
```

The results are

```
. logistic outcome treatment##group age c.age#c.age treatment#c.age, vce(robust)
  (output omitted)
. margins, dydx(*) vce(unconditional)
Average marginal effects                                Number of obs = 3,000
Expression: Pr(outcome), predict()
dy/dx wrt:  1.treatment 2.group 3.group age
```

	Unconditional					
	dy/dx	std. err.	z	P>|z|	[95% conf.	interval]
1.treatment	.0385625	.0163872	2.35	0.019	.0064442	.0706808
group						
2	-.0776906	.0179573	-4.33	0.000	-.1128863	-.0424949
3	-.1505652	.0411842	-3.66	0.000	-.2312848	-.0698456
age	.0095868	.0007814	12.27	0.000	.0080553	.0111183

```
Note: dy/dx for factor levels is the discrete change from the base level.
```

Example 21: Inferences for populations with svyset data

See example 3 in [SVY] **svy postestimation**.

Standardizing margins

A standardized margin is the margin calculated on data different from the data used to fit the model. Typically, the word standardized is reserved for situations in which the alternate population is a reference population, which may be real or artificial, and which is treated as fixed.

Say that you work for a hospital and have fit a model of mortality on the demographic characteristics of the hospital's patients. At this stage, were you to type

```
. margins
```

you would obtain the mortality rate for your hospital. You have another dataset, `hstandard.dta`, that contains demographic characteristics of patients across all hospitals along with the population of each hospital recorded in the `pop` variable. You could obtain the expected mortality rate at your hospital if your patients matched the characteristics of the standard population by typing

```
. use https://www.stata-press.com/data/r18/hstandard, clear
. margins [fw=pop], noesample
```

You specified `noesample` because the margin is being calculated on data other than the data used to fit the model. You specified `[fw=pop]` because the reference dataset you are using included population counts, as many reference datasets do.

Obtaining margins as though the data were balanced

Here we discuss what are commonly called estimated marginal means or least-squares means. These are margins assuming that all levels of factor variables are equally likely or, equivalently, that the design is balanced. The seminal reference on these margins is Searle, Speed, and Milliken (1980).

In designed experiments, observations are often allocated in a balanced way so that the variances can be easily compared and decomposed. At the Acme Portable Widget Company, they are experimenting with a new machine. The machine has three temperature settings and two pressure settings; a combination of settings will be optimal on any particular day, determined by the weather. At start-up, one runs a quick test and chooses the optimal setting for the day. Across different days, each setting will be used about equally, says the manufacturer.

In experiments with the machine, 10 widgets were collected for stress testing at each of the settings over a six-week period. We wish to know the average stress-test value that can be expected from these machines over a long period.

Balancing using asbalanced

The data were intended to be balanced, but unfortunately, the stress test sometimes destroys samples before the stress can be measured. Thus, even though the experiment was designed to be balanced, the data are not balanced. You specify the `asbalanced` option to estimate the margins as if the data were balanced. So that you can compare the `asbalanced` results with the observed results, we will also include `margins` without the `asbalanced` option in what follows:

```
. use https://www.stata-press.com/data/r18/acmemanuf
. regress y pressure##temp
  (output omitted)
. margins
Predictive margins                                      Number of obs = 49
Model VCE: OLS
Expression: Linear prediction, predict()
```

	Margin	Delta-method std. err.	t	P>|t|	[95% conf.	interval]
_cons	109.9214	1.422629	77.27	0.000	107.0524	112.7904

```
. margins, asbalanced
Adjusted predictions                                    Number of obs = 49
Model VCE: OLS
Expression: Linear prediction, predict()
At: pressure   (asbalanced)
    temp       (asbalanced)
```

	Margin	Delta-method std. err.	t	P>|t|	[95% conf.	interval]
_cons	115.3758	1.530199	75.40	0.000	112.2899	118.4618

❑ Technical note

Concerning how `asbalanced` calculations are performed, if a factor variable has l levels, then each level's coefficient contributes to the response weighted by $1/l$. If two factors, a and b, interact, then each coefficient associated with their interaction is weighted by $1/(l_a \times l_b)$.

If a balanced factor interacts with a continuous variable, then each coefficient in the interaction is applied to the value of the continuous variable, and the results are weighted equally. So, if the factor being interacted has l_a levels, the effect of each coefficient on the value of the continuous covariate is weighted by $1/l_a$.

❑

Balancing by standardization

To better understand the balanced results, we can perform the balancing ourselves by using the standardizing method shown in *Standardizing margins*. To do that, we will input a balanced dataset and then type `margins, noesample`.

```
. use https://www.stata-press.com/data/r18/acmemanuf
. regress y pressure##temp
  (output omitted)
. drop _all
. input pressure temp
     pressure       temp
  1. 1 1
  2. 1 2
  3. 1 3
  4. 2 1
  5. 2 2
  6. 2 3
  7. end
. margins, noesample
Predictive margins                                      Number of obs = 6
Model VCE: OLS
Expression: Linear prediction, predict()
```

	Margin	Delta-method std. err.	t	P>|t|	[95% conf.	interval]
_cons	115.3758	1.530199	75.40	0.000	112.2899	118.4618

We obtain the same results as previously.

Balancing nonlinear responses

If our testing had produced a binary outcome, say, acceptable/unacceptable, rather than a continuous variable, we would type

```
. use https://www.stata-press.com/data/r18/acmemanuf, clear
. logistic acceptable pressure##temp
. margins, asbalanced
```

The result of doing that would be 0.680. If we omitted the `asbalanced` option, the result would have been 0.667. The two results are so similar because `acmemanuf.dta` is nearly balanced.

Even though the `asbalanced` option can be used on both linear and nonlinear responses, such as probabilities, there is an issue of which you should be aware. The most widely used formulas for balancing responses apply the balancing to the linear prediction, average that as if it were balanced, and then apply the nonlinear transform. That is the calculation that produced 0.680.

An alternative would be to apply the standardization method. That amounts to making the linear predictions observation by observation, applying the nonlinear transform to each, and then averaging the nonlinear result as if it were balanced. You could do that by typing

```
. use https://www.stata-press.com/data/r18/acmemanuf, clear
. logistic acceptable pressure##temp
. clear
. input pressure temp
  (see above for entered data)
. margins, noesample
```

The result from the standardization procedure would be 0.672. These two ways of averaging nonlinear responses are discussed in detail in Lane and Nelder (1982) within the context of general linear models.

Concerning the method used by the `asbalanced` option, if your data start balanced and you have a nonlinear response, you will get different results with and without the `asbalanced` option!

Treating a subset of covariates as balanced

So far, we have treated all the covariates as if they were balanced. `margins` will allow you to treat a subset of the covariates as balanced, too. For instance, you might be performing an experiment in which you are randomly allocating patients to a treatment arm and so want to balance on arm, but you do not want to balance the other characteristics because you want mean effects for the experiment's population.

In this example, we will imagine that the outcome of the experiment is continuous. We type

```
. use https://www.stata-press.com/data/r18/margex, clear
. regress y arm##sex sex##agegroup
. margins, at((asbalanced) arm)
```

If we wanted results balanced on `agegroup` as well, we could type

```
. margins, at((asbalanced) arm agegroup)
```

If we wanted results balanced on all three covariates, we could type

```
. margins, at((asbalanced) arm agegroup sex)
```

or we could type

```
. margins, at((asbalanced) _factor)
```

or we could type

```
. margins, asbalanced
```

Using fvset design

As a convenience feature, equivalent to

```
. regress y arm##sex sex##agegroup
. margins, at((asbalanced) arm sex)
```

is

```
. fvset design asbalanced arm sex
. regress y arm##sex sex##agegroup
. margins
```

The advantage of the latter is that you have to set the variables as balanced only once. This is useful when balancing is a design characteristic of certain variables and you wish to avoid accidentally treating them as unbalanced.

If you `save` your data after `fvset`ting, the settings will be remembered in future sessions. If you want to clear the setting(s), type

```
. fvset clear varlist
```

See [R] **fvset**.

Balancing in the presence of empty cells

The issue of empty cells is not exclusively an issue of balancing, but there are special considerations when balancing. Empty cells are discussed generally in *Estimability of margins*.

An empty cell is an interaction of levels of two or more factor variables for which you have no data. Usually, margins involving empty cells cannot be estimated. When balancing, there is an alternate definition of the margin that allows the margin to be estimated. `margins` makes the alternate calculation when you specify the `emptycells(reweight)` option. By default, `margins` uses the `emptycells(strict)` option.

If you have empty cells in your data and you request margins involving the empty cells, those margins will be marked as not estimable even if you specify the `asbalanced` option.

