Regression, poststratification, and small-area estimation with sampling weights[*]

Andrew Gelman,[†] Yajuan Si,[‡] and Brady T. West[‡]

12 Mar 2024

Abstract

A well-known rule in practical survey research is to include weights when estimating a population average but not to use weights when fitting a regression model—as long as the regression predictors x include all the information that went into the sampling weights w. But what if you don't know where the weights came from? We propose a quasi-Bayesian approach using a joint regression of the outcome and the sampling weight, followed by poststratifcation on x and w, thus using design information within a model-based context to obtain inferences for small-area estimates, regressions, and other population quantities of interest.

1. Background

1.1. Survey weights

One of the central challenges of statistics is generalizing from sample to population. The natural first step here is to adjust for known, expected, or assumed discrepancies between sample and population[1]—but even this basic level of correction can be challenging, especially when sample and population diverge in many dimensions (for example, age, sex, education, ethnicity, geography, and political affiliation in social surveys).

Weighting is a way to summarize an adjustment: each item in the sample gets a nonnegative weight which is intended to be proportional to its representation in the population. Population estimates can then be obtained as weighted averages of the sample.

Five difficulties arise with classical survey weighting: construction of weights, uncertainty estimates, small-area estimation, regression modeling, and the general inflation in the variance of weighted estimates due to the use of weights in estimation (the so-called unequal-weighting effect).

Construction of weights is difficult because real-world surveys will require adjustment for many factors, and simple approaches based on poststratification or estimated probabilities of sampling often result in highly noisy weights. Noisier weights lead to losses in the efficiency of weighted estimates: the more variability that exists in the weights, the less efficient the weighted survey estimates become (Korn and Graubard, 1999). This in turn motivates more complicated approaches based on smoothing or modeling the weights, which can be done but at the cost of many choices in modeling and estimation (Little, 1991; Gelman and Little, 1998; Little and Vartivarian, 2003; Chen et al., 2006; Gelman, 2007; Chen et al., 2012, 2017; Xie et al., 2020; Si et al., 2020; Ben-Michael et al., 2024).

Standard errors or other uncertainty measures with weighted averages are challenging because a set of weights is sufficient to define a weighted average but does not specify a full probability

[*] Data and code are at `http://www.stat.columbia.edu/~gelman/weight_regression/`. We thank Rod Little, Michael Elliott, Jae-Kwang Kim, and Terrance Savitsky for helpful comments and the U.S. National Science Foundation, National Institutes of Health, and Office of Naval Research for partial support of this work.

[†] Department of Statistics and Department of Political Science, Columbia University, New York.

[‡] Institute for Social Research, University of Michigan, Ann Arbor.

[1] An example of a *known* discrepancy between sample and population would be a sample of 60 women and 40 men that is intended to represent a population that is 52% women and 48% men. An example of an *expected* discrepancy would be clusters sampled with probability proportional to a known measure of size. These discrepancies become *assumed* if the population proportions and sampling probabilities are approximate and not known.

model; additional assumptions must be added beyond those implied by the weights (Lumley, 2004; Solon et al., 2015).

Small-area estimation using weights is difficult because a small area may have so few observations that no purely local estimate, weighted or otherwise, would be reasonable (Fay and Herriott, 1979; Rao, 2003). Consider, for example, a national political survey that contains five responses from Wyoming, all of whom support the Republican candidate for president. Any weighted average would result in an obviously wrong estimate of 100% Republican support in the state. Weighting is defeated by data granularity, and modeling is required.

Regression modeling with weights can work in simple settings, replacing least squares or maximum likelihood with weighted versions of these methods. Several procedures exist that allow analysts to test whether survey weights are needed for the estimation of a given regression model (Bollen et al., 2016), which can lead to more efficient estimates of regression model parameters if the weights are not in fact necessary. Procedures have also been developed to minimize the impact of noisy weights on estimated regression model parameters (Pfeffermann, 2011). But the use of weights to fit models becomes a more difficult issue when moving to more advanced multilevel, Bayesian, or regularized methods that are needed to answer complex questions in the presence of data granularity (DuMouchel and Duncan, 1983; Pfeffermann et al., 1998; Rabe-Hesketh and Skrondal, 2006; Lumley and Scott, 2017).

This is not to say that weighting-based methods are useless. Much work has gone into population inference using survey weights. Our point here is that there are no generally applicable or easy solutions to the problem of adjusting for discrepancies between sample and population, and so there are theoretical, methodological, and applied reasons for wanting a generally-applicable and unified approach to regression modeling and small-area estimation using survey weights. The approach presented here follows ideas of Särndal (1978), Kalton (1983), Pfefferman (1993), Little (2015), and others that incorporate design information into model-based inference.

