The Stata Journal (*yyyy*) *vv*, Number *ii*, pp. 1–24

# Attributable and unattributable risks and fractions and other scenario comparisons

Roger B. Newson
National Heart and Lung Institute, Imperial College London
London, United Kingdom
r.newson@imperial.ac.uk

**Abstract.**

Scenarios can be defined as alternative versions of the same dataset, with the same variables but different observations and/or values. Applied scientists frequently want to predict how much good an intervention will do, by comparing outcomes from the same model betwen different scenarios. Alternatively, they may want to compare outcomes between different models applied to the same scenario, as when standardizing statistics from different subpopulations to a common gender and age distribution. Standard Stata tools for scenario means and comparisons are `margins` and `pwcompare`. A suite of packages is presented for estimating scenario means and comparisons using `margins`, together with Normalizing and variance–stabilizing transformations, implemented using `nlcom`. `margprev` estimates marginal prevalences, `marglmean` estimates marginal arithmetic means, `regpar` estimates the difference between 2 marginal prevalences (the population attributable risk or PAR), `punaf` estimates the ratio between 2 marginal arithmetic means (the population unattributable fraction or PUF), and `punafcc` estimates a marginal mean between–scenario risk or hazard ratio for case–control or survival data, also known as a PUF. The PUF and its confidence limits are subtracted from 1 to estimate the population attributable fraction (PAF). Formulas and examples are presented, including an example from the Global Allergy and Asthma European Network ($GA^2LEN$).

**Keywords:** st0001, `margprev`, `marglmean`, `regpar`, `punaf`, `punafcc`, `margins`, `nlcom`, population, unattributable, attributable, risk, fraction, PAR, PAF, PUF, scenario, comparison, standardization

## 1 Introduction

Applied scientists, especially in the public health sector, usually want to know how much good they can do. In particular, they might want to estimate, from the available data, how much reduction we would see in a disease rate, if everybody stopped smoking, or if all children received a proposed vaccine. Alternatively, they might compare disease rates between different subpopulations, and discover heterogeneity, and wonder whether that heterogeneity is caused by confounding factors, such as differences in the age distribution between different subpoopulations. After all, if Subpopulation $A$ has a higher rate of a particular cancer than Subpopulation $B$, then this might be because of something in the environment of Subpopulation $A$, to which Subpopulation $B$ is not exposed, or

 

it might be because Subpopulation $A$ is mostly older than Subpopulation $B$. If we could eliminate the second possibility by standardizing the disease rates to a standard age distribution, then we might have evidence for the first possibility. In both cases, we are comparing scenarios. In the first case, we are comparing 2 different scenarios, using data from the same sample. In the second case, we are comparing the same scenario, using data from 2 different samples, one from Subpopulation $A$, and one from Subpopulation $B$.

In statistics, scenarios can be defined as alternative versions of the same data matrix, with equivalent columns (variables), but with different rows (observations). Different scenarios have a one–to–one correspondence between the columns, so that equivalent columns have the same variable names. However, different scenarios may or may not have a one–to–one correspondence between equivalent rows. If we use regression methods, then we might want to estimate scenario means of an outcome variable $Y$, under different scenarios defined by specifying values for particular $X$–variables. The $X$–variables that vary between scenarios are known as exposures, and the other $X$–variables, which are invariant between scenarios, are known as concomitant variables.

A seminal reference for scenario means and comparisons in generalized linear models (GLMs) is Lane and Nelder (1982). However, an important case is the estimation of population attributable fractions after fitting a logistic regression model, which is given, with different formulas for cohort studies and for case–control studies, by Greenland and Drescher (1993). These formulas were implemented in Stata by Brady (1998), who introduced the Stata Version 5 package `aflogit`. This package is still downloadable using the command `findit aflogit`, although it does not support factor variable lists, and the Stata Version 5 code sometimes has problems with the long variable names used in subsequent Stata versions. Another special case of a scenario comparison is the population attributable risk (PAR), defined in Gordis (2000).

In Stata Version 11, a new command `margins` was added (see [R] **margins**). `margins` inputs a set of estimation results, and a set of $X$–variables, and outputs scenario means for expressions involving predicted $Y$–values under one or more scenarios. These scenario means are estimated with covariance matrices, so the user can calculate confidence intervals for them. In Stata Version 12, the commands `contrast` and `pwcompare` were added (see [R] **contrast** and [R] **pwcompare**), together with the `pwcompare` and `pwcompare()` options for `margins` (see [R] **margins, pwcompare)**. These can be used to calculate confidence intervals for differences between scenario means. However, users frequently want to estimate scenario means, and their differences and ratios, using Normalizing and variance–stabilizing transformations to generate confidence limits in which the user can have confidence. This can be done using `nlcom` (see [R] **nlcom**).

This article introduces a suite of programs, which call `margins` and `nlcom` to calculate scenario prevalences and means, their differences, their ratios, and other comparison statistics. These statistics are known as marginal means, marginal prevalences, and attributable and unattributable risks and fractions. Section 2 describes the packages. Section 3 describes the methods and formulas used. Finally, Section 4 gives practical examples of the use of these packages.

# 2 The packages margprev, marglmean, regpar, punaf and punafcc

## 2.1 Syntax

margprev [ *if* ] [ *in* ] [ *weight* ] [ , atspec(*atspec*) subpop(*subspec*) predict(*pred_opt*) vce(*vcespec*) noesample force iterate(#) eform level(#) post ]

marglmean [ *if* ] [ *in* ] [ *weight* ] [ , atspec(*atspec*) subpop(*subspec*) predict(*pred_opt*) vce(*vcespec*) noesample force iterate(#) eform level(#) post ]

regpar [ *if* ] [ *in* ] [ *weight* ] [ , atspec(*atspec*) atzero(*atspec0*) subpop(*subspec*) predict(*pred_opt*) vce(*vcespec*) noesample force iterate(#) level(#) post ]

punaf [ *if* ] [ *in* ] [ *weight* ] [ , atspec(*atspec*) atzero(*atspec0*) subpop(*subspec*) predict(*pred_opt*) vce(*vcespec*) noesample force iterate(#) eform level(#) post ]

punafcc [ *if* ] [ *in* ] [ *weight* ] [ , atspec(*atspec*) subpop(*subspec*) vce(*vcespec*) noesample force iterate(#) eform level(#) post ]

where *atspec* and *atspec0* are at–specifications recognized by the `at()` option of `margins`, *subspec* is a subpopulation specification of the form recognized by the `subpop()` option of `margins`, and *vcespec* is a variance–covariance specification of the form recognized by `margins`, and must have one of the values

`delta` | `unconditional`

`fweight`s, `aweight`s, `pweight`s and `iweight`s are allowed. They are handled as by `margins`.

## 2.2 Description

The packages `margprev`, `marglmean`, `regpar`, `punaf` and `punafcc` are for use after the parameters of a regression model have been estimated, using an estimation command. They estimate a range of scenario prevalences, means and mean risk ratios, and their between–scenario comparisons (differences and ratios). These are estimated with con-

Table 1: List of packages with parameters estimated and transformations used.

| *Package* | *Estimated parameters* | *Transformations* |
|---|---|---|
| `margprev` | 1 marginal prevalence | Logit |
| `marglmean` | 1 marginal arithmetic mean | Log |
| `regpar` | 2 marginal prevalences and their difference (PAR) | Logit, Fisher's $z$ |
| `punaf` | 2 marginal arithmetic means and their ratio (PUF) | Log |
| `punafcc` | 1 mean between–scenario risk or hazard ratio (PUF) | Log |

fidence limits, derived using Normalizing and variance–stabilizing transformations to estimate the transformed parameter(s) and their dispersion matrix. A difference between 2 scenario prevalences is known as a population attributable risk (PAR), and a ratio between 2 scenario arithmetic means, or a mean between–scenario risk ratio or hazard ratio, is known as a population unattributable fraction (PUF). When a PUF is estimated, a confidence interval is also calculated, using end–point transformation, for the population attributable fraction (PAF), which is derived by subtracting the PUF from 1. Table 1 lists the 5 packages, the parameters estimated, and the transformations used.