```
. use https://www.stata-press.com/data/r18/estimability, clear
(margins estimability)
. regress y sex##group
  (output omitted)
. margins sex, asbalanced
Adjusted predictions                                      Number of obs = 69
Model VCE: OLS
Expression: Linear prediction, predict()
At: sex     (asbalanced)
    group   (asbalanced)
```

	Margin	Delta-method std. err.	t	P>|t|	[95% conf.	interval]
sex						
Male	21.91389	1.119295	19.58	0.000	19.67572	24.15206
Female	.	(not estimable)				

This example is discussed in *Estimability of margins*, although without the `asbalanced` option. What is said there is equally relevant to the `asbalanced` case. For reasons explained there, the margin for `sex` = 1 (female) cannot be estimated.

The margin for `sex` = 1 can be estimated in the `asbalanced` case if you are willing to make an assumption. Remember that `margins` makes the balanced calculation by summing the responses associated with the levels and then dividing by the number of levels. If you specify `emptycells(reweight)`, `margins` sums what is available and divides by the number available. Thus, you are assuming that, whatever the responses in the empty cells, those responses are such that they would not change the overall mean of what is observed.

The results of specifying `emptycells(reweight)` are

```
. margins sex, asbalanced emptycells(reweight)
Adjusted predictions                                    Number of obs = 69
Model VCE: OLS
Expression:  Linear prediction, predict()
Empty cells: reweight
At: sex     (asbalanced)
    group   (asbalanced)
```

	Margin	Delta-method std. err.	t	P>|t|	[95% conf.	interval]
sex						
Male	21.91389	1.119295	19.58	0.000	19.67572	24.15206
Female	24.85185	1.232304	20.17	0.000	22.38771	27.316

Obtaining margins with nested designs

Introduction to nested designs

Factors whose meaning depends on other factors are called nested factors, and the factors on which their meaning depends are called the nesting factors. For instance, assume that we have a sample of patients and each patient is assigned to one doctor. Then, patient is nested within doctor. Let the identifiers of the first 5 observations of our data be

Doctor	Patient	Name
1	1	Fred
1	2	Mary
1	3	Bob
2	1	Karen
2	2	Hank

The first patient on one doctor's list has nothing whatsoever to do with the first patient on another doctor's list. The meaning of `patient` = 1 is defined only when the value of `doctor` is supplied.

Nested factors enter into models as interactions of nesting and nested; the nested factor does not appear by itself. We might estimate a model such as

```
. regress y ... i.doctor doctor#patient ...
```

You do not include `i.patient` because the coding for patient has no meaning except within doctor. Patient 1 is Fred for doctor 1 and Karen for doctor 2, etc.

`margins` provides an option to help account for the structure of nested models. The `within(`*varlist*`)` option specifies that `margins` estimate and report a set of margins for the value combinations of *varlist*. We might type

```
. margins, within(doctor)
```

Margin calculations are performed first for `doctor` = 1, then for `doctor` = 2, and so on.

Sometimes you need to specify `within()`, and other times you do not. Let's consider the particular model

```
. regress y i.doctor doctor#patient i.sex sex#doctor#patient
```

The guidelines are the following:

1. You may compute overall margins by typing
   `margins`.
2. You may compute overall margins within levels of a nesting factor by typing
   `margins, within(doctor)`.
3. You may compute margins of a nested factor within levels of its nesting factor by typing
   `margins patient, within(doctor)`.
4. You may compute margins of factors in your model, as long as the factor does not nest other factors and is not nested within other factors, by typing
   `margins sex`.
5. You may not compute margins of a nesting factor, such as `margins doctor`, because they are not estimable.

For examples using `within()`, see [R] **anova**.

Margins with nested designs as though the data were balanced

To obtain margins with nested designs as though the data were balanced, the guidelines are the same as above except that 1) you add the `asbalanced` option and 2) whenever you do not specify `within()`, you specify `emptycells(reweight)`. The updated guidelines are

1. You may compute overall margins by typing
   `margins, asbalanced emptycells(reweight)`.
2. You may compute overall margins within levels of a nesting factor by typing
   `margins, asbalanced within(doctor)`.
3. You may compute margins of a nested factor within levels of its nesting factor by typing
   `margins patient, asbalanced within(doctor)`.
4. You may compute margins of factors in your model, as long as the factor does not nest other factors and is not nested within other factors, by typing
   `margins sex, asbalanced emptycells(reweight)`.
5. You may not compute margins of a nesting factor, such as `margins doctor`, because they are not estimable.

Just as explained in *Using fvset design*, rather than specifying the `asbalanced` option, you may set the balancing characteristic on the factor variables once and for all by using the command `fvset design asbalanced` *varlist*.

❑ Technical note

Specifying either `emptycells(reweight)` or `within(`*varlist*`)` causes `margins` to rebalance over all empty cells in your model. If you have interactions in your model that are not involved in the nesting, `margins` will lose its ability to detect estimability.

❑

❑ Technical note

Careful readers will note that the description of `within(`*varlist*`)` matches closely the description of `over(`*varlist*`)`. The concept of nesting is similar to the concept of subpopulations. `within()` differs from `over()` in that it gracefully handles the missing cells when margins are computed as balanced.

❑

Coding of nested designs

In the *Introduction* to this section, we showed a coding of the nested variable `patient`, where the coding started over with each `doctor`:

Doctor	Patient	Name
1	1	Fred
1	2	Mary
1	3	Bob
2	1	Karen
2	2	Hank

That coding style is not required. The data could just as well have been coded

Doctor	Patient	Name
1	1	Fred
1	2	Mary
1	3	Bob
2	4	Karen
2	5	Hank

or even

Doctor	Patient	Name
1	1037239	Fred
1	2223942	Mary
1	0611393	Bob
2	4433329	Karen
2	6110271	Hank

Actually, either of the above two alternatives is better than the first one because `margins` will be better able to give you feedback about estimability should you make a mistake following the guidelines. On the other hand, both of these alternatives require more memory at the estimation step. If you run short of memory, you will need to recode your patient ID to the first coding style, which you could do by typing

```
. sort doctor patient
. by doctor: generate newpatient = _n
```

Alternatively, you can `set emptycells drop` and continue to use your patient ID variable just as it is coded. If you do this, we recommend that you remember to type `set emptycells keep` when you are finished; `margins` is better able to determine estimability that way. If you regularly work with large nested models, you can `set emptycells keep, permanently` so that the setting persists across sessions. See [R] **set emptycells**.

Special topics

Requirements for model specification

The results that margins reports are based on the most recently fit model or, in Stata jargon, the most recently issued estimation command. Here we discuss 1) mechanical requirements for how you specify that estimation command, 2) workarounds to use when those restrictions prove impossible, and 3) requirements for margins' predict(*pred_opt*) option to work.

Concerning 1, when you specify the estimation command, covariates that are logically factor variables must be Stata factor variables, and that includes indicator variables, binary variables, and dummies. It will not do to type

```
. regress y ... sex ...
```

even if sex is a 0/1 variable. You must type

```
. regress y ... i.sex ...
```

If you violate this rule, you will not get incorrect results, but you will discover that you will be unable to obtain margins on sex:

```
. margins sex
factor sex not found in list of covariates
r(111);
```

It is also important that if the same continuous variable appears in your model more than once, differently transformed, those transforms be performed via Stata's factor-variable notation. It will not do to type