1.2. Multilevel regression and poststratification

Multilevel regression and poststratification (MRP) or, more generally, regularized regression and poststratification, is an approach to survey analysis that combines modeling of the data with adjustment for nonrepresentativeness of the sample. In the basic MRP setup, an outcome y and background variables x are observed in the sample, and the distribution of x is known in the population. If the variables in x are discrete, then their interactions define poststratification cells. If the observed data are independently sampled with probabilities of selection that do not vary within poststratification cells, population inference can be performed by fitting a regression model of y on x and then averaging over the cells in proportion to their known population counts (Holt and Smith, 1979; Little, 1993).

So far, this is simply regression and poststratification. The multilevel part comes in because, given the implicit assumption of constant probability of inclusion within cells, there is a desire to poststratify on as many factors as possible, and a regression model with a large number of predictors and interactions cannot be estimated stably using least squares. Multilevel modeling is a good way to fit a regression with many predictors such as arise when modeling survey responses given demographic and geographic factors (Gelman and Little, 1997). Other approaches are possible, hence we have also used the more general term, "regularized regression and poststratification" or RRP (Gelman, 2018; Bisbee, 2019; Broniecki et al., 2021; Gopelrud, 2024). A key attribute of MRP (or RRP) is that it allows predictions for y given values of x that are not observed in the sample, or which have such small counts in the sample that it would be impossible to make predictions for them from local data alone.

There is a growing literature on MRP and its generalizations. Challenges include obtaining good group-level predictors for multilevel regressions (so that, for example, inferences for small states in a national survey are partially pooled toward reasonable state-level estimates rather than to a national baseline); adjusting for non-census variables, in which case the population counts of the poststratification cells themselves must be estimated from the data (Su and Gelman, 2023; Li and Si, 2024); analyzing cluster samples when the cluster sizes in the population are unknown (Graubard and Korn, 2002; Stanek and Singer, 2004; Makela et al., 2018); and, with particular relevance to the present research, modeling unequal sampling probabilities within poststratification cells. One quick way to incorporate survey weights is to replace the observed mean response within each cell by its weighted mean and use an adjusted within-cell variance estimate (Potthoff et al., 1992; Ghitza and Gelman, 2013; Chen et al., 2014), but this approach fails when data are sparse and many cells have only a single respondent, in which case important variation in the weights can be missed.

1.3. Analyzing surveys collected by others

Textbooks on survey sampling focus on the scenario in which the data are analyzed by the same team that conducted the survey. There is some literature on the construction of sampling weights, but not much on the analysis of surveys collected by others, even though this type of *secondary analysis* is a common mode of social science research (Kish, 1992; Korn and Graubard, 1999; West et al., 2016; Heeringa et al., 2017; Haziza and Beaumont, 2017; Lohr, 2022). Publicly-available surveys typically come with weights but often do not fully explain how the weighting scheme was chosen or exactly how the weights are computed, hence it can be difficult or impossible to reproduce the procedure starting from the data (Voss et al., 1995). Unfortunately, the documentation provided by these surveys for data users, talking about the weights and other design features and how they should be used, varies tremendously in detail and usefulness (Kolenikov et al., 2020).

When conducting analyses of data collected by others, researchers are often advised to use the weights when estimating population averages but not when fitting regression models, as long as all the variables that went into the weighting are included as predictors; see, for example, Winship and Radbill (1994). This advice is useful where it can be followed, but it does not resolve the question of what to do when fitting a regression whose predictors do not include all the variables that went into the weights. In addition, it is awkward to consider averaging and regression as different problems, given that averaging is a special case of regression. For example, when estimating the average within a subgroup (for example, average responses for women or men), we might simply use the weighted average from the relevant group in the sample, but if the subgroup is small enough (for example, individual states or geographic/demographic categories in a national survey), we would want to perform small-area estimation using regression.

The literature on design-based secondary analysis of survey data is clear on the point that using correctly-specified inverse-probability weights will produce consistent and asymptotically unbiased estimates of regression parameters with respect to the sample design, even if that regression model model has been poorly specified (Korn and Graubard, 1999; Heeringa et al., 2017). But such weighted estimates can be noisy; in addition, if the weights provide little or no predictive power beyond what is in the regression predictors, than weighting can simply add unnecessary noise. We would like to get the best of both worlds, using weighting adjustments just to the extent that the weights add relevant information.