## 2.3 Options

`atspec(`*atspec*`)` is an at–specification, allowed as a value of the `at()` option of `margins` (see [R] **margins**). This at–specification must specify a single scenario ("Scenario 1"), defined as a fantasy world in which a subset of the predictor variables in the model are set to values, which may be different from their values in the real world. In the case of `punafcc`, which is intended for use with case–control or survival data, the at–specification is restricted, and may set variables only to values (not to statistics). If `atspec()` is not specified, then its default value is `atspec((asobserved) _all)`, implying that Scenario 1 is the baseline scenario, represented by the predictor values actually present in the dataset currently in memory.

`atzero(`*atspec0*`)` is available for `regpar` and `punaf` only. It specifies an at–specification, allowed as a value of the `at()` option of `margins`. This at–specification must specify a single baseline scenario ("Scenario 0"), defined as an alternative fantasy world, in which a subset of predictors in the model are set to the values specified by *atspec0*. Scenario 0 will then be compared to the "Scenario 1" specified by the `atspec()` option. If `atzero()` is not specified, then its default value is `atzero((asobserved) _all)`, implying that Scenario 0 is the baseline scenario, represented by the predictor values actually present in the dataset currently in memory.

`subpop(`*subspec*`)`, `predict(`*pred_opt*`)`, `vce(`*vcespec*`)`, `noesample` and `force` function

as the options of the same names for `margins`. `subpop()` specifies a subpopulation, `predict()` specifies a predict option, `vce()` specifies the formula used for calculating the dispersion matrix of the estimated parameters, `noesample` specifies that the estimated statistics will not be restricted to the current estimation sample, and `force` specifies that the scenario means will still be estimated, even if there are potential problems detectable by `margins`. The `predict()` option is not available at present for `punafcc`, but it enables the use of the other 4 packages after a multiple–equation command. For instance, after `mlogit`, the option `predict(outcome(2))` allows scenario prevalences to be estimated and/or compared for the second value of a multinomial outcome. (See [R] **mlogit**.)

`iterate(#)` has the same form and function as the option of the same name for `nlcom` (see [R] **nlcom**). It specifies the number of iterations used by `nlcom` to find the optimal step size to calculate the numerical derivatives of the transformed scenario means and comparisons, with respect to the original scenario means calculated by `margins`.

`eform` specifies that the command will display an estimate, $p$–value, and confidence limits instead of the log estimate; see the help files for `margprev`, `marglmean`, `punaf`, and `punafcc` for complete descriptions.

`level(#)` specifies the percentage confidence level to be used in calculating the confidence intervals. If it is not specified, then it is taken from the current value of the c–class value `c(level)`, which is usually 95.

`post` specifies that the command will post in `e()` the estimation results for estimating the transformed scenario means and any comparisons (differences or ratios). If `post` is not specified, then any existing estimation results are left in `e()`. Note that the estimation results posted are for the transformed parameters, and not for the parameters themselves. This is done because the estimation results are intended to define symmetric confidence intervals for the transformed parameters, which can be back–transformed to define asymmetric confidence intervals for the untransformed parameters, and for the PAR in the case of `punaf` and `punafcc`.

## 2.4 Saved results

`margprev`, `marglmean`, `regpar`, `punaf` and `punafcc` save the following results in `r()`:

Scalars

| | |
|---|---|
| `r(rank)` | Rank of `r(V)` |
| `r(N)` | number of observations |
| `r(N_sub)` | subpopulation observations |
| `r(N_clust)` | number of clusters |
| `r(N_psu)` | number of samples 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_margins)` | number of terms in *marginlist* |
| `r(k_by)` | number of subpopulations |
| `r(k_at)` | number of `at()` options (always 1 or 2) |
| `r(level)` | confidence level |

Macros

| | |
|---|---|
| `r(atzero)` | `atzero()` option (`regpar` and `punaf` only) |
| `r(atspec)` | `atspec()` option |

Matrices

| | |
|---|---|
| `r(cimat)` | matrix of asymmetric confidence intervals (not saved by `marglmean`) |
| `r(b)` | vector of estimated transformed parameters |
| `r(V)` | dispersion matrix for transformed estimated parameters |

The matrix `r(cimat)` is not saved by `marglmean`. It contains asymmetric confidence intervals (one per row) for the untransformed marginal prevalence in the case of `margprev`, for the untransformed marginal prevalences and their untransformed difference (the PAR) in the case of `regpar`, and for the population attributable fraction (PAF, equal to $1 - \text{PUF}$) in the case of `punaf` and `punafcc`. The matrices `r(b)` and `r(V)` contain the estimate and dispersion matrix, respectively, for the transformed parameters, as indicated in Table 1.

If `post` is specified, then `margprev`, `marglmean`, `regpar`, `punaf` and `punafcc` also save the following results in `e()`:

Scalars

| | |
|---|---|
| `e(rank)` | Rank of `e(V)` |
| `e(N)` | number of observations |
| `e(N_sub)` | subpopulation observations |
| `e(N_clust)` | number of clusters |
| `e(N_psu)` | number of samples 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_margins)` | number of terms in *marginlist* |
| `e(k_by)` | number of subpopulations |
| `e(k_at)` | number of `at()` options (always 1 or 2) |

Macros

| | |
|---|---|
| `e(cmd)` | Command name |
| `e(predict)` | program used to implement `predict` |
| `e(atzero)` | `atzero()` option (`regpar` and `punaf` only) |
| `e(atspec)` | `atspec()` option |
| `e(properties)` | `b V` |

Matrices

| | |
|---|---|
| `r(cimat)` | matrix of asymmetric confidence intervals (not saved by `marglmean`) |
| `e(b)` | vector of estimated transformed parameters |
| `e(V)` | dispersion matrix for transformed estimated parameters |
| `e(V_srs)` | simple–random–sampling–without–replacement (co)variance $\hat{V}_{\text{srswor}}$, if `svy` |
| `e(V_srswr)` | simple–random–sampling–with–replacement (co)variance $\hat{V}_{\text{srswr}}$, if `svy` and `fpc()` |
| `e(V_msp)` | misspecification (co)variance $\hat{V}_{msp}$, if `svy` and available |

Functions

| | |
|---|---|
| `e(sample)` | marks estimation sample |

# 3 Methods and formulas

This section is highly technical. The casual reader might like to skip it and proceed to the Examples, and possibly return to this section for reference later.

The methods used are a combination of those in [R] **margins** and in [R] **nlcom**. We denote by $\theta$ the vector of parameters estimated by the most recent model fit, and denote by $f(\mathbf{z}, \theta)$ the function of the covariate row vector $\mathbf{z}$ and the parameter vector $\theta$ whose mean we want to estimate. In general, we aim to estimate a population parameter of the form

$$p(\theta) = \frac{1}{M_R} \sum_{j=1}^{M} R_j f(\mathbf{Z}_j, \theta) \tag{1}$$

where $\mathbf{Z}_j$ is the value of the covariate vector in the $j$th member of the population of $M$ observations, $R_j$ is a binary variable identifying membership of the $j$th observation in a subpopulation (0 for non–members and 1 for members), and $M_R$ is the size of the subpopulation identified by the $R_j$, equal to

$$M_R = \sum_{j=1}^{M} R_j \tag{2}$$

(Note that this population of $M$ observations may or may not be the population from which our data are sampled.)