```
. generate age2 = age^2
. regress y ... age age2 ...
```

You must type

```
. regress y ... age c.age#c.age ...
```

You must do that because margins needs to know everywhere that variable appears in the model if it is to be able to set covariates to fixed values.

Concerning 2, sometimes the transformations you desire may not be achievable using the factor-variable notation; in those situations, there is a workaround. Let's assume you wish to estimate

```
. generate age1_5 = age^1.5
. regress y ... age age1_5 ...
```

There is no factor-variable notation for including age and $\texttt{age}^{1.5}$ in a model, so obviously you are going to obtain the estimates by typing just what we have shown. In what follows, it would be okay if there are interactions of age and age1_5 with other variables specified by the factor-variable notation, so the model could just as well be

```
. regress y ... age age1_5 sex#c.age sex#c.age1_5 ...
```

Let's assume you have fit one of these two models. On any subsequent margins command where you leave age free to vary, there will be no issue. You can type

```
. margins sex
```

and results will be correct. Issues arise when you attempt to fix age at predetermined values. The following would produce incorrect results:

```
. margins sex, at(age=20)
```

The results would be incorrect because they leave age1_5 free to vary, and, logically, fixing age implies that age1_5 should also be fixed. Because we were unable to state the relationship between age and age1_5 using the factor-variable notation, margins does not know to fix age1_5 at $20^{1.5}$ when it fixes age at 20. To get the correct results, you must fix the value of age1_5 yourself:

```
. margins sex, at(age=20 age1_5=89.442719)
```

That command produces correct results. In the command, 89.442719 is $20^{1.5}$.

In summary, when there is a functional relationship between covariates of your model and that functional relationship is not communicated to margins via the factor-variable notation, then it becomes your responsibility to ensure that all variables that are functionally related are set to the appropriate fixed values when any one of them is set to a fixed value.

Concerning 3, we wish to amend our claim that you can calculate margins for anything that predict will produce. We need to add a qualifier. Let us show you an example where the statement is not true. After regress, predict will predict something it calls pr(a,b), which is the probability $a \leq y \leq b$. Yet if we attempted to use pr() with margins after estimation by regress, we would obtain

```
. margins sex, predict(pr(10,20))
prediction is a function of possibly stochastic quantities other than e(b)
r(498);
```

What we should have stated was that you can calculate margins for anything that predict will produce for which all the estimated quantities used in its calculation appear in e(V), the estimated VCE. pr() is a function of β, the estimated coefficients, and of s^2, the estimated variance of the residual. regress does not post the variance of the residual variance (sic) in e(V), or even estimate it, and therefore, predict(pr(10,20)) cannot be specified with margins after estimation by regress.

It is unlikely that you will ever encounter these kinds of problems because there are so few predictions where the components are not posted to e(V). If you do encounter the problem, the solution may be to specify nose to suppress the standard error calculation. If the problem is not with computing the margin, but with computing its standard error, margins will report the result:

```
. margins sex, predict(pr(10,20)) nose
  (output appears with SEs, tests, and CIs left blank)
```

❑ Technical note

Programmers: If you run into this after running an estimation command that you have written, be aware that as of Stata 11, you are supposed to set in e(marginsok) the list of options allowed with predict that are okay to use with margins. When that list is not set, margins looks for violations of its assumptions and, if it finds any, refuses to proceed.

❑

Estimability of margins

Sometimes margins will report that a margin cannot be estimated:

```
. use https://www.stata-press.com/data/r18/estimability, clear
(margins estimability)
. regress y sex##group
  (output omitted)
. margins sex
Predictive margins                                      Number of obs = 69
Model VCE: OLS
Expression: Linear prediction, predict()
```

	Margin	Delta-method std. err.	t	P>|t|	[95% conf.	interval]
sex						
Male	21	.8500245	24.71	0.000	19.30027	22.69973
Female	.	(not estimable)				

In the above output, the margin for sex = 0 (male) is estimated, but the margin for sex = 1 (female) is not estimable. This occurs because of empty cells. An empty cell is an interaction of levels of two or more factor variables for which you have no data. In the example, the lack of estimability arises because we have two empty cells:

```
. table sex group, nototals
```

	Group				
	1	2	3	4	5
Sex					
Male	2	9	27	8	2
Female	9	9	3		

To calculate the marginal mean response for sex = 1, we have no responses to average over for group = 4 and group = 5. We obviously could calculate that mean for the observations that really are sex = 1, but remember, the marginal calculation for sex = 1 treats everyone as if female, and we will thus have 8 and 2 observations for which we have no basis for estimating the response.

There is no solution for this problem unless you are willing to treat the data as if it were balanced and adjust your definition of a margin; see *Balancing in the presence of empty cells*.

Manipulability of tests

Manipulability is a problem that arises with some tests, and in particular, arises with Wald tests. Tests of margins are based on Wald tests, hence our interest. This is a generic issue and not specific to the margins command.

Let's understand the problem. Consider performing a test of whether some statistic ϕ is 0. Whatever the outcome of that test, it would be desirable if the outcome were the same were we to test whether the sqrt(ϕ) were 0, or whether ϕ^2 were 0, or whether any other monotonic transform of ϕ were 0 (for ϕ^2, we were considering only the positive half of the number line). If a test does not have that property, it is manipulable.

Wald tests are manipulable, and that means the tests produced by margins are manipulable. You can see this for yourself by typing

```
. use https://www.stata-press.com/data/r18/margex, clear
. replace y = y - 65
. regress y sex##group
. margins, df(.)
. margins, expression(predict(xb)^2)
```

To compare the results from the two `margins` commands, we added the `df(.)` option to the first one, forcing it to report a z statistic even though a t statistic would have been appropriate in this case. We would prefer if the test against zero produced by `margins, df(.)` was equal to the test produced by `margins, expression(predict(xb)^2)`. But alas, they produce different results. The first produces $z = 12.93$, and the second produces $z = 12.57$.

The difference is not much in our example, but behind the scenes, we worked to make it small. We subtracted 65 from `y` so that the experiment would be for a case where it might be reasonable that you would be testing against 0. One does not typically test whether the mean income in the United States is zero or whether the mean blood pressure of live patients is zero. Had we left `y` as it was originally, we would have obtained $z = 190$ and $z = 96$. We did not want to show that comparison to you first because the mean of `y` is so far from 0 that you probably would never be testing it. The corresponding difference in ϕ is tiny.

Regardless of the example, it is important that you base your tests in the metric where the likelihood surface is most quadratic. For further discussion on manipulability, see *Manipulability* in [R] **predictnl**.

This manipulability is not limited to Wald tests after estimation; you can also see the manipulability of results produced by linear regression just by applying nonlinear transforms to a covariate (Phillips and Park 1988; Gould 1996).

Using margins after the estimates use command

Assume you fit and used `estimates save` (see [R] **estimates save**) to save the estimation results:

```
. regress y sex##group age c.age*c.age if site==1
. ...
. estimates save mymodel
file mymodel.ster saved
```

Later, perhaps in a different Stata session, you reload the estimation results by typing

```
. estimates use mymodel
```

You plan to use `margins` with the reloaded results. You must remember that `margins` bases its results not only on the current estimation results but also on the current data in memory. Before you can use `margins`, you must reload the dataset on which you fit the model or, if you wish to produce standardized margins, some other dataset.