The goal in the present paper is to share a general approach to analyzing surveys with weights, under the scenario that the weights have already been constructed before the analyst sees the data, as in West and McCabe (2012). This is similar to the idea in multiple imputation for public surveys, in which the organization in charge of the survey uses sophisticated methods to construct imputations, and then users can analyze the imputed datasets, taking the imputations as given (Rubin, 1996; Meng, 1994). Dividing the problem in two parts—first the construction of the weights, then the analysis of the weighted dataset—entails an inevitable loss of statistical efficiency (except in some special cases), but, as with imputation, offers practical gains of division of labor and facilitates comparability of analyses by different users of the same survey.

2. A quasi-Bayesian approach to regression with survey weights

2.1. Model

Suppose we have a vector of background variables x that are observed in the sample and whose distribution is known in the population, and a weight variable $w > 0$ and scalar outcome y that are known only in the sample. Assume the data have been sampled independently from the population with probabilities inversely proportional to the weights.[2] The poststratification cells $j = 1, \dots, J$ correspond to the possible values of x in the population; we label these as x_j, with N_j being the size of cell j in the population. We assume that the N_j's are known with certainty. In practice, these quantities may be estimated based on other large probability surveys, and this uncertainty should be addressed as part of this procedure, as done by Dever and Valliant (2010).

Our goal is to perform Bayesian inference for the population values of y, given the known background variables x. Inference for $y|x$ can then be combined to get inference for the entire population or for subgroups of interest; this is the poststratification step. For example, if we are poststratifying a national poll into 4 ethnic categories, 4 age categories, 2 sex categories, 5 education categories, and 50 states, then the number of cells is $J = 4 \cdot 4 \cdot 2 \cdot 5 \cdot 50 = 8000$, and the population mean value of the outcome for white people in Alabama, for example, is the weighted average of $\mathrm{E}(y|x_j)$ over the 40 cells corresponding to that group.

If there were no survey weights and we could assume equal-probability sampling, we would simply regress y on x in the sample and then use the fitted models to make predictions (with uncertainty) for the rest of the population. The challenge is that the data are sampled with unequal probabilities. We use the notation p and p_{sample} for the distributions of the population and sample, respectively; that is, we are considering the items in the population to be drawn at random from an infinite superpopulation with distribution $p(y, x, w)$, so that the sample can be considered a draw from the distribution $p_{\text{sample}}(y, x, w) \propto p(y, x, w)/w$.

We handle the problem of unequal sampling probabilities by modeling the joint distribution of outcome and weights, following Skinner (1994), Beaumont (2008), Si et al. (2015), and Léon-Novelo and Savitsky (2019):

$$\begin{aligned} \text{Model for the outcome:} \quad & p(y|x, w, \theta) = p_{\text{sample}}(y|x, w, \theta) && (1) \\ \text{Model for the weights:} \quad & p(w|x, \phi) \propto w\, p_{\text{sample}}(w|x, \phi), && (2) \end{aligned}$$

where θ and ϕ represent the parameters in the outcome and weight models. Both models (1) and (2) are conditional on x, and we are assuming that x is known in the population. The advantage of the above formulation is that it makes clear how both models can be estimated from the sample

[2] We use the term "sampling" here to include all factors relating to inclusion in the sample, including nonresponse; see Rubin (1976) and Brick (2013).

data. Because we are doing this work using simulation, it can be thought of as a design-consistent and model-based approach to generating a synthetic population as proposed by Dong et al. (2014).

In effect, we are poststratifying on (x, w), which requires estimation of $p(w|x)$ so that we can construct the joint distribution of x and w in the population.

Three key aspects of this approach are:

- The outcome model (1), which, following the principles of MRP, can include many predictors x, the functions of w, and their interactions;
- The adjustment for the sampling weights in the transition from sample to population distributions in (2), which captures the adjustment information in the weights;
- The adjustment for w is performed using a model, rather than simply reweighting individual data points. Using a model allows the method to work with sparse data, using MRP, and the observed data are used to estimate a complete population distribution.

2.2. Quantities of interest

We consider several different inferential goals.

The *overall population mean* can be written as $\overline{Y} = \sum_{j=1}^{J} N_j \overline{Y}_j / \sum_{j=1}^{J} N_j$, where $\overline{Y}_j = \mathrm{E}(y|x_j)$ is the population mean within poststratification cell j. We are assuming an infinite or essentially infinte population so that there is no need for a finite-population correction.