We aim to estimate $p(\theta)$ using the sample statistic

$$\hat{p} = \frac{1}{w_{\cdot}} \sum_{j=1}^{N} r_j w_j f(\mathbf{z}_j, \hat{\theta}) \tag{3}$$

where $N$ is the number of observations in the sample, $\mathbf{z}_j$ is the vector of covariates in the $j$th observation in the sample, $\hat{\theta}$ is the estimate of the parameter $\theta$ derived from the sample, $r_j$ is a binary variable identifying membership of the $j$th observation in a subsample corresponding to the subpopulation identified by the $R_j$, $w_j$ is the weight for the $j$th observation in the sample, and

$$w_{\cdot} = \sum_{j=1}^{N} r_j w_j \tag{4}$$

is the sum of weights in the subsample. These weights are normally chosen so that (3) is a consistent estimate of the population parameter $p(\theta)$ in (1).

## 3.1 Scenario means estimated

The packages `margprev`, `marglmean`, `regpar`, `punaf` and `punafcc` all start by estimating one or two population scenario means of the form (1) using one or two corresponding sample scenario means of the form (3). Scenarios are here defined as alternative versions of the population and sample datasets, identified by alternative versions of the covariate vectors $\mathbf{Z}_j$ and $\mathbf{z}_j$, respectively. The scenarios are denoted "Scenario 1" (used by all 5 packages) and "Scenario 0" (currently used only by `regpar` and `punaf`). We will denote by $\mathbf{Z}_j^{(0)}$ and $\mathbf{Z}_j^{(1)}$ the values of the covariate vector for the $j$th population observation in Scenarios 0 and 1, respectively, and denote by $\mathbf{z}_j^{(0)}$ and $\mathbf{z}_j^{(1)}$ the values of the covariate vector for the $j$th sample observation in Scenarios 0 and 1, respectively. (We will continue to denote by $\mathbf{Z}_j$ and $\mathbf{z}_j$ the real–world values of the covariate vectors for the $j$th population observation and for the $j$th sample observation, respectively. And we will assume that there exists a mathematical function, deriving $\mathbf{Z}_j^{(i)}$ from $\mathbf{Z}_j$ and deriving $\mathbf{z}_j^{(i)}$ from $\mathbf{z}_j$, for $i \in \{0, 1\}$.)

Each of the packages estimates 1 or 2 scenario means $p^{(i)}(\theta)$ of functions $f^{(i)}(\mathbf{z}, \theta)$, using estimators $\hat{p}^{(i)}$, for scenario indices $i \in \{0, 1\}$, over subpopulations defined by subpopulation indicators $R_j$ as in (1), using subsample indicators $r_j$ as in (3). The subpopulations and subsamples are the same for both scenarios. Therefore, for Scenario $i$, the population scenario mean of (1) becomes

$$p^{(i)}(\theta) = \frac{1}{M_R} \sum_{j=1}^{M} R_j f^{(i)}(\mathbf{Z}_j, \theta) \tag{5}$$

and the corresponding estimator of (3) becomes

$$\hat{p}^{(i)} = \frac{1}{w_.} \sum_{j=1}^{N} r_j w_j f^{(i)}(\mathbf{z}_j, \hat{\theta}) \tag{6}$$

The packages vary in the specification of the functions to be averaged and of the subpopulations over which these functions are to be averaged. The subpopulation is governed by the `subpop()` option, which functions as the option of the same name for `margins` (see [R] **margins**). For a population index $j$ from 1 to $M$, we will denote by $S_j$ the binary variable indicating membership of the $j$th population observation in the subpopulation specified by the `subpop()` option. Similarly, for a sample index $j$ from 1 to $N$, we will denote by $s_j$ the binary variable indicating membership of the $j$th sample observation in the subsample specified by the `subpop()` option.

In the case of the packages `margprev`, `marglmean`, `regpar` and `punaf`, the right hand sides of (5) and (6) are specified by

$$R_j = S_j, \quad r_j = s_j, \quad f^{(i)}(\mathbf{Z}_j, \theta) = \mu(\mathbf{Z}_j^{(i)}, \theta), \quad f^{(i)}(\mathbf{z}_j, \hat{\theta}) = \mu(\mathbf{z}_j^{(i)}, \hat{\theta}) \tag{7}$$

where $\mu(\mathbf{z}, \theta)$ specifies the conditional arithmetic mean calculated by `predict` for the covariate vector $\mathbf{z}$ and the parameter vector $\theta$.

In the case of the package `punafcc`, used for case–control and survival data, the definitions are slightly more complicated, and depend on whether the most recent estimation command is `stcox` or some other estimation command. We will define the truth–value $T(x)$ of a numeric value $x$ to be 1 if $x$ is nonzero, 0 if $x$ is zero, and missing if $x$ is missing. For a population index $j$ from 1 to $M$, we will define $Y_j$ to be the failure indicator variable `_d`, generated by the command `stset`, if the most recent estimation command is `stcox`, and to be the dependent variable given by the estimation result `e(depvar)`, if the most recent estimation command is another estimation command. Similarly, for a sample index $j$ from 1 to $N$, we will define $y_j$ to be the failure indicator variable `_d`, generated by the command `stset`, if the most recent estimation command is `stcox`, and to be the dependent variable given by the estimation result `e(depvar)`, if the most recent estimation command is another estimation command. (See [ST] **stcox** for documentation of `stcox`, and [ST] **stset** for documentation of `stset`.) We will also denote by $\beta$ the column vector containing the sub–vector of the parameter vector $\theta$ containing the coefficients corresponding to the covariates of the $\mathbf{z}$–vector, and denote by $\hat{\beta}$ the column vector containing the corresponding sub–vector of the parameter–estimate vector $\hat{\theta}$. The right hand sides of (5) and (6) are then specified by

$$\begin{aligned} R_j &= S_j T(Y_j) \\ r_j &= s_j T(y_j) \\ f^{(i)}(\mathbf{Z}_j, \theta) &= \exp\left[(\mathbf{Z}_j^{(i)} - \mathbf{Z}_j)\beta)\right] \\ f^{(i)}(\mathbf{z}_j, \hat{\theta}) &= \exp\left[(\mathbf{z}_j^{(i)} - \mathbf{z}_j)\hat{\beta})\right] \end{aligned} \tag{8}$$

This implies that (5) is the population mean risk ratio (or hazard ratio), between Scenario $i$ and the real world, for the "sub–subpopulation" of cases (or failures) of the

subpopulation specified by the `subpop()` option, and that (6) is a corresponding sample mean risk ratio (or hazard ratio) for the "sub–subsample" of cases (or failures) of the subsample specified by the `subpop()` option. A mean between–scenario ratio is a subtly different quantity from a ratio between scenario means, although both of these quantities are known as population unattributable fractions, and can be subtracted from 1 to give population attributable fractions.

Note that, in all the above equations, the packages `margprev`, `marglmean`, `regpar` and `punaf` assume that `predict` specifies a conditional arithmetic mean, and that the package `punafcc` assumes that the parameters of the model are log odds or hazard ratios, while the truth–values of the dependent or failure variable indicate case status or failure. It is the responsibility of the user to ensure that these assumptions are true.

Dispersion–matrix estimates for the estimated scenario means (6) are calculated using methods depending on the `vce()` option, as specified in [R] **margins**.

## 3.2 Symmetric confidence intervals for transformed parameters

Having estimated the scenario means, and their sampling dispersion matrix, using `margins`, we then estimate the transformed parameters, using the Normalizing and variance–stabilizing transformations specified in Table 1. This is done using `nlcom`, so we will use similar notation to [R] **nlcom**. We will denote by $H$ the number of transformed parameters that we want to estimate, and denote the vector of transformed parameters by

$$g(\theta) = [g_1(\theta), \ldots, g_H(\theta)] \tag{9}$$

The $g_h(\theta)$ are functions of the originally–estimated parameter vector $\theta$, and estimated using the corresponding $g_h(\hat{\theta})$. However, we will define them in terms of the scenario means (5) estimated by **margins**. Table 2 gives a list of the transformed parameters estimated by each package, identified by their formulas and their commonly–used parameter names. The logit and log transformations are standard Normalizing and variance–stabilizing transformations for the prevalences of binary variables and for the arithnetic means of non–negative–valued variables and their ratios, respectively. The hyperbolic arctangent `arctanh()`, also known as Fisher's $z$–transform, was recommended by Edwardes (1995) for the general Somers' $D$ parameter, which is discussed extensively in Newson (2006), and which includes as a special case the difference between 2 proportions, exemplified in the scenario–comparison case by the population attributable risk (PAR).