```
. use mydata, clear
(data for fitting models)
```

If the dataset you loaded contained the data for standardization, you can stop reading; you know that to produce standardized margins, you need to specify the `noesample` option.

We reloaded the original data and want to produce margins for the estimation sample. In addition to the data, `margins` requires that `e(sample)` be set, as `margins` will remind us:

```
. margins sex
e(sample) does not identify the estimation sample
r(322);
```

The best solution is to use `estimates esample` to rebuild `e(sample)`:

```
. estimates esample: y sex group age if site==1
```

If we knew we had no missing values in `y` and the covariates, we could type

```
. estimates esample: if site==1
```

Either way, `margins` would now work:

```
. margins sex
(usual output appears)
```

There is an alternative. We do not recommend it, but we admit that we have used it. Rather than rebuilding `e(sample)`, you can use `margins`' `noesample` option to tell `margins` to skip using `e(sample)`. You could then specify the appropriate `if` statement (if necessary) to identify the estimation sample:

```
. estimates use mymodel
. use mydata, clear
(data for fitting models)
. margins sex if !missing(y, sex, group age) & site==1, noesample
(usual output appears)
```

In the above, we are not really running on a sample different from the estimation sample; we are merely using `noesample` to fool `margins`, and then we are specifying on the `margins` command the conditions equivalent to re-create `e(sample)`.

If we wish to obtain `vce(unconditional)` results, however, `noesample` will be insufficient. We must also specify the `force` option,