The population means will never be known, so for Bayesian inference we can write, $\overline{Y} = \sum_{j=1}^{J} N_j \mathrm{E}(y|x_j, \theta, \phi) / \sum_{j=1}^{J} N_j$, so that uncertainty in the model parameters θ, ϕ induces uncertainty in $\overline{Y}$. Posterior simulations of θ, ϕ propagate to posterior simulations of the vector of $\overline{Y}_j$'s, which in turn propagate to posterior simulations of the population average, $\overline{Y}$.

This approach also applies to *individual poststratification cells* or *groupings of such cells*. For example, in political surveys there can be interest in estimating the average within a state, which uses the same poststratification formula as above, but just including a subset of cells.

Another goal of interest is the *population regression*, that is, the result that would be obtained by a regression of y on x fit to the entire population. This is a more general concept than trying to estimate a "true regression coefficient" because it does not rely on the assumption that the fitted regression is a correct generative model for the data.

We can estimate a population regression within our framework in two steps. First, we estimate the population average, $\overline{Y}_j = \mathrm{E}(y|x_j)$, within each poststratification cell j. Second, we fit a cell-level regression using these averages. For a linear model, we can simply fit weighted least squares to these J "data points," with point j getting a weight of N_j corresponding to its size in the population. For logistic regression, we can maximize a weighted pseudo-likelihood with $2J$ data points, with cell j being assigned an observation of 1 with weight $N_j \overline{Y}_j$ and an observation of 0 with weight $N_j(1 - \overline{Y}_j)$. In either case, we would propagate uncertainty as described above, performing these analyses for each posterior simulation of θ, ϕ, thus obtaining posterior simulations of the desired population regression.

The problem becomes more difficult if there is a desire to go beyond simple averages and proportions. For example, in the study of political polarization there is interest in estimating the *variation* within U.S. congressional districts: when considering swing districts, there is a big difference between a population of political moderates and a population divided evenly between strong partisans of the left and right. The only general way to estimate this variation is to simulate the entire distribution of responses, not just the mean, within each poststratification cell.

2.3. Inference

Before getting into details of Bayesian inference, uncertainty, and computation, let us consider how to fit (1) and (2) using point estimation. The first step is to regress y on x and w, yielding $p(y|x, w, \theta)$. The second step is to regress w on x, again using the observed data, thus yielding $p_{\text{sample}}(w|x, \phi)$. Here we are simply taking θ and ϕ as their point estimates. Next we convert from sample to population distribution,

$$p(w|x, \phi) = \frac{w\, p_{\text{sample}}(w|x, \phi)}{\int w\, p_{\text{sample}}(w|x, \phi) dw}. \tag{3}$$

This latter expression needs to be evaluated for each value of x in the population (that is, for all the poststratification cells), hence the integral in (3) must either be determined analytically or though some fast approximation. Conceptually, though, the problem is now solved: for each poststratification cell j, we determine $p(w|x_j, \phi)$ from (3) and then average over this distribution to get the predictive distribution of y in cell j:

$$p(y|x_j, \theta, \phi) = \int p(y|x_j, w, \theta) p(w|x_j, \phi) dw. \tag{4}$$

This integral can be derived analytically or else approximated. In any case, we now have estimated the population predictive distribution within each cell and can then poststratify by averaging over the assumed-known cell counts N_j in the population.

Bayesian inference is performed the same way, with the only difference being that inferential inference about θ and ϕ is propagated through (3) and (4). Here is a generic computational implementation:

1. Define a prior distribution for (θ, ϕ), $p(\theta)$ and $p(\phi)$, or use some form of a noninformative prior in the absence of any prior information on the parameter of interest. Informative priors should be possible in historical or pilot studies.

2. Fit the model $p(y|x, w, \theta)$ to the sample data; obtain posterior simulations $\theta^s, s = 1, \ldots, S$.

3. Given a prior distribution for ϕ, fit the model $p_{\text{sample}}(w|x, \phi)$ to the sample data; obtain posterior simulations $\phi^s, s = 1, \ldots, S$. If θ and ϕ share parameters or are dependent in their prior distribution, these two models would be fit together in one step.

4. For each draw (θ^s, ϕ^s):

   (a) For each poststratification cell j:

      i. Draw weights w^l, $l = 1, \ldots, L$, from $p_{\text{sample}}(w|x_j, \phi^s)$.

      ii. For each w^l, compute $\mathrm{E}(y|x_j, w^l, \theta^s)$ from the regression model.

      iii. Compute the weighted average $\widehat{Y}_j^s = \sum_{l=1}^L w^l \mathrm{E}(y|x_j, w^l, \theta^s, \phi^s) / \sum_{l=1}^L w^l$, which is a Monte Carlo estimate of $\mathrm{E}(y|x_j, \theta^s, \phi^s)$, the population mean within cell j under the model.