The `nlcom` command inputs the estimates and dispersion matrix for the scenario means $p^{(i)}(\theta)$, generated by `margins`, and outputs the estimates and dispersion matrix for the $g_h(\theta)$, using numerically–estimated derivatives of the transformed parameters with respect to the scenario means. The output estimates vector and dispersion matrix are saved in `r(b)` and `r(V)`, respectively. If the user specifies the `post` option, then these matrices are also saved in `e(b)` and `e(V)`, respectively. In either case, the matrices can be used in the same way to compute symmetric confidence intervals for the transformed parameters.

Table 2: Transformed parameters expressed as functions of scenario means.

| *Package* | *Parameter formulas* | *Parameter names* |
|---|---|---|
| `margprev` | $g_1(\theta) = \text{logit}[\, p^{(1)}(\theta)\,]$ | Logit prevalence |
| `marglmean` | $g_1(\theta) = \log[\, p^{(1)}(\theta)\,]$ | Log arithmetic mean |
| `regpar` | $g_1(\theta) = \text{logit}[\, p^{(0)}(\theta)\,]$ | Logit prevalence |
| | $g_2(\theta) = \text{logit}[\, p^{(1)}(\theta)\,]$ | Logit prevalence |
| | $g_3(\theta) = \text{arctanh}[\, p^{(0)}(\theta) - p^{(1)}(\theta)\,]$ | $z$–transformed PAR |
| `punaf` | $g_1(\theta) = \log[\, p^{(0)}(\theta)\,]$ | Log arithmetic mean |
| | $g_2(\theta) = \log[\, p^{(1)}(\theta)\,]$ | Log arithmetic mean |
| | $g_3(\theta) = \log[\, p^{(1)}(\theta) \,/\, p^{(0)}(\theta)\,]$ | Log PUF |
| `punafcc` | $g_1(\theta) = \log[\, p^{(1)}(\theta)\,]$ | Log PUF |

Table 3: Untransformed parameters expressed as functions of transformed parameters.

| *Package* | *Parameter formulas* | *Parameter names* |
|---|---|---|
| `margprev` | $c_1(\theta) = \text{invlogit}[\, g_1(\theta)\,]$ | Scenario 1 prevalence |
| `regpar` | $c_1(\theta) = \text{invlogit}[\, g_1(\theta)\,]$ | Scenario 0 prevalence |
| | $c_2(\theta) = \text{invlogit}[\, g_2(\theta)\,]$ | Scenario 1 prevalence |
| | $c_3(\theta) = \tanh[\, g_3(\theta)\,]$ | PAR |
| `punaf` | $c_1(\theta) = 1 - \exp[\, g_3(\theta)\,]$ | PAF (cohort or cross–sectional) |
| `punafcc` | $c_1(\theta) = 1 - \exp[\, g_1(\theta)\,]$ | PAF (case–control or survival) |

## 3.3 Asymmetric confidence intervals for untransformed parameters

Usually, the user really wanted to see confidence intervals for arithmetic means and their ratios, or for prevalences and their differences, instead of seeing confidence intervals for the transformed parameters of Table 2. In the case of the logged parameters estimated by `marglmean`, `punaf` and `punafcc`, the `eform` option allows the user to view the untransformed parameters and their confidence limits. However, in the case of `margprev`, the `eform` option displays the odds and not the prevalence, and the `eform` option is not available for `regpar`. Moreover, even in the case of the logged parameters of `punaf` and `punafcc`, the user usually really wanted to estimate the population attributable fraction (PAF), instead of the population unattributable fraction (PUF). To cater for these cases, the packages of the `punaf` suite (except for `marglmean`) also output a matrix of confidence intervals for the untransformed parameters of interest. This confidence interval matrix is stored in `r(cimat)`, and is also automatically listed in the output. For each package, it has one row for each of $K$ parameters $c_k(\theta)$, for $k \in \{1 \ldots K\}$, and 3 columns, containing the estimates, lower confidence limits and upper confidence limits, respectively, of these parameters. The confidence intervals in this matrix are asymmetric.

Table 3 lists the parameters whose asymmetric confidence intervals are listed and saved in the confidence interval matrix by the 4 packages that produce such a matrix.

In each case, the package computes a confidence interval for the transformed parameter $g_h(\theta)$, with estimates and lower and upper confidence limits corresponding to the confidence level specified by the `level()` option, which defaults to `level(95)`. The estimate, lower confidence limit and upper confidence limit for the untransformed parameter $c_k(\theta)$ are then derived by transforming the estimate, lower confidence limit and upper confidence limit, respectively, for the transformed parameter (in the case of `margprev` and `regpar`), or by transforming the estimate, upper confidence limit and lower confidence limit, respectively, for the transformed parameter (in the case of `punaf` and `punafcc`).

# 4 Examples

## 4.1 Scenario comparisons in the `lbw` data using `regpar`

The `lbw` dataset was discussed by Hosmer Jr. et al. (1988) and is distributed by Stata Press. It has one observation for each of a sample of 189 pregnancies, and data on the birth weight of the baby, and on a list of predictive variables. The most interesting of these variables is probably the mother's smoking status during pregnancy, coded as the binary variable `smoke`, which is equal to 1 if the mother smoked during pregnancy and 0 otherwise We will estimate scenario comparisons from a logistic regression model to predict the binary variable `low`, indicating that the baby's birthweight was below 2500 grams.

After loading the `lbw` data, we fit a logistic model of `low` with respect to the exposure factor `smoke` and the confounding factor `race` (1 for "white", 2 for "black", or 3 for "other"):

```
. logit low i.race i.smoke, or vce(robust)
Iteration 0:   log pseudolikelihood =   -117.336
Iteration 1:   log pseudolikelihood = -110.10441
Iteration 2:   log pseudolikelihood = -109.98749
Iteration 3:   log pseudolikelihood = -109.98736
Iteration 4:   log pseudolikelihood = -109.98736
Logistic regression                             Number of obs   =        189
                                                Wald chi2(3)    =      14.30
                                                Prob > chi2     =     0.0025
Log pseudolikelihood = -109.98736               Pseudo R2       =     0.0626
```

| low | Odds Ratio | Robust Std. Err. | z | P>\|z\| | [95% Conf. | Interval] |
|---|---|---|---|---|---|---|
| race | | | | | | |
| 2 | 2.956742 | 1.420439 | 2.26 | 0.024 | 1.153162 | 7.581175 |
| 3 | 3.030001 | 1.187272 | 2.83 | 0.005 | 1.405753 | 6.530954 |
| 1.smoke | 3.052631 | 1.10296 | 3.09 | 0.002 | 1.503568 | 6.197631 |
| _cons | .1587319 | .0515235 | -5.67 | 0.000 | .0840173 | .2998882 |

We see that maternal smoking trebles the odds of low birth weight, and that having a mother of either of the two non–white maternal races has a similar effect on the odds. However, few of the public really understand odds ratios. They might understand more

easily the difference that might result, if all mothers quit smoking before pregnancy, but their racial mix remained the same as in the real world. The `regpar` package can estimate this difference, using the saved estimation results:

```
. regpar, at(smoke=0)
Scenario 0: (asobserved) _all
Scenario 1: smoke=0
Symmetric confidence intervals for the logit proportions
under Scenario 0 and Scenario 1
and for the z-transformed population attributable risk (PAR)
Total number of observations used: 189
```

| | Coef. | Std. Err. | z | P>\|z\| | [95% Conf. | Interval] |
|---|---|---|---|---|---|---|
| Scenario_0 | -.789997 | .1519305 | -5.20 | 0.000 | -1.087775 | -.4922187 |
| Scenario_1 | -1.215955 | .2051031 | -5.93 | 0.000 | -1.61795 | -.8139606 |
| PAR | .0837153 | .0266196 | 3.14 | 0.002 | .0315419 | .1358887 |

```
Asymmetric 95% CIs for the untransformed proportions
under Scenario 0 and Scenario 1
and for the untransformed population attributable risk (PAR)
               Estimate     Minimum     Maximum
  Scenario_0   .31216931   .25203743   .37937104
  Scenario_1   .22864901   .16548776   .30704715
         PAR   .08352031   .03153146   .13505843
```