```
. margins sex if !missing(y, sex, group age) & site==1,
> vce(unconditional) noesample force
(usual output appears)
```

Regardless of the approach you choose—resetting `e(sample)` or specifying `noesample` and possibly `force`—make sure you are right. In the `vce(delta)` case, you want to be right to ensure that you obtain the results you want. In the `vce(unconditional)` case, you need to be right because otherwise results will be statistically invalid.

Syntax of at()

In `at(`*atspec*`)`, *atspec* may contain one or more of the following specifications:

varlist

(*stat*) *varlist*

varname = #

varname = (*numlist*)

varname = `generate(`*exp*`)`

where

1. *varname*s must be covariates in the previously fit model (estimation command).
2. Variable names (whether in *varname* or *varlist*) may be continuous variables, factor variables, or specific level variables, such as `age`, `group`, or `3.group`.

3. *varlist* may also be one of three standard lists:
    a. _all (all covariates),
    b. _factor (all factor-variable covariates), or
    c. _continuous (all continuous covariates).
4. *stat* can be any of the following:

stat	Description	Variables allowed
asobserved	at observed values in the sample (default)	all
mean	means (default for *varlist*)	all
median	medians	continuous
p1	1st percentile	continuous
p2	2nd percentile	continuous
...	3rd–49th percentiles	continuous
p50	50th percentile (same as median)	continuous
...	51st–97th percentiles	continuous
p98	98th percentile	continuous
p99	99th percentile	continuous
min	minimums	continuous
max	maximums	continuous
zero	fixed at zero	continuous
base	base level	factors
asbalanced	all levels equally probable and sum to 1	factors

Any *stat* except zero, base, and asbalanced may be prefixed with an o to get the overall statistic—the sample over all over() groups. For example, omean, omedian, and op25. Overall statistics differ from their correspondingly named statistics only when the over() or within() option is specified. When no *stat* is specified, mean is assumed. If *stat* is not followed by a varlist, *stat* is ignored.

atspec can involve multiple settings for one covariate as well as settings for multiple covariates. The following rules are applied when more than one covariate or value is included:

1. When more than one covariate is referenced in *atspec* but each covariate is set to only one value, all settings are applied in combination. For example, at(x1=5 x2=0) results in margins being estimated under one scenario, with x1 set to 5, x2 set to 0, and all other covariates set to their observed values (the default).
2. When multiple values are specified for a covariate, the covariate will be set to each of the values in turn. For example, at(x1=5 x1=10) or, equivalently, at(x1=(5 10)) specifies that x1 be set first to 5 and then to 10.
3. When multiple values are specified for more than one covariate, all possible combinations of settings are applied in turn. For example, at(x1=(5 10) x2=(0 1)) results in margins being estimated under four scenarios: (x1 = 5 x2 = 0), (x1 = 5 x2 = 1), (x1 = 10 x2 = 0), and (x1 = 10 x2 = 1).
4. Settings may be specified for groups of covariates using three general varlists—_all, _factor, and _continuous. When *atspec* includes both specifications with general varlists and specifications with named covariates, the specifications for named covariates take precedence over general ones. For example, at(x1=10 (means) _all) sets x1 to 10 while setting all other covariates to their means.

5. Only one (*stat*) *varlist* specification can be applied to a covariate. If more than one is specified, the rightmost specification is respected. For example, `at((means) x1 x2 (medians) x1 x2)` sets both `x1` and `x2` to their medians.
6. When both a (*stat*) specification and another specification are included for a named covariate, the other specification takes precedence. For example, `at(x1=5 (means) x1)` sets `x1` to 5.

In addition, `at()` can be repeated. When multiple `at()` options are specified, *atspec*s are processed sequentially. For instance, `at(x1=5) at(x2=0)` results in `margins` being estimated under two scenarios. The first sets `x1` to 5 and all other covariates, including `x2`, to their observed values. The second sets `x2` to 0 and all other covariates to their observed values. Note that this is different from the single `at(x1=5 x2=0)` specification, which sets `x1` and `x2` to the specified values simultaneously.

Estimation commands that may be used with margins

`margins` may be used after most estimation commands.

`margins` cannot be used after estimation commands that do not produce full variance matrices, such as `exlogistic` and `expoisson` (see [R] **exlogistic** and [R] **expoisson**).

`margins` is all about covariates and cannot be used after estimation commands that do not post the covariates, which eliminates `gmm` (see [R] **gmm**).

`margins` cannot be used after some estimation commands that have an odd data organization, such as `nlogit`. However, `margins` has been specially adapted to work with many choice model estimation commands; see [CM] **margins**.

Video examples

Introduction to margins, part 1: Categorical variables

Introduction to margins, part 2: Continuous variables

Introduction to margins, part 3: Interactions

Glossary

adjusted mean. A *margin* when the response is the linear predictor from linear regression, ANOVA, etc. For some authors, adjusting also implies adjusting for unbalanced data. See *Obtaining margins of responses* and see *Obtaining margins as though the data were balanced*.

average marginal effect. See *marginal effect and average marginal effect*.

average partial effect. See *partial effect and average partial effect*.

conditional margin. A *margin* when the response is evaluated at fixed values of all the covariates. If any covariates are left to vary, the margin is called a predictive margin.

effect. The effect of x is the derivative of the *response* with respect to covariate x, or it is the difference in responses caused by a discrete change in x. Also see *marginal effect*.

The effect of x measures the change in the response for a change in x. Derivatives or differences might be reported as elasticities. If x is continuous, the effect is measured continuously. If x is a factor, the effect is measured with respect to each level of the factor and may be calculated as a discrete difference or as a continuous change, as measured by the derivative. `margins` calculates the discrete difference by default and calculates the derivative if the `continuous` option is specified.

elasticity and **semielasticity**. The elasticity of y with respect to x is $d(\ln y)/d(\ln x) = (x/y)\times(dy/dx)$, which is approximately equal to the proportional change in y for a proportional change in x.

The semielasticity of y with respect to x is either 1) $dy/d(\ln x) = x \times (dy/dx)$ or 2) $d(\ln y)/dx = (1/y) \times (dy/dx)$, which is approximately 1) the change in y for a proportional change in x or 2) the proportional change in y for a change in x.

`margins` uses model predictions or user-specified statistics in `expression()` in place of y above.

empty cell. An interaction of levels of two or more factor variables for which you have no data. For instance, you have sex interacted with group in your model, and in your data there are no females in group 1. Empty cells affect which margins can be estimated; see *Estimability of margins*.

estimability. Estimability concerns whether a margin can be uniquely estimated (identified); see *Estimability of margins*.

estimated marginal mean. This is one of the few terms that has the same meaning across authors. An estimated marginal mean is a margin assuming the levels of each factor covariate are equally likely (balanced), including interaction terms. This is obtained using `margins'` `asbalanced` option. In addition, there is an alternate definition of estimated marginal mean in which margins involving empty cells are redefined so that they become estimable. This is invoked by `margins'` `emptycells(reweight)` option. See *Balancing in the presence of empty cells*.

least-squares mean. Synonym for *estimated marginal mean*.

margin. A statistic calculated from predictions or other statistics of a previously fit model at fixed values of some covariates and averaging or otherwise integrating over the remaining covariates. The prediction or other statistic on which the margin is based is called the response.

If all the covariates are fixed, then the margin is called a conditional margin. If any covariates are left to vary, the margin is called a predictive margin.

In this documentation, we divide margins on the basis of whether the statistic is a response or a derivative of a response; see *Obtaining margins of responses* and *Obtaining margins of derivatives of responses*.

marginal effect and **average marginal effect**. The marginal effect of x is the *margin* of the *effect* of x. The term is popular with social scientists, and because of that, you might think the word marginal in marginal effect means derivative because of terms like marginal cost and marginal revenue. Marginal used in that way, however, refers to the derivative of revenue and the derivative of cost; it refers to the numerator, whereas marginal effect refers to the denominator. Moreover, *effect* is already a derivative or difference.

Some researchers interpret marginal in marginal effect to mean instantaneous, and thus a marginal effect is the instantaneous derivative rather than the discrete first difference, corresponding to `margins'` `continuous` option. Researchers who use marginal in this way refer to the discrete difference calculation of an effect as a partial effect.

Other researchers define marginal effect to be the margin when all covariates are held fixed and the average marginal effect when some covariates are not fixed.

out-of-sample prediction. Predictions made in one dataset using the results from a model fit on another. Sample here refers to the sample on which the model was fit, and out-of-sample refers to the dataset on which the predictions are made.

partial effect and **average partial effect**. Some authors restrict the term *marginal effect* to mean derivatives and use the term partial effect to denote discrete differences; see *marginal effect and average marginal effect*.

population marginal mean. The theoretical (true) value that is estimated by *estimated marginal mean*. We avoid this term because it can be confused with the concept of a population in survey statistics, with which the population marginal mean has no connection.