   (b) Compute the inferred population mean $\widehat{Y}^s = \sum_{j=1}^J N_j \widehat{Y}_j^s / \sum_{j=1}^J N_j$ and any subpopulation means or comparisons of interest.

5. Approximate the posterior distribution of all quantities saved in the previous step by their S simulations.

The workflow would then be continued with the usual steps of checking computational accuracy, model fit, and sensitivity, and altering or expanding the model as necessary (Gelman et al., 2013, 2020).

This approach should automatically give stable small-area estimates, as long as the factors defining the small areas are included in x, and as long as a rich enough set of models is used to fit regressions (1) and (2). Indeed, this is the main selling point of our approach, that it seamlessly performs weighting adjustment within a modeling context that allows small-area estimation and poststratification.

If there is interest in within-cell population summaries other than averages, then step 4(b) of the above algorithm must be made more general. Instead of simply computing a weighted average over the draws w^l, we can use Pareto-smoothed importance resampling to draw a subset $M < L$ of these weights with probabilities proportional to w^l (Vehtari et al., 2015). Collect the M resampled draws and renumber them as $w^m, m = 1, \dots, M$. These approximate a set of draws from the population model, $p(w|x_j, \phi^s)$. For each w^m, we can then continue by sampling one value y from the predictive distribution, $p(y|x_j, w^m, \theta^s)$. We can then complete the process by computing whatever summaries are desired using the M draws of y within that cell (including regression coefficients).

3. Implementation

We consider how to efficiently apply our method for the problem of estimating summaries based on population cell means; as discussed in Section 2.2, this should suffice for most applications of survey research, including estimating the mean of the outcome y in the population or subsets of the population determined by the predictors x, along with population linear and logistic regressions of y on x.

3.1. Closed-form solution with a lognormal or gamma model for the weights

The algorithm just described has a cumbersome nested design requiring a new draw of $w^1, \dots, w^L$ for each posterior draw of the model parameters, along with a potentially unstable weighted averaging step.

One way to speed the computation is to use a model for the weights where the denominator of (3) can be evaluated in closed form. One such model, proposed by Skinner (1994), is lognormal regression.

Suppose we define $v = \log w$ and fit the model, $p_{\text{sample}}(v|x) = \text{normal}(v|g(x, \beta), \sigma)$, where g is a function of x given parameter vector β, so that $\phi = (\beta, \sigma)$. Then (3) can be written as,

$$p(v|x, \phi) = \frac{e^v p_{\text{sample}}(v|x, \phi)}{\int e^v p_{\text{sample}}(v|x, \phi) dv}, \tag{5}$$

and we can simplify the expression that appears in the numerator and denominator:

$$\begin{aligned} e^v p_{\text{sample}}(v|x, \phi) &= e^v \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2\sigma^2}(v-g)^2} \\ &= \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2\sigma^2}((v-g)^2 - 2\sigma^2 v)} \\ &= \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2\sigma^2}((v-(g+\sigma^2))^2 - \sigma^4 - 2g\sigma^2)} \\ &= e^{g+\frac{1}{2}\sigma^2} \,\text{normal}(v \mid g + \sigma^2, \sigma), \end{aligned}$$

so that (5) becomes,

$$\begin{aligned} p(v|x,\phi) &= \frac{e^{g+\frac{1}{2}\sigma^2}\,\text{normal}(v\,|\,g+\sigma^2,\sigma)}{e^{g+\frac{1}{2}\sigma^2}} \\ &= \text{normal}(v\,|\,g+\sigma^2,\sigma). \qquad (6) \end{aligned}$$

Thus, under the lognormal model, the population distribution of the weights is identical to the sample distribution except that it is shifted to the right by σ^2. This makes sense. First, a large weight corresponds to more representation in the population, so we should expect higher weights to be more common in the population than in the sample. Second, σ^2 is the residual variance of the log weights, so the higher the value of σ, the more consequential will be the weighting (in terms of estimates, if the weights are correlated with the variable of interest, or in terms of precision, if the weights are independent of the variable of interest), hence the larger the shift. At the extreme of $\sigma = 0$, the weights do not vary within poststratification cells at all, and no adjustment is needed.