`regpar` starts its output by specifying Scenarios 0 and 1, in the language of the `at()` option of `margins`. Scenario 0 is `(asobserved) _all`, implying that all covariates and factors are as observed in our real–world sample. Scenario 1 is `smoke=0`, implying that no mothers smoke, but (by default) the factor `race` is distributed as in our real–world sample. `regpar` then displays the logit proportions with low birth rate under Scenarios 0 and 1, and the $z$–transform of the difference between these proportions, known as the population attributable risk (PAR), with their standard errors, $z$–statistics, $P$–values and symmetric confidence limits. Finally, it displays the more comprehensible asymmetric confidence intervals for the untransformed scenario proportions, and for their difference. We see that, in the real world ("`Scenario_0`"), 31.2% of babies are expected to have a low birth weight, but that, in the dream scenario where no mothers smoke and their races stay the same ("`Scenario_1`"), only 22.9% of babies are expected to have a low birth weight. The difference between these scenario percentages ("`PAR`") is 8.4%, with confidence limits from 3.2% to 13.5%. The PAR can be interpreted as the proportion of all babies that have low birth weight because they were born in Scenario 0, instead of in Scenario 1.

Alternatively, we might want to communicate our message to an audience of smoking mothers, who might want to know how much *they* could do for *their* children, if only they quit smoking before pregnancy. To answer this, we might use `regpar` with a `subpop()` option, to compute an exposed–population attributable risk for the sub–population of smoking mothers:

```
. regpar, at(smoke=0) subpop(if smoke==1)
Scenario 0: (asobserved) _all
Scenario 1: smoke=0
Symmetric confidence intervals for the logit proportions
under Scenario 0 and Scenario 1
and for the z-transformed population attributable risk (PAR)
```

```
Total number of observations used: 189
```

| | Coef. | Std. Err. | z | P>\|z\| | [95% Conf. | Interval] |
|---|---|---|---|---|---|---|
| Scenario_0 | -.3829923 | .2373852 | -1.61 | 0.107 | -.8482587 | .0822742 |
| Scenario_1 | -1.436486 | .2279922 | -6.30 | 0.000 | -1.883343 | -.9896299 |
| PAR | .2166422 | .0707321 | 3.06 | 0.002 | .0780098 | .3552746 |

```
Asymmetric 95% CIs for the untransformed proportions
under Scenario 0 and Scenario 1
and for the untransformed population attributable risk (PAR)
               Estimate     Minimum     Maximum
  Scenario_0   .40540541   .29979827   .52055695
  Scenario_1   .19209003   .13200536   .27098519
         PAR   .21331537   .07785194   .34104503
```

This time, the option `subpop(if smoke==1)` restricts the prediction to the subpopulation of smoking mothers, but Scenarios 0 and 1 are defined as before. Once again, `regpar` displays the incomprehensible symmetric confidence intervals for the transformed parameters, followed by the asymmetric confidence intervals for the transformed parameters, which are probably more easily explained to smoking mothers. We see that the children of smoking mothers have a 40.1% prevalence of low birth weight, which might be reduced to 19.2%, if their mothers quit smoking before pregnancy, while their racial mix remained the same. The difference is 21.3%, with confidence limits from 7.8% to 34.1%.

Another possibility is to compare our zero–smoking dream scenario with the "nightmare scenario" where all mothers started smoking, instead of with the intermediate world in which we live. This is done using the `atzero()` option, which can be used to reset Scenario 0, as follows:

```
. regpar, at(smoke=0) atzero(smoke=1)
Scenario 0: smoke=1
Scenario 1: smoke=0
Symmetric confidence intervals for the logit proportions
under Scenario 0 and Scenario 1
and for the z-transformed population attributable risk (PAR)
Total number of observations used: 189
```

| | Coef. | Std. Err. | z | P>\|z\| | [95% Conf. | Interval] |
|---|---|---|---|---|---|---|
| Scenario_0 | -.1697027 | .2464163 | -0.69 | 0.491 | -.6526697 | .3132642 |
| Scenario_1 | -1.215955 | .2051031 | -5.93 | 0.000 | -1.61795 | -.8139606 |
| PAR | .2331622 | .0759652 | 3.07 | 0.002 | .0842732 | .3820512 |

```
Asymmetric 95% CIs for the untransformed proportions
under Scenario 0 and Scenario 1
and for the untransformed population attributable risk (PAR)
               Estimate     Minimum     Maximum
  Scenario_0   .45767584   .34238817   .57768182
  Scenario_1   .22864901   .16548776   .30704715
         PAR   .22902683   .08407429   .36448745
```

We see that Scenario 0 is set by the `atzero()` option to `smoke=1`, while Scenario 1 is still `smoke=0`. Once again, `regpar` displays the symmetric confidence intervals for the transformed parameters, followed by the asymmetric confidence intervals for the

untransformed parameters. We see that, if all mothers smoked and the racial mix stayed the same, then 45.8% of children might have low birth weight. The dream–scenario prevalence, where no mothers smoke and the racial mix stays the same, is still 22.9%, as before. The difference in prevalence between the nightmare Scenario 0 and the dream Scenario 1 is 22.9%, with confidence limits from 8.4% to 36.4%.

`regpar` might be even more useful if we had a large number of confounders, instead of the single confounder `race`. In that case, we might want to reduce the potentially infinite–dimensioned confounder space to a finite–dimensioned confounder space, by defining a propensity score for smoking, as recommended by Rosenbaum and Rubin (1983). Such a propensity score might be defined using a logistic regression model to regress `smoke` with respect to the multiple confounders, followed by using `predict` to define the smoking propensity score for each subject as the predicted probability of smoking for that subject. We might then define a grouping variable for the propensity score using `xtile` (see [D] **pctile**), and then use the propensity–group variable in a second logistic regression model, with `low` as the outcome, and with smoking exposure and smoking–propensity group as the predictors. A problem with using propensity scores or groups as covariates in a logistic regression model is that the conditional odds ratio with respect to exposure, adjusted for the propensity score, is not the same quantity as the conditional odds ratio with respect to exposure, adjusted for the original confounders. This is in contrast to conditional mean differences (including prevalence differences) between exposed and unexposed subjects, where the mean difference, conditional on the propensity score, is equal to the mean difference, conditional on the original covariates. Austin et al. (2007) argue that, if we use the propensity–adjusted odds ratio to estimate the confounder–adjusted odds ratio, then our estimate is likely to be biassed towards the "null hypothesis" that the odds ratio is 1, leading to "underestimation of the magnitude of the exposure effect". This problem can arguably be solved by fitting a logistic regression of disease with respect to exposure–propensity and exposure, and then using `regpar` to define the "exposure effect" as a difference in marginal disease prevalences between a "nightmare scenario", where exposure–propensity stays the same and all subjects are exposed, and a "dream scenario", where exposure–propensity stays the same and all subjects are unexposed.