posting results, **posting margins**. A Stata concept having to do with storing the results from the `margins` command in `e()` so that those results can be used as if they were estimation results, thus allowing the subsequent use of postestimation commands, such as `test`, `testnl`, `lincom`, and `nlcom` (see [R] **test**, [R] **testnl**, [R] **lincom**, and [R] **nlcom**). This is achieved by specifying `margins`' `post` option. See *Example 10: Testing margins—contrasts of margins*.

predictive margin. A *margin* in which all the covariates are not fixed. When all covariates are fixed, it is called a *conditional margin*.

recycled prediction. A synonym for *predictive margin*.

response. A prediction or other statistic derived from combining the parameter estimates of a fitted model with data or specified values on covariates. Derivatives of responses are themselves responses. Responses are what we take *margins* of.

standardized margin. The margin calculated on data different from the data used to fit the model. The term standardized is usually reserved for situations in which the alternate population is a reference population, which may be real or artificial, and which is treated as fixed.

subpopulation. A subset of your sample that represents a subset of the population, such as the males in a sample of people. In survey contexts when it is desired to account for sampling of the covariates, standard errors for marginal statistics and effects need to account for both the population and the subpopulation. This is accomplished by specifying the `vce(unconditional)` option and one of the `subpop()` or `over()` options. In fact, the above is allowed even when your data are not `svyset` because `vce(unconditional)` implies that the sample represents a population.

Stored results

margins stores the following in r():

Scalars

r(N)	number of observations
r(N_sub)	subpopulation observations
r(N_clust)	number of clusters
r(N_psu)	number of sampled PSUs, survey data only
r(N_strata)	number of strata, survey data only
r(df_r)	variance degrees of freedom, survey data only
r(N_poststrata)	number of post strata, survey data only
r(k_predict)	number of predict() options
r(k_margins)	number of terms in *marginlist*
r(k_by)	number of subpopulations
r(k_at)	number of at() options
r(level)	confidence level of confidence intervals

Macros

r(cmd)	margins
r(cmdline)	command as typed
r(est_cmd)	e(cmd) from original estimation results
r(est_cmdline)	e(cmdline) from original estimation results
r(title)	title in output
r(subpop)	*subspec* from subpop()
r(model_vce)	*vcetype* from estimation command
r(model_vcetype)	Std. err. title from estimation command
r(vce)	*vcetype* specified in vce()
r(vcetype)	title used to label Std. err.
r(clustvar)	name of cluster variable
r(margins)	*marginlist*
r(predict#_opts)	the #th predict() option
r(predict#_label)	label from the #th predict() option
r(expression)	response expression
r(xvars)	*varlist* from dydx(), dyex(), eydx(), or eyex()
r(derivatives)	" ", "dy/dx", "dy/ex", "ey/dx", "ey/ex"
r(over)	*varlist* from over()
r(within)	*varlist* from within()
r(by)	union of r(over) and r(within) lists
r(by#)	interaction notation identifying the #th subpopulation
r(atstats#)	the #th at() specification
r(emptycells)	*empspec* from emptycells()
r(mcmethod)	*method* from mcompare()
r(mcadjustall)	adjustall or empty

Matrices

r(b)	estimates
r(V)	variance–covariance matrix of the estimates
r(Jacobian)	Jacobian matrix
r(_N)	sample size corresponding to each margin estimate
r(at)	matrix of values from the at() options
r(chainrule)	chain rule information from the fitted model
r(error)	margin estimability codes; 0 means estimable, 8 means not estimable
r(table)	matrix containing the margins with their standard errors, test statistics, p-values, and confidence intervals

margins with the post option also stores the following in e():

Scalars

e(N)	number of observations
e(N_sub)	subpopulation observations
e(N_clust)	number of clusters
e(N_psu)	number of sampled PSUs, survey data only
e(N_strata)	number of strata, survey data only
e(df_r)	variance degrees of freedom, survey data only
e(N_poststrata)	number of post strata, survey data only
e(k_predict)	number of predict() options
e(k_margins)	number of terms in *marginlist*
e(k_by)	number of subpopulations
e(k_at)	number of at() options

Macros

e(cmd)	margins
e(cmdline)	command as typed
e(est_cmd)	e(cmd) from original estimation results
e(est_cmdline)	e(cmdline) from original estimation results
e(wtype)	weight type
e(wexp)	weight expression
e(title)	title in estimation output
e(subpop)	*subspec* from subpop()
e(model_vce)	*vcetype* from estimation command
e(model_vcetype)	Std. err. title from estimation command
e(vce)	*vcetype* specified in vce()
e(vcetype)	title used to label Std. err.
e(clustvar)	name of cluster variable
e(properties)	b V, or just b if nose is specified
e(margins)	*marginlist*
e(asbalanced)	factor variables fvset as asbalanced
e(asobserved)	factor variables fvset as asobserved
e(predict#_opts)	the #th predict() option
e(predict#_label)	label from the #th predict() option
e(expression)	prediction expression
e(xvars)	*varlist* from dydx(), dyex(), eydx(), or eyex()
e(derivatives)	"", "dy/dx", "dy/ex", "ey/dx", "ey/ex"
e(over)	*varlist* from over()
e(within)	*varlist* from within()
e(by)	union of r(over) and r(within) lists
e(by#)	interaction notation identifying the #th subpopulation
e(atstats#)	the #th at() specification
e(emptycells)	*empspec* from emptycells()

Matrices

e(b)	estimates
e(V)	variance–covariance matrix of the estimates
e(Jacobian)	Jacobian matrix
e(_N)	sample size corresponding to each margin estimate
e(error)	error code corresponding to e(b)
e(at)	matrix of values from the at() options
e(chainrule)	chain rule information from the fitted model

Functions

e(sample)	marks estimation sample

Methods and formulas

Margins are statistics calculated from predictions of a previously fit model at fixed values of some covariates and averaging or otherwise integrating over the remaining covariates. There are many names for the different statistics that margins can compute: estimates marginal means (see Searle,

Speed, and Milliken [1980]), predictive margins (see Graubard and Korn [2004]), marginal effects (see Greene [2018]), and average marginal/partial effects (see Wooldridge [2010] and Bartus [2005]).

Methods and formulas are presented under the following headings:

Notation

Let $\boldsymbol{\theta}$ be the vector of parameters in the current model fit, let $\mathbf{z}$ be a vector of covariate values, and let $f(\mathbf{z}, \boldsymbol{\theta})$ be a scalar-valued function returning the value of the predictions of interest. The following table illustrates the parameters and default prediction for several of Stata's estimation commands.

Command	$\boldsymbol{\theta}$	$\mathbf{z}$	$f(\mathbf{z}, \boldsymbol{\theta})$
`regress`	$\boldsymbol{\beta}$	$\mathbf{x}$	$\mathbf{x}\boldsymbol{\beta}$
`cloglog`	$\boldsymbol{\beta}$	$\mathbf{x}$	$1 - e^{-e^{x\beta}}$
`logit`	$\boldsymbol{\beta}$	$\mathbf{x}$	$1/(1 + e^{-x\beta})$
`poisson`	$\boldsymbol{\beta}$	$\mathbf{x}$	$e^{x\beta}$
`probit`	$\boldsymbol{\beta}$	$\mathbf{x}$	$\Phi(\mathbf{x}\boldsymbol{\beta})$
`biprobit`	$\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \rho$	$\mathbf{x}_1, \mathbf{x}_2$	$\Phi_2(\mathbf{x}_1\boldsymbol{\beta}_1, \mathbf{x}_2\boldsymbol{\beta}_2, \rho)$
`mlogit`	$\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \ldots, \boldsymbol{\beta}_k$	$\mathbf{x}$	$e^{-x\beta_1}/(\sum_i e^{-x\beta_i})$
`nbreg`	$\boldsymbol{\beta}$, $\ln\alpha$	$\mathbf{x}$	$e^{x\beta}$

$\Phi()$ and $\Phi_2()$ are cumulative distribution functions: $\Phi()$ for the standard normal distribution and $\Phi_2()$ for the standard bivariate normal distribution.

`margins` computes estimates of

$$p(\boldsymbol{\theta}) = \frac{1}{M_{S_\mathrm{p}}} \sum_{j=1}^{M} \delta_j(S_\mathrm{p}) f(\mathbf{z}_j, \boldsymbol{\theta})$$

where $\delta_j(S_\mathrm{p})$ identifies elements within the subpopulation S_p (for the prediction of interest),

$$\delta_j(S_\mathrm{p}) = \begin{cases} 1, & j \in S_\mathrm{p} \\ 0, & j \notin S_\mathrm{p} \end{cases}$$

M_{S_p} is the subpopulation size,

$$M_{S_\mathrm{p}} = \sum_{j=1}^{M} \delta_j(S_\mathrm{p})$$

and M is the population size.

Let $\widehat{\boldsymbol{\theta}}$ be the vector of parameter estimates. Then, `margins` estimates $p(\boldsymbol{\theta})$ via

$$\widehat{p} = \frac{1}{w.} \sum_{j=1}^{N} \delta_j(S_{\mathrm{p}}) w_j f(\mathbf{z}_j, \widehat{\boldsymbol{\theta}})$$

where

$$w. = \sum_{j=1}^{N} \delta_j(S_{\mathrm{p}}) w_j$$

$\delta_j(S_{\mathrm{p}})$ indicates whether observation j is in subpopulation S_{p}, w_j is the weight for the jth observation, and N is the sample size.

Marginal effects

`margins` also computes marginal and partial effects. For the marginal effect of continuous covariate x, `margins` computes

$$\widehat{p} = \frac{1}{w.} \sum_{j=1}^{N} \delta_j(S_{\mathrm{p}}) w_j h(\mathbf{z}_j, \widehat{\boldsymbol{\theta}})$$

where

$$h(\mathbf{z}, \boldsymbol{\theta}) = \frac{\partial f(\mathbf{z}, \boldsymbol{\theta})}{\partial x}$$

The marginal effect for level k of factor variable A is the simple contrast (also known as difference) comparing its margin with the margin at the base level.

$$h(\mathbf{z}, \boldsymbol{\theta}) = f(\mathbf{z}, \boldsymbol{\theta} | A = k) - f(\mathbf{z}, \boldsymbol{\theta} | A = \text{base})$$

Elasticities and semielasticities given by `eyex()` and `dyex()` are only available for continuous covariates. Elasticities are computed using

$$h(\mathbf{z}, \boldsymbol{\theta}) = \frac{\partial \ln\{f(\mathbf{z}, \boldsymbol{\theta})\}}{\partial \ln(x)}$$

and semielasticities, `dyex()`, using

$$h(\mathbf{z}, \boldsymbol{\theta}) = \frac{\partial f(\mathbf{z}, \boldsymbol{\theta})}{\partial \ln(x)}$$

Semielasticities of the form `eydx()` for continuous covariates are computed using

$$h(\mathbf{z}, \boldsymbol{\theta}) = \frac{\partial \ln\{f(\mathbf{z}, \boldsymbol{\theta})\}}{\partial x}$$

and for discrete covariates using

$$h(\mathbf{z}, \boldsymbol{\theta}) = \ln\{f(\mathbf{z}, \boldsymbol{\theta} | A = k)\} - \ln\{f(\mathbf{z}, \boldsymbol{\theta} | A = \text{base})\}$$

Fixing covariates and balancing factors

`margins` controls the values in each **z** vector through the *marginlist*, the `at()` option, the `atmeans` option, and the `asbalanced` and `emptycells()` options. Suppose **z** is composed of the elements from the equation specification