The above calculation took advantage of a conjugacy property of e^v with the normal density. Closed-form computation is available under other models as well. For example, if the weights follow a gamma regression, then multiplying the density function by w has the effect of adding 1 to the shape parameter of the model and correspondingly shifting the mean upward, so that if the mean of the distribution of $p_{\text{sample}}(w|x,\phi)$ in the sample is g, then the mean in the population distribution $p(w|x,\phi)$ becomes $\frac{\alpha+1}{\alpha}g$, which again makes sense, both in that it is an increase compared to the sample and that the increase goes to zero in the limit of $\alpha \to \infty$, which corresponds to a gamma distribution with zero variance.

3.2. Closed-form solution with a mixture of lognormals or gammas

For various reasons, the distribution of weights can be far from normal or gamma. But we can retain the clean computation of these conjugate forms using a mixture model. We demonstrate with the lognormal.

Start with the model, $p_{\text{sample}}(v|x,\phi) = \sum_{k=1}^K \lambda_k\,\text{normal}(v|g_k(x,\beta),\sigma_k)$, where g_k is a family of regression functions given parameter vector β, so that $\phi = (\beta,\lambda,\sigma)$. Similar algebra as before yields the population distribution,

$$p(v|x,\phi) = \frac{\sum_{k=1}^K \lambda_k\, e^{g_k(x,\beta)+\frac{1}{2}\sigma_k^2}\,\text{normal}(v\,|\,g_k(x,\beta)+\sigma_k^2,\sigma_k)}{\sum_{k=1}^K \lambda_k\, e^{g_k(x,\beta)+\frac{1}{2}\sigma_k^2}}, \qquad (7)$$

which again is a mixture of lognormals. In addition to each mean being shifted by σ_k^2 as before, the mixture proportions change, with components with higher values counting more in the population, which makes sense. For the poststratification, we will need to compute the mixture components in (7) for each poststratification cell, using the predictors x_j.

In practice it may be enough to model the weights using a lognormal or gamma error term or perhaps mixtures of one of these. But if a more general model is desired, it should be possible to get much of the computational benefits by first fitting the closed-form model and then using it as an approximation for the desired model. Instead of importance ratios w^l in step 4(b) of our algorithm, one would use the ratio of the exact and approximate densities, which should be more stable.

3.3. Example where the distribution of weighting-model residuals varies across poststratification cells

In our procedure, the adjustment for unequal-probability sampling is performed by reweighting the distribution of log weights v conditional on predictors x. Sections 3.1 and 3.2 considered models where the distribution of $v|x$ has a common form across postratification cells, with mean determined by a regression model and variance estimated based on the residuals from the regression fit to all the data. The variance of the residuals then determines the amount that the estimated distribution of weights need to be shifted upward to account for unequal-probability sampling.

But what if the variance of the weights itself varies across poststratification cells? In that case it is not appropriate to shift the log weights in all cells by the same amount; such a procedure can lead to a biased estimate, even asymptotically.

We demonstrate the problem using a simple hypothetical example of a survey with only two poststratification cells, women and men, coded as $x = 0$ and 1, respectively. We assume that women have been oversampled and comprise two-thirds of the sample. We further assume that the weights depend on various unobserved factors that vary more with men than with women, so that the log weights are higher and more variable for men than for women, following these distributions: $p_{\rm sample}(v|x\!=\!0)\!=\!{\rm normal}(v|0,0.5)$, $p_{\rm sample}(v|x\!=\!1)\!=\!{\rm normal}(v|0.5,0.8)$. (We have set these values so that the average weight for men in the sample is approximately twice that of the average weight of women: $e^{0+\frac{1}{2}(0.5)^2} = 1.13$; $e^{0.5+\frac{1}{2}(0.8)^2} = 2.27$.) To get the population distribution of $v|x$, we simply follow equation (6) and shift upward by the residual variance; thus, $p(v|x\!=\!0)\!=\!{\rm normal}(v|0.5^2,0.5)\!=\!{\rm normal}(v|0.25,0.5)$, $p(v|x\!=\!1)\!=\!{\rm normal}(v|0.5+0.8^2,0.8)\!=\!{\rm normal}(v|1.14,0.8)$.

Unfortunately, this is not the estimated distribution of $v|x$ obtained by fitting a regression model with pooled variance. In this case there is only a single predictor, and the regression picks up the sample mean of v within each sex and a pooled sample variance, which will be estimated as approximately $\frac{2}{3}(0.5)^2 + \frac{1}{3}(0.8)^2 = 0.38$, thus an estimated residual standard deviation σ of $\sqrt{0.38} = 0.62$, and so the inferred (but incorrect) population distribution of v is ${\rm normal}(0.38, 0.62)$ for women and ${\rm normal}(0.88, 0.62)$ for men.