## 4.2 Scenario comparisons in the `lbw` data using `punaf`

Alternatively, again, we might want to estimate the possibility for disease prevention as a proportion of the total "disease burden" of low birth weight, instead of as a proportion of all babies. This can be done using `punaf` after the same logistic regression model as before. `punaf` compares scenario arithmetic means (including scenario prevalences) using ratios, instead of differences. These ratios, known as population unattributable fractions (PUFs), can then be subtracted from 1 to obtain population attributable fractions (PAFs). As a simple example, we compare the smoking–free dream scenario to the real world once again:

```
. punaf, at(smoke=0) eform
Scenario 0: (asobserved) _all
Scenario 1: smoke=0
Confidence intervals for the means under Scenario 0 and Scenario 1
```

```
and for the population unattributable faction (PUF)
Total number of observations used: 189
```

| | Mean/Ratio | Std. Err. | z | P>\|z\| | [95% Conf. | Interval] |
|---|---|---|---|---|---|---|
| Scenario_0 | .3121693 | .0326225 | -11.14 | 0.000 | .2543534 | .3831271 |
| Scenario_1 | .228649 | .0361738 | -9.33 | 0.000 | .1676887 | .3117704 |
| PUF | .7324519 | .0818807 | -2.79 | 0.005 | .5883333 | .911874 |

```
95% CI for the population attributable fraction (PAF)
               Estimate     Minimum     Maximum
        PAF    .2675481   .08812601   .41166675
```

We see that the scenarios are as in our first example with `regpar`, and that the scenario means computed using `punaf` are the same as the untransformed scenario prevalences using `regpar`. The confidence limits are slightly different, because they are computed using the log transform instead of the logit transform. The PUF is the ratio between the Scenario 1 mean and the Scenario 0 mean, and represents the fraction of the Scenario 0 disease burden that would remain, if the babies were born in Scenario 1. (Note that the `eform` option ensures that we see confidence intervals for the scenario means and their ratio, instead of for their logs.) Finally, `punaf` subtracts the PUF (and its upper and lower confidence limits) from 1 to obtain the PAF (and its lower and upper confidence limits), and displays these in the bottom line of output. We see that 26.8% of the "disease burden" of low birth weight might be eliminated by eliminating maternal smoking, assuming that the racial mix stays the same, with confidence limits from 8.8% to 41.2%.

## 4.3 `margprev` and `marglmean` in the `lbw` data

We can also estimate marginal prevalences and means without comparing them between different scenarios. The `marglprev` package can estimate marginal odds, and the corresponding marginal prevalences, from the current estimation results. For instance, the marginal odds and prevalence of low birthweight, in a world of smoking mothers with the existing race distribution, could be estimated as follows:

```
. margprev, at(smoke==1) eform
Scenario 1: smoke==1
Confidence interval for the marginal odds
under Scenario 1
Total number of observations used: 189
```

| | Odds | Std. Err. | z | P>\|z\| | [95% Conf. | Interval] |
|---|---|---|---|---|---|---|
| Scenario_1 | .8439156 | .2079545 | -0.69 | 0.491 | .5206539 | 1.367883 |

```
Asymmetric 95% CI for the untransformed marginal prevalence
under Scenario 1
               Estimate     Minimum     Maximum
  Scenario_1   .45767584   .34238817   .57768182
```

This time, only Scenario 1 is specified, as there is no Scenario 0. `margprev` displays first the marginal odds (not the marginal log odds, because `eform` has been specified), and then a confidence interval for the marginal prevalence, which is the same as the one calculated for the same "nightmare scenario" by `regpar`.

The `marglmean` package can estimate general marginal means for general non–negative variables, using the log transform to calculate confidence intervals. For instance, we might fit a gamma–family regression model for the non–negative variable `bwt`, representing birth weight in grams, with respect to race and smoking status, as follows, using the `glm` command detailed in Hardin and Hilbe (2007):

```
. glm bwt i.race i.smoke, family(gamma) link(log) eform vce(robust)
Iteration 0:   log pseudolikelihood = -1698.0172
Iteration 1:   log pseudolikelihood = -1697.9741
Iteration 2:   log pseudolikelihood = -1697.9741
Generalized linear models                         No. of obs      =       189
Optimization     : ML                             Residual df     =       185
                                                  Scale parameter =  .0555296
Deviance         =   12.0823464                   (1/df) Deviance =    .06531
Pearson          =  10.27297009                   (1/df) Pearson  =  .0555296
Variance function: V(u) = u^2                     [Gamma]
Link function    : g(u) = ln(u)                   [Log]
                                                  AIC             =  18.01031
Log pseudolikelihood = -1697.974084               BIC             = -957.6409
```

| bwt | exp(b) | Robust Std. Err. | z | P>\|z\| | [95% Conf. | Interval] |
|---|---|---|---|---|---|---|
| race | | | | | | |
| 2 | .8594198 | .042562 | -3.06 | 0.002 | .7799205 | .9470227 |
| 3 | .863627 | .0360104 | -3.52 | 0.000 | .795855 | .9371702 |
| 1.smoke | .8697043 | .032986 | -3.68 | 0.000 | .8073975 | .9368193 |
| _cons | 3332.454 | 97.62645 | 276.88 | 0.000 | 3146.499 | 3529.398 |

The parameters are a baseline arithmetic mean `_cons` (in grams) for the babies of non–smoking white mothers, 2 arithmetic mean ratios for the babies of black and miscellaneous–race mothers, and an arithmetic mean ratio for the babies of smoking mothers, compared to the babies of non–smoking mothers of the same race. We can now use `marglmean` to estimate the marginal arithmetic mean, with asymmetric confidence limits, that would be expected if all mothers smoked and the race distribution remained the same:

```
. marglmean, at(smoke=1) eform
Scenario 1: smoke=1
Asymmetric confidence interval for the marginal mean
under Scenario 1
Total number of observations used: 189
```

| | Mean | Std. Err. | z | P>\|z\| | [95% Conf. | Interval] |
|---|---|---|---|---|---|---|
| Scenario_1 | 2702.087 | 80.18231 | 266.28 | 0.000 | 2549.416 | 2863.902 |

We see that the mean birthweight, in this scenario, would be 2702 grams, with confidence limits from 2549 grams to 2864 grams. We could also use `punaf` to estimate the ratio

(or PUF) between this scenario mean and the scenario mean where no mothers smoked (not shown to save space).

## 4.4 `punafcc` in case–control and survival data

The `punafcc` package calculates unattributable and attributable fractions for case–control and survival data. The unattributable fraction, in this case, is a mean between–scenario odds ratio for cases (if used after a logistic estimation), or a mean between–scenario hazard ratio for lifetimes that terminated from the cause of interest (if used after a Cox survival regression), instead of a ratio of scenario means. Currently, the only scenarios that can be compared in this way are "Scenario 1" and the world in which we sampled the data.

The `downs` dataset is an example of a case–control study dataset, described and used in [ST] **epitab** to demonstrate the `cci` command. The data are from Rothman et al. (2008), and represent a case–control study, whose outcome variable is Down's syndrome in infants, with maternal spermicide use as the exposure, and maternal age group as a confounding factor. The dataset has 8 observations and 4 variables. These variables are 3 binary key variables `case`, `exposed` and `age`, identifying the 8 observations uniquely and indicating case status, exposure status, and maternal age at or above 35 years, respectively, and 1 integer variable `pop`, containing frequency weights for the combination of case status, exposure status and age group indicated by the 3 key variables.