```
A##B x
```

where `A` is a factor variable with a levels, `B` is a factor variable with b levels, and `x` is a continuous covariate. To simplify the notation for this discussion, assume the levels of `A` and `B` start with 1 and are contiguous. Then

$$\mathbf{z} = (A_1, \ldots, A_a, B_1, \ldots, B_b, A_1B_1, A_1B_2, \ldots, A_aB_b, x, 1)$$

where A_i, B_j, and A_iB_j represent the indicator values for the factor variables `A` and `B` and the interaction `A#B`.

When factor `A` is in the *marginlist*, `margins` replaces `A` with i and then computes the mean of the subsequent prediction, for $i = 1, \ldots, a$. When the interaction term `A#B` is in the *marginlist*, `margins` replaces `A` with i and `B` with j, and then computes the mean of the subsequent prediction, for all combinations of $i = 1, \ldots, a$ and $j = 1, \ldots, b$.

The `at()` option sets model covariates to fixed values. For example, `at(x=15)` causes `margins` to temporarily set x to 15 for each observation in the dataset before computing any predictions. Similarly, `at((median) x)` causes `margins` to temporarily set x to the median of `x` using the current dataset.

When factor variable `A` is specified as `asbalanced`, `margins` sets each A_i to $1/a$. Thus, each **z** vector will look like

$$\mathbf{z} = (1/a, \ldots, 1/a, B_1, \ldots, B_b, B_1/a, B_2/a, \ldots, B_b/a, x, 1)$$

If `B` is also specified as `asbalanced`, then each B_j is set to $1/b$, and each **z** vector will look like

$$\mathbf{z} = (1/a, \ldots, 1/a, 1/b, \ldots, 1/b, 1/ab, 1/ab, \ldots, 1/ab, x, 1)$$

If `emptycells(reweight)` is also specified, then `margins` uses a different balancing weight for each element of **z**, depending on how many empty cells the element is associated with. Let δ_{ij} indicate that the ijth cell of `A#B` was observed in the estimation sample.

$$\delta_{ij} = \begin{cases} 0, & \texttt{A} = i \text{ and } \texttt{B} = j \text{ was an empty cell} \\ 1, & \text{otherwise} \end{cases}$$

For the grand margin, the affected elements of **z** and their corresponding balancing weights are

$$A_i = \frac{\sum_j \delta_{ij}}{\sum_k \sum_j \delta_{kj}}$$

$$B_j = \frac{\sum_i \delta_{ij}}{\sum_i \sum_k \delta_{ik}}$$

$$A_i B_j = \frac{\delta_{ij}}{\sum_k \sum_l \delta_{kl}}$$

For the jth margin of `B`, the affected elements of $\mathbf{z}$ and their corresponding balancing weights are

$$A_i = \frac{\delta_{ij}}{\sum_k \delta_{kj}}$$

$$B_l = \begin{cases} 1, & \text{if } l = j \text{ and not all } \delta_{ij} \text{ are zero} \\ 0, & \text{otherwise} \end{cases}$$

$$A_i B_l = \frac{\delta_{il}}{\sum_k \delta_{kl}} B_l$$

Estimable functions

The fundamental idea behind estimable functions is clearly defined in the statistical literature for linear models; see Searle and Gruber (2017). Assume that we are working with the following linear model:

$$\mathbf{y} = \mathbf{X}\mathbf{b} + \mathbf{e}$$

where $\mathbf{y}$ is an $N \times 1$ vector of responses, $\mathbf{X}$ is an $N \times p$ matrix of covariate values, $\mathbf{b}$ is a $p \times 1$ vector of coefficients, and $\mathbf{e}$ is a vector of random errors. Assuming a constant variance for the random errors, the normal equations for the least-squares estimator, $\widehat{\mathbf{b}}$, are

$$\mathbf{X}'\mathbf{X}\widehat{\mathbf{b}} = \mathbf{X}'\mathbf{y}$$

When $\mathbf{X}$ is not of full column rank, we will need a generalized inverse (g-inverse) of $\mathbf{X}'\mathbf{X}$ to solve for $\widehat{\mathbf{b}}$. Let $\mathbf{G}$ be a g-inverse of $\mathbf{X}'\mathbf{X}$.

Searle and Gruber (2017) defines a linear function of the parameters as *estimable* if it is identically equal to some linear function of the expected values of the $\mathbf{y}$ vector. Let $\mathbf{H} = \mathbf{G}\mathbf{X}'\mathbf{X}$. Then this definition simplifies to the following rule:

$$\mathbf{z}\mathbf{b} \text{ is estimable if } \mathbf{z} = \mathbf{z}\mathbf{H}$$

`margins` generalizes this to nonlinear functions by assuming the prediction function $f(\mathbf{z}, \boldsymbol{\theta})$ is a function of one or more of the linear predictions from the equations in the model that $\boldsymbol{\theta}$ represents.

$$f(\mathbf{z}, \boldsymbol{\theta}) = h(\mathbf{z}_1\boldsymbol{\beta}_1, \mathbf{z}_2\boldsymbol{\beta}_2, \ldots, \mathbf{z}_k\boldsymbol{\beta}_k)$$

$\mathbf{z}_i\boldsymbol{\beta}_i$ is considered estimable if $\mathbf{z}_i = \mathbf{z}_i\mathbf{H}_i$, where $\mathbf{H}_i = \mathbf{G}_i\mathbf{X}_i'\mathbf{X}_i$, $\mathbf{G}_i$ is a g-inverse for $\mathbf{X}_i'\mathbf{X}_i$, and $\mathbf{X}_i$ is the matrix of covariates from the ith equation of the fitted model. `margins` considers $p(\boldsymbol{\theta})$ to be estimable if every $\mathbf{z}_i\boldsymbol{\beta}_i$ is estimable.

Standard errors conditional on the covariates

By default, `margins` uses the delta method to estimate the variance of $\widehat{p}$.

$$\widehat{\mathrm{Var}}(\widehat{p}\,|\,\mathbf{z}) = \mathbf{v}'\mathbf{V}\mathbf{v}$$

where $\mathbf{V}$ is a variance estimate for $\widehat{\boldsymbol{\theta}}$ and

$$\mathbf{v} = \left.\frac{\partial \widehat{p}}{\partial \boldsymbol{\theta}}\right|_{\theta=\widehat{\theta}}$$

This variance estimate is conditional on the $\mathbf{z}$ vectors used to compute the marginalized predictions.

Unconditional standard errors

`margins` with the `vce(unconditional)` option uses linearization to estimate the unconditional variance of $\widehat{\boldsymbol{\theta}}$. Linearization uses the variance estimator for the total of a score variable for $\widehat{p}$ as an approximate estimator for $\mathrm{Var}(\widehat{p})$; see [SVY] **Variance estimation**. `margins` requires that the model was fit using some form of linearized variance estimator and that `predict, scores` computes the appropriate score values for the linearized variance estimator.

The score for $\widehat{p}$ from the jth observation is given by

$$s_j = \frac{\partial \widehat{p}}{\partial w_j} = -\frac{\delta_j(S_{\mathrm{p}})}{w_{\cdot}}\widehat{p} + \frac{\delta_j(S_{\mathrm{p}})}{w_{\cdot}} f(\mathbf{z}_j, \widehat{\boldsymbol{\theta}}) + \frac{1}{w_{\cdot}}\sum_{i=1}^{N} \delta_i(S_{\mathrm{p}}) w_i \frac{\partial f(\mathbf{z}_i, \widehat{\boldsymbol{\theta}})}{\partial w_j}$$

The remaining partial derivative can be decomposed using the chain rule.

$$\frac{\partial f(\mathbf{z}_i, \widehat{\boldsymbol{\theta}})}{\partial w_j} = \left(\left.\frac{\partial f(\mathbf{z}_i, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right|_{\theta=\widehat{\theta}}\right)\left(\frac{\partial \widehat{\boldsymbol{\theta}}}{\partial w_j}\right)'$$

This is the inner product of two vectors, the second of which is not a function of the i index. Thus, the score is

$$s_j = -\frac{\delta_j(S_{\mathrm{p}})}{w_{\cdot}}\widehat{p} + \frac{\delta_j(S_{\mathrm{p}})}{w_{\cdot}} f(\mathbf{z}_j, \widehat{\boldsymbol{\theta}}) + \left(\left.\frac{\partial \widehat{p}}{\partial \boldsymbol{\theta}}\right|_{\theta=\widehat{\theta}}\right)\left(\frac{\partial \widehat{\boldsymbol{\theta}}}{\partial w_j}\right)'$$

If $\widehat{\boldsymbol{\theta}}$ was derived from a system of equations (such as in linear regression or maximum likelihood estimation), then $\widehat{\boldsymbol{\theta}}$ is the solution to

$$\mathbf{G}(\boldsymbol{\theta}) = \sum_{j=1}^{N} \delta_j(S_{\mathrm{m}}) w_j \mathbf{g}(\boldsymbol{\theta}, \mathbf{y}_j, \mathbf{x}_j) = \mathbf{0}$$

where S_{m} identifies the subpopulation used to fit the model, $\mathbf{g}()$ is the model's gradient function, and $\mathbf{y}_j$ and $\mathbf{x}_j$ are the values of the dependent and independent variables for the jth observation. We can use linearization to derive a first-order approximation for $\partial\widehat{\boldsymbol{\theta}}/\partial w_j$.

$$\mathbf{G}(\widehat{\boldsymbol{\theta}}) \approx \mathbf{G}(\boldsymbol{\theta}_0) + \left.\frac{\partial \mathbf{G}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right|_{\theta=\theta_0} (\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta}_0)$$