These estimated distributions of log weights for the two sexes are much different from the true population distributions, even in the limit of large amounts of data. And this can have a devastating impact on inferences for any outcome y that is correlated with v. Suppose, for example, that y is height in centimeters, x is an indicator for being male, and the data are well fit by a model, $y = 161 + 6x + 7xv + \text{error}$. We have set up a scenario in which, conditional on sex, taller men (but not taller women) are less likely to be included in the sample (on average, they have higher weights) and have set up the numbers so that the average heights for women and men approximately correspond to known population averages: given the numbers above, ${\rm E}(y|x\!=\!0) = 161$ and ${\rm E}(y|x\!=\!1) = 161 + 6 + 7 \cdot 1.14 = 175$. If instead we use the estimated population distribution of v from the regression model with pooled variance, we get the wrong answer of ${\rm E}(y|x\!=\!1) = 161+6+7\cdot 0.62 = 171.3$. By using the pooled error distribution for v in this example, we greatly underestimated the variance of weights among men, which in turn reduced the shift when adjusting from sample to population distribution of ${\rm E}(v)$ in (6). What is distressing is that the error does not go away as sample size increases; that is, the estimate is inconsistent.

3.4. Extending the computation to allow the distribution of weighting-model residuals to vary across poststratification cells

We can address this problem by setting up the model for $v|x$ so that the variance as well as the mean varies with x, following ideas used by Maiti et al. (2014), Sugasawa et al. (2017), and Savitsky et al. (2022) for small-area estimation and adapting them to the weighting problem. The simplest approach is to extend the normal model of Section 3.1 as follows:

$$p_{\text{sample}}(v|x,\phi) = \text{normal}(v|g(x,\beta), h(x,\gamma)), \tag{8}$$

with separate regression models for the mean and variance, and with parameter vectors β and γ estimated from the data, and $\phi = (\beta, \gamma)$. A natural choice of parametric form would be linear on the location and log-linear on the scale:

$$p_{\text{sample}}(v|x) = \text{normal}(v|x\beta, e^{x\gamma}),$$

In any case, the distribution of the log weights would be shifted by the variances, as with (6); thus (8) yields,

$$p(v|x,\phi) = \text{normal}(v \mid g(x,\beta) + h(x,\gamma)^2, h(x)). \tag{9}$$

One could similarly alter the gamma and mixture models as well.

3.5. Weighted bootstrap of regression residuals

An alternative to modeling the distribution of regression residuals is to bootstrap them, resampling in proportion to the weights (Bertail and Combris, 1997; Cohen, 1997).

First consider the basic model with a shared error distribution across cells. We can then apply a weighted bootstrap to the full set of n residuals. If the residuals from the regression of v on x are $r_1, \ldots, r_n$, then for each poststratification cell j we sample L residuals with replacement from $\{r_1, \ldots, r_n\}$, with probabilities proportional to $\exp(r_i)$.

If the residuals average to zero, then, from Jensen's inequality, the distribution of residuals weighted by their exponentials has positive expectation. This makes sense: to estimate the population distribution we are oversampling the larger weights, so the expected value of the log weights should be greater for the population distribution than for the sampling distribution, and this should hold within each cell.

If the model for $\text{E}(y|x,v,\theta)$ is linear in v, we can then proceed simply by computing the mean of the distribution of resampled residuals, that is $\bar{r}_{\text{weighted}} = \sum_{i=1}^{n}(r_i e^{r_i})/\sum_{i=1}^{n} e^{r_i}$, hence we just plug in $\text{E}(v|x_j,\phi) + \bar{r}_{\text{weighted}}$ instead of $\text{E}(v|x_j,\phi)$ for v_j when estimating $\text{E}(y|x_i, v_j, \theta)$ within each poststratification cell.

If $\text{E}(y|x,v,\theta)$ is nonlinear in v, as with logistic regression, then we would want to estimate the expectation averaging over v, $\text{E}(y|x,\theta)$ using sampling, for each cell j imputing values for $v|x_j$ by taking the predicted value for the cell, $\text{E}(v|x_j,\phi)$, and then adding random draws from the residuals sampled using the above-described weighted bootstrap.