We start by loading the `downs` dataset, and fitting a full logistic regression model, allowing age odds ratios and different exposure odds ratios for the 2 age groups:

```
. webuse downs, clear
. logit case i.age i.exposed i.age#i.exposed [fweight=pop], or vce(robust)
Iteration 0:   log pseudolikelihood = -85.885722
Iteration 1:   log pseudolikelihood = -82.752975
Iteration 2:   log pseudolikelihood = -81.552365
Iteration 3:   log pseudolikelihood = -81.451562
Iteration 4:   log pseudolikelihood = -81.451332
Iteration 5:   log pseudolikelihood = -81.451332
Logistic regression                               Number of obs   =       1270
                                                  Wald chi2(3)    =      11.64
                                                  Prob > chi2     =     0.0087
Log pseudolikelihood = -81.451332                 Pseudo R2       =     0.0516
```

| case | Odds Ratio | Robust Std. Err. | z | P>\|z\| | [95% Conf. | Interval] |
|---|---|---|---|---|---|---|
| 1.age | 4.104651 | 2.775961 | 2.09 | 0.037 | 1.090465 | 15.45044 |
| 1.exposed | 3.394231 | 2.290446 | 1.81 | 0.070 | .9043692 | 12.73905 |
| age#exposed | | | | | | |
| 1 1 | 1.689141 | 2.389726 | 0.37 | 0.711 | .105541 | 27.034 |
| _cons | .0084986 | .002846 | -14.24 | 0.000 | .0044086 | .0163831 |

These odds ratios are not easy to interpret at first sight, especially the interaction odds

ratio, which is a ratio of ratios. We might find it easier to understand the fractions of Down's syndrome births unattributable and attributable to spermicide exposure. These can be estimated using `punafcc`. It is *probably* a good idea to use the option `vce(unconditional)`, because the "covariates"exposure status and maternal age will definitely be subject to sampling error, if we sample cases and controls and then measure exposure status and maternal age.

```
. punafcc, at(exposed=0) eform vce(unconditional)
Scenario 0: (asobserved) _all
Scenario 1: exposed=0
Confidence interval for the population unattributable faction (PUF)
Total number of observations used: 1270
```

| | Ratio | Std. Err. | z | P>\|z\| | [95% Conf. | Interval] |
|---|---|---|---|---|---|---|
| PUF | .816142 | .1181495 | -1.40 | 0.160 | .6145268 | 1.083903 |

```
95% CI for the population attributable fraction (PAF)
               Estimate     Minimum     Maximum
         PAF   .18385804  -.08390349   .38547325
```

We see from the population unattributable fraction (PUF) that, in a fantasy scenario where no mothers were exposed to spermicide, we might expect the rate of Down's syndrome to be 81.6% of that observed in the population from which our cases and controls were sampled, with 95% confidence limits from 61.5% to 108.4%. This allows the possibility that spermicide use might even be slightly protective, at least at some maternal ages. The population attributable fraction (PAF) is computed by subtracting the PUF from 1, and therefore has confidence limits from -8.4% to 38.5%. These limits are wide enough to include zero, and even a small range of negative values.

Similarly, we can estimate unattributable and attributable fractions in the Stanford heart transplant dataset `heart3`, which 1 observation per study subject per time interval, where the time interval can be a pre–transplant interval (present for all subjects) or a post–transplant interval (present only for subjects who received a transplant). We will fit the Cox regression model used in [ST] **stcox**, where death is regressed with respect to the quantitative covariates `year` (year of acceptance) and `age` (age in years at start), and the binary variables `posttran` (indicating that the interval is post–transplant) and `surgery` (indicating prior heart surgery on entry). We do not need to use `stset`, as this has already been done to the dataset.

```
. use http://www.stata-press.com/data/r12/stan3, clear
(Heart transplant data)
. stcox age posttran surg year, vce(robust)
         failure _d:  died
   analysis time _t:  t1
                 id:  id
Iteration 0:   log pseudolikelihood = -298.31514
Iteration 1:   log pseudolikelihood =  -289.7344
Iteration 2:   log pseudolikelihood = -289.53498
Iteration 3:   log pseudolikelihood = -289.53378
Iteration 4:   log pseudolikelihood = -289.53378
Refining estimates:
Iteration 0:   log pseudolikelihood = -289.53378
Cox regression -- Breslow method for ties
```

```
No. of subjects      =          103                 Number of obs    =       172
No. of failures      =           75
Time at risk         =      31938.1
                                                    Wald chi2(4)     =     19.68
Log pseudolikelihood =   -289.53378                 Prob > chi2      =    0.0006
                                        (Std. Err. adjusted for 103 clusters in id)
```

| _t | Haz. Ratio | Robust Std. Err. | z | P>\|z\| | [95% Conf. | Interval] |
|---|---|---|---|---|---|---|
| age | 1.030224 | .0148771 | 2.06 | 0.039 | 1.001474 | 1.059799 |
| posttran | .9787243 | .2961736 | -0.07 | 0.943 | .5408498 | 1.771104 |
| surgery | .3738278 | .1304912 | -2.82 | 0.005 | .1886013 | .7409665 |
| year | .8873107 | .0613176 | -1.73 | 0.084 | .7749139 | 1.01601 |

We see the hazard ratios associated with each binary or quantitative covariate, with Huber (or "Robust") confidence limits.

We might want to know the fractions of mortality attributable and unattributable to subjects *not* having prior surgery. That is to say, we might want to ask how much the death rate in the study might have decreased, if all patients had received heart surgery prior to joining the study, and accceptance years, ages and transplant history during the study had been the same as in the real world, and how much hazard would have remained. This can be done using `punafcc`, with the option `vce(unconditional)` as before, because the covariate values of lifetimes that ended in death will be subject to sampling error, assuming that deaths do not occur by design.

```
. punafcc, at(surgery==1) eform vce(unconditional)
Scenario 0: (asobserved) _all
Scenario 1: surgery==1
Confidence interval for the population unattributable faction (PUF)
Total number of observations used: 172
```

| | Ratio | Std. Err. | z | P>\|z\| | [95% Conf. | Interval] |
|---|---|---|---|---|---|---|
| PUF | .4239216 | .1317422 | -2.76 | 0.006 | .2305459 | .7794955 |

```
95% CI for the population attributable fraction (PAF)
               Estimate     Minimum     Maximum
         PAF    .5760784   .22050449   .76945406
```

We see, from the PUF, that giving all the subjects prior surgery, and changing nothing else, might have reduced mortality to 42.4% of the level observed. When this PUF is subtracted from 100% to get a PAF, we conclude that 57.6% of the mortality observed is attributable to subjects *not* having prior surgery, with confidence limits from 22.1% to 76.9%.

Note that the option `vce(unconditional)`, recommended here for use with `punafcc`, requires that the user must specify `vce(robust)` in the estimation command generating the parameter estimates. Note, also, that the interpretation of the unattributable and attributable fractions requires the assumption that the association between the outcome and the exposure altered in the fantasy scenarios is indeed causal, meaning that the outcome will change as predicted, if we intervene to change the exposure.

## 4.5 Standardization as out–of–sample prediction

We can also compare outcomes between different models applied to the same scenario, instead of between the same model applied to different scenarios. For instance, in a multi–center study, we might fit a logistic regression model of disease with respect to gender and age to the data from a center, and then input a dataset specifying a standard distribution of gender and age, and use `margprev` to estimate the marginal prevalence expected, if the logistic model is applied to that standard population. This is an example of out–of–sample prediction, and the 5 packages introduced here have a `noesample` option to make this possible, similar to the option of the same name for `margins`.

The GA$^2$LEN (Global Allergy and Asthma European Network) Survey is part of a multi–regional European study on asthma and allergy in Europe. Sensitivity to a range of allergens was measured on a sub–sample of subjects in each region, using skin prick tests. We wanted to compare sensitivity prevalences, standardized to a common age distribution, between 13 European regions. To do this, we fitted a logistic regression model for sensitivity to each allergen in each region, with respect to gender and age, and then used `margprev` to estimate a standardized sensitivity prevalence.