Let $\mathbf{H}$ be the Hessian matrix

$$\mathbf{H} = \left.\frac{\partial \mathbf{G}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right|_{\theta=\theta_0}$$

Then

$$\widehat{\boldsymbol{\theta}} \approx \boldsymbol{\theta}_0 + (-\mathbf{H})^{-1}\mathbf{G}(\boldsymbol{\theta}_0)$$

and

$$\frac{\partial \widehat{\boldsymbol{\theta}}}{\partial w_j} \approx (-\mathbf{H})^{-1} \left.\frac{\partial \mathbf{G}(\boldsymbol{\theta})}{\partial w_j}\right|_{\theta=\widehat{\theta}} = (-\mathbf{H})^{-1}\delta_j(S_{\mathrm{m}})\mathbf{g}(\widehat{\boldsymbol{\theta}}, \mathbf{y}_j, \mathbf{x}_j)$$

The computed value of the score for $\widehat{p}$ for the jth observation is

$$s_j = \mathbf{v}'\mathbf{u}_j$$

where

$$\mathbf{v} = \begin{bmatrix} -\frac{\widehat{p}}{w.} \\ \frac{1}{w.} \\ \frac{\partial \widehat{p}}{\partial \widehat{\boldsymbol{\theta}}}(-\mathbf{H})^{-1} \end{bmatrix}$$

and

$$\mathbf{u}_j = \begin{bmatrix} \delta_j(S_{\mathrm{p}}) \\ \delta_j(S_{\mathrm{p}})f(\mathbf{z}_j, \widehat{\boldsymbol{\theta}}) \\ \delta_j(S_{\mathrm{m}})\mathbf{g}(\widehat{\boldsymbol{\theta}}, \mathbf{y}_j, \mathbf{x}_j) \end{bmatrix}$$

Thus, the variance estimate for $\widehat{p}$ is

$$\widehat{\mathrm{Var}}(\widehat{p}) = \mathbf{v}'\widehat{\mathrm{Var}}(\widehat{\mathbf{U}})\mathbf{v}$$

where

$$\widehat{\mathbf{U}} = \sum_{j=1}^{N} w_j \mathbf{u}_j$$

`margins` uses the model-based variance estimates for $(-\mathbf{H})^{-1}$ and the scores from `predict` for $\mathbf{g}(\widehat{\boldsymbol{\theta}}, \mathbf{y}_j, \mathbf{x}_j)$.

References

Bartus, T. 2005. Estimation of marginal effects using margeff. *Stata Journal* 5: 309–329.

Baum, C. F. 2010. Stata tip 88: Efficiently evaluating elasticities with the margins command. *Stata Journal* 10: 309–312.

Bruun, N. H. 2019. Visualizing effect modification on contrasts. *Stata Journal* 19: 566–580.

Buis, M. L. 2010. Stata tip 87: Interpretation of interactions in nonlinear models. *Stata Journal* 10: 305–308.

Chang, I.-M., R. Gelman, and M. Pagano. 1982. Corrected group prognostic curves and summary statistics. *Journal of Chronic Diseases* 35: 669–674. https://doi.org/10.1016/0021-9681(82)90019-4.

Cummings, P. 2011. Estimating adjusted risk ratios for matched and unmatched data: An update. *Stata Journal* 11: 290–298.

Drukker, D. M. 2016a. Doctors versus policy analysts: Estimating the effect of interest. *The Stata Blog: Not Elsewhere Classified.* http://blog.stata.com/2016/07/19/doctors-versus-policy-analysts-estimating-the-effect-of-interest/.

——. 2016b. Probability differences and odds ratios measure conditional-on-covariate effects and population-parameter effects. *The Stata Blog: Not Elsewhere Classified.* http://blog.stata.com/2016/07/26/probability-differences-and-odds-ratios-measure-conditional-on-covariate-effects-and-population-parameter-effects/.

Falcaro, M., R. B. Newson, and P. D. Sasieni. 2022. Stata tip 146: Using margins after a Poisson regression model to estimate the number of events prevented by an intervention. *Stata Journal* 22: 224–230.

Gould, W. W. 1996. crc43: Wald test of nonlinear hypotheses after model estimation. *Stata Technical Bulletin* 29: 2–4. Reprinted in *Stata Technical Bulletin Reprints*, vol. 5, pp. 15–18. College Station, TX: Stata Press.

Graubard, B. I., and E. L. Korn. 2004. Predictive margins with survey data. *Biometrics* 55: 652–659. https://doi.org/10.1111/j.0006-341X.1999.00652.x.

Greene, W. H. 2018. *Econometric Analysis*. 8th ed. New York: Pearson.

Korn, E. L., and B. I. Graubard. 1999. *Analysis of Health Surveys*. New York: Wiley.

Lane, P. W., and J. A. Nelder. 1982. Analysis of covariance and standardization as instances of prediction. *Biometrics* 38: 613–621. https://doi.org/10.2307/2530043.

Lindsey, C. 2015a. Using mlexp to estimate endogenous treatment effects in a heteroskedastic probit model. *The Stata Blog: Not Elsewhere Classified.* http://blog.stata.com/2015/12/10/using-mlexp-to-estimate-endogenous-treatment-effects-in-a-heteroskedastic-probit-model/.

——. 2015b. Using mlexp to estimate endogenous treatment effects in a probit model. *The Stata Blog: Not Elsewhere Classified.* http://blog.stata.com/2015/11/05/using-mlexp-to-estimate-endogenous-treatment-effects-in-a-probit-model/.

——. 2016. Estimating covariate effects after gmm. *The Stata Blog: Not Elsewhere Classified.* http://blog.stata.com/2016/10/04/estimating-covariate-effects-after-gmm/.

Lindsey, C., and E. Pinzon. 2016. Multiple equation models: Estimation and marginal effects using gsem. *The Stata Blog: Not Elsewhere Classified.* http://blog.stata.com/2016/06/07/multiple-equation-models-estimation-and-marginal-effects-using-gsem/.

MacDonald, K. 2018. Exploring results of nonparametric regression models. *The Stata Blog: Not Elsewhere Classified.* https://blog.stata.com/2018/06/18/exploring-results-of-nonparametric-regression-models/.

Mitchell, M. N. 2015. *Stata for the Behavioral Sciences*. College Station, TX: Stata Press.

——. 2021. *Interpreting and Visualizing Regression Models Using Stata*. 2nd ed. College Station, TX: Stata Press.

Newson, R. B. 2013. Attributable and unattributable risks and fractions and other scenario comparisons. *Stata Journal* 13: 672–698.

Phillips, P. C. B., and J. Y. Park. 1988. On the formulation of Wald tests of nonlinear restrictions. *Econometrica* 56: 1065–1083. https://doi.org/10.2307/1911359.

Pinzon, E. 2016a. Effects of nonlinear models with interactions of discrete and continuous variables: Estimating, graphing, and interpreting. *The Stata Blog: Not Elsewhere Classified.* http://blog.stata.com/2016/07/12/effects-for-nonlinear-models-with-interactions-of-discrete-and-continuous-variables-estimating-graphing-and-interpreting/.

——. 2016b. probit or logit: ladies and gentlemen, pick your weapon. *The Stata Blog: Not Elsewhere Classified.* http://blog.stata.com/2016/01/07/probit-or-logit-ladies-and-gentlemen-pick-your-weapon/.

——. 2016c. regress, probit, or logit? *The Stata Blog: Not Elsewhere Classified.* http://blog.stata.com/2016/01/14/regress-probit-or-logit/.

Radean, M. 2023. ginteff: A generalized command for computing interaction effects. *Stata Journal* 23: 301–335.

Rios-Avila, F. 2021. Estimation of marginal effects for models with alternative variable transformations. *Stata Journal* 21: 81–96.

Searle, S. R. 1997. *Linear Models for Unbalanced Data*. New York: Wiley.

Searle, S. R., and M. H. J. Gruber. 2017. *Linear Models*. 2nd ed. Hoboken, NJ: Wiley.

Searle, S. R., F. M. Speed, and G. A. Milliken. 1980. Population marginal means in the linear model: An alternative to least squares means. *American Statistician* 34: 216–221. https://doi.org/10.2307/2684063.

Terza, J. V. 2017a. Two-stage residual inclusion estimation: A practitioners guide to Stata implementation. *Stata Journal* 17: 916–938.

——. 2017b. Causal effect estimation and inference using Stata. *Stata Journal* 17: 939–961.

Williams, R. 2012. Using the margins command to estimate and interpret adjusted predictions and marginal effects. *Stata Journal* 12: 308–331.

Wooldridge, J. M. 2010. *Econometric Analysis of Cross Section and Panel Data*. 2nd ed. Cambridge, MA: MIT Press.

Also see

[R] **contrast** — Contrasts and linear hypothesis tests after estimation

[R] **margins, contrast** — Contrasts of margins

[R] **margins, pwcompare** — Pairwise comparisons of margins

[R] **margins postestimation** — Postestimation tools for margins

[R] **marginsplot** — Graph results from margins (profile plots, etc.)

[R] **lincom** — Linear combinations of parameters

[R] **nlcom** — Nonlinear combinations of parameters

[R] **predict** — Obtain predictions, residuals, etc., after estimation

[R] **predictnl** — Obtain nonlinear predictions, standard errors, etc., after estimation

[U] **20 Estimation and postestimation commands**