Next consider the more general case in which the error variance itself varies across cells. It would not work to simply bootstrap the residuals within each cell, as some cells have no data at all, thus no residuals to bootstrap—and it would be wrong to set the imputed residuals to zero for such cells, as this would underestimate the population cell mean, $\text{E}(v|x_j)$. In addition, if a cell has very few data points, then it will have very few residuals to booststrap, and the resulting adjustment wil be very noisy.

So it makes sense to do some sort of partial pooling between a bootstrap of within-cell residuals and a bootstrap of all the residuals in the data. For each cell j, this can be done using a weighted bootstrap of the residuals $r_i, i = 1, \ldots, n$ from the fitted regression of v on x, using the following rule:

$$\text{bootstrap weight for } r_i = \begin{cases} A\exp(r_i) & \text{if } i \text{ is within cell } j \\ \exp(r_i) & \text{otherwise.} \end{cases} \tag{10}$$

where A is some number greater than 1 that ensures that the residuals within the cell count more than those outside when imputing the population distribution of the error term for $v|x_j$.

We make the somewhat arbitrary decision to set

$$A = \frac{n}{30},$$

so that in a cell with $n_j = 30$, our bootstrap will give roughly equal total weight to the residuals within and outside the cell, cells with much fewer than 30 observations will mostly rely on the full sample of residuals, and cells with much more than 30 observations will mostly rely on the residuals within the cell. As n increases, all the cell sizes increase in expectation; thus, in the asymptotic limit, each cell's bootstrap is determined by the residuals within the cell, so that this part of the inference is consistent.

The reweighting in (10) will do for now, but it is an incomplete solution, for two reasons. First, it is a simple mix of hyperlocal (residuals within the single cell) and global (the entire sample). Ideally we would want a procedure closer to what is done in multilevel modeling, giving higher weights to residuals from cells that are close in the space of x. This should be possible—once we have established the general idea of reweighting, we can construct some modeling scheme that has the effect of giving the higher weights to observations whose predictors x are similar, using some distance measure, to those in the poststratification cell, without restricting to a simple in-or-out rule.

The second weakness of our reweighting approach is that it does not work with continuous predictors x, in which case there are no "poststratification cells"; there is just a poststratification list, a large matrix of predictors corresponding to some large pre-set population. In this case, the approach of (10) is meaningless. Again, we should be able to solve this problem by using a distance-based weighting scheme.

3.6. Integrated Bayesian computation

We can perform all the steps of regression and poststratification in a single probabilistic program when performing the weighting adjustment using a closed-form solution or bootstrap simulation. The computation goes as follows:

1. Specify models $p_{\text{sample}}(v|x, \phi)$ and $p(y|x, v, \theta)$ and estimate ϕ and θ together. Joint estimation of the two models would not, strictly speaking, be necessary if the parameter vectors ϕ and θ are distinct and independent in their prior distribution, but it is convenient to perform inference within the same probabilistic program so that we can work with posterior simulations from both of them together in the next step.

2. Loop through the poststratification cells $j = 1, \dots, J$: for each cell j, sample L draws v^l from the estimated or approximated $p(v|x_j, \phi)$ using the closed-form solution or weighted bootstrap. Propagate each simulated v^l through the regression model for the outcome variable to compute $\mathrm{E}(y|x_j, v^l, \theta)$, and then average over these to obtain a Monte Carlo estimate of $\mathrm{E}(y|x_j, \theta)$.

3. The result of the above steps is an $S \times J$ matrix of simulations representing the posterior distribution of the population mean in the J poststratification cells; these can be combined to get posterior estimates and uncertainties for the poststratified population mean or any subset of the population defined in terms of the predictors x.

4. Concerns

4.1. Unrealistic assumptions of the model

We call our method quasi rather than fully Bayesian because it is based on a generative model in which the weights w are defined in the population and are drawn to create the sample, but in real surveys the weights are constructed from the sample and do not have a population distribution to be estimated. In that way, our approach is similar to many applications of statistics in which a probability model is used even in the absence of any superpopulation or physical randomization (Little, 2004; Elliott and Valliant, 2017).

Our model assumes independent sampling with probabilities proportional to $1/w$, but survey weights are often constructed by raking and do not represent sampling probabilities at all. Even when weights are intended to represent inverse sampling probabilities, they generally do not, as the construction of weights is only approximate.

Why would we purposely construct a model that is wrong in these crucial ways? The short answer is that, to the extent that weights are well constructed in a practical sense and used as intended, an item with weight w in the sample is intended to represent w items in the population. Our procedure can be viewed