For instance, in the case of sensitivity to cat allergen in the United Kingdom, the logistic model (fitted using sampling probability weights) was as follows:

```
. logit spt_cat male fquesagec [pwei=sampwt5], or
Iteration 0:   log pseudolikelihood = -1030.8768
Iteration 1:   log pseudolikelihood = -977.80033
Iteration 2:   log pseudolikelihood = -973.41056
Iteration 3:   log pseudolikelihood = -973.39866
Iteration 4:   log pseudolikelihood = -973.39866
Logistic regression                               Number of obs   =        159
                                                  Wald chi2(2)    =       4.04
                                                  Prob > chi2     =     0.1328
Log pseudolikelihood = -973.39866                 Pseudo R2       =     0.0558
```

| spt_cat | Odds Ratio | Robust Std. Err. | z | P>\|z\| | [95% Conf. | Interval] |
|---|---|---|---|---|---|---|
| male | 2.527963 | 1.535882 | 1.53 | 0.127 | .7684525 | 8.316188 |
| fquesagec | .6700974 | .2209261 | -1.21 | 0.225 | .3511585 | 1.278712 |
| _cons | .0794547 | .0300632 | -6.69 | 0.000 | .0378487 | .1667967 |

The variables `spt_cat` and `male` are binary indicators of skin–prick sensitivity to cat allergen and male gender, and the variable `fquesagec` is a continuous age, centered by subtracting 48 years and divided by 10 years to be expressed in decades over 48 years. Therefore, the parameter `_cons` is a baseline sensitivity odds for 48–year–old women, the parameter `male` is a male–gender odds ratio, and the parameter `fquesagec` is a per–decade odds ratio for age, assuming the effect of age on odds to be exponential. To derive a standardized prevalence from these parameters, we first loaded (and listed) a new dataset, with 1 observation per gender per age group, and data on the numbers of individuals in that gender and age group in a European standard population:

```
. use estanpop, clear
```

```
. list male agemin agemax agemean fquesagec stanpop, abbr(32) sepby(male)
```

| | male | agemin | agemax | agemean | fquesagec | stanpop |
|---|---|---|---|---|---|---|
| 1. | 0 | 20 | 24 | 22 | -2.6 | 7000 |
| 2. | 0 | 25 | 29 | 27 | -2.1 | 7000 |
| 3. | 0 | 30 | 34 | 32 | -1.6 | 7000 |
| 4. | 0 | 35 | 39 | 37 | -1.1 | 7000 |
| 5. | 0 | 40 | 44 | 42 | -.6 | 7000 |
| 6. | 0 | 45 | 49 | 47 | -.1 | 7000 |
| 7. | 0 | 50 | 54 | 52 | .4 | 7000 |
| 8. | 0 | 55 | 59 | 57 | .9 | 6000 |
| 9. | 0 | 60 | 64 | 62 | 1.4 | 5000 |
| 10. | 0 | 65 | 69 | 67 | 1.9 | 4000 |
| 11. | 0 | 70 | 74 | 72 | 2.4 | 3000 |
| 12. | 1 | 20 | 24 | 22 | -2.6 | 7000 |
| 13. | 1 | 25 | 29 | 27 | -2.1 | 7000 |
| 14. | 1 | 30 | 34 | 32 | -1.6 | 7000 |
| 15. | 1 | 35 | 39 | 37 | -1.1 | 7000 |
| 16. | 1 | 40 | 44 | 42 | -.6 | 7000 |
| 17. | 1 | 45 | 49 | 47 | -.1 | 7000 |
| 18. | 1 | 50 | 54 | 52 | .4 | 7000 |
| 19. | 1 | 55 | 59 | 57 | .9 | 6000 |
| 20. | 1 | 60 | 64 | 62 | 1.4 | 5000 |
| 21. | 1 | 65 | 69 | 67 | 1.9 | 4000 |
| 22. | 1 | 70 | 74 | 72 | 2.4 | 3000 |

In this dataset, `male` indicates male gender, `agemin`, `agemax` and `agemean` contain minimum, maximum and mean ages in years, `fquesagec` contains the mean age in decades centered at 48 years, and `stanpop` contains the number of individuals with that gender and age group in the European standard population. We can now estimate the marginal odds and prevalence by applying our model to this dataset, using `stanpop` as a frequency–weight variable:

```
. margprev [fwei=stanpop], eform noesample
Scenario 1: (asobserved) _all
Confidence interval for the marginal odds
under Scenario 1
Total number of observations used: 134000
```

| | Odds | Std. Err. | z | P>\|z\| | [95% Conf. | Interval] |
|---|---|---|---|---|---|---|
| Scenario_1 | .1782219 | .07486 | -4.11 | 0.000 | .0782391 | .4059742 |

```
Asymmetric 95% CI for the untransformed marginal prevalence
under Scenario 1
               Estimate     Minimum     Maximum
  Scenario_1   .15126346   .07256191    .2887494
```

We see the marginal odds, and the more comprehensible marginal prevalence of 15.1% (95% CI, 7.3% to 28.9%). The marginal odds for this region (the UK) and the 12 others were entered into the SSC package `parmhet` to compute heterogeneity statistics. The $I^2$ statistic of Higgins and Thompson (2002) was 46.4%, with a $P$–value of .033, so there seems to be heterogeneity in cat allergy prevalence between European regions, not attributable to heterogeneity in gender and age distribution.

## 5 Acknowledgements

I would like to thank all at StataCorp for making the `margins` command available. In particular, I would like to thank Kristin MacDonald, Jennifer Rolfes, Miguel Dorta, and Jeff S. Pitblado for handling the demanding and highly technical `margins` queries with which I routinely bombarded them during the development of the suite of packages described here. I would also like to thank Dr Mohammadreza Bozorgmanesh of the Research Institute for Endocrine Sciences, Tehran, Iran for drawing my attention to the literature on the PAF, especially on extensions of the PAF to survival analyses. I would also like to thank the administrators of the GA$^2$LEN survey for allowing me to include the example of Subsection 4.5 in this article. The Global Allergy and Asthma Network of Excellence (GA$^2$LEN) conducted an epidemiological project from 2007 to 2009 to examine respiratory and allergic disease in Europe. This work involved 24 research centers in 15 countries across Europe and was funded by the European Union. My own work at Imperial College London is financed by the UK Department of Health.

## 6 References

Austin, P. C., P. Grootendorst, S. T. Normand, and G. M. Anderson. 2007. Conditioning on the propensity score can result in biased estimation of common measures of treatment effect: A Monte Carlo study. *Statistics in Medicine* 26: 754–768.

Brady, A. 1998. sbe21: Adjusted population attributable fractions from logistic regression. *Stata Technical Bulletin* 42: 8–12. Reprinted in *Stata Technical Bulletin Reprints*, vol. 7, pp. 137–143. College Station, TX: Stata Press.

Edwardes, M. D. 1995. A confidence interval for $\Pr(X < Y) - \Pr(X > Y)$ estimated from simple cluster samples. *Biometrics* 51: 571–578.

Gordis, L. 2000. *Epidemiology*. 2nd ed. Philadelphia: Saunders.

Greenland, S., and K. Drescher. 1993. Maximum likelihood estimation of the attributable fraction from logistic models. *Biometrics* 49: 865–872.

Hardin, J. W., and J. M. Hilbe. 2007. *Generalized Linear Models and Extensions*. 2nd ed. College Station, TX: Stata Press.

Higgins, J. P. T., and S. G. Thompson. 2002. Quantifying heterogeneity in a meta–analysis. *Statistics in Medicine* 21: 1539–1558.

Hosmer Jr., D. W., S. Lemeshow, and J. Klar. 1988. Goodness-of-fit testing for the logistic regression model when the estimated probabilities are small. *Biometrical Journal* 30: 911–924.

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

Newson, R. 2006. Confidence intervals for rank statistics: Somers' *D* and extensions. *Stata Journal* 6(3): 309–334.

Rosenbaum, P. R., and D. B. Rubin. 1983. The central role of the propensity score in observational studies for causal effects. *Biometrika* 70: 41–55.

Rothman, K. J., S. Greenland, and T. L. Lash. 2008. *Modern Epidemiology*. 3rd ed. Philadelphia: Lippincott Williams & Wilkins.

**About the author**

Roger B. Newson is a Lecturer in Medical Statistics at Imperial College London, UK, working principally in asthma research. He wrote the packages `margprev`, `marglmean`, `regpar`, `punaf`, `punafcc` and `parmhet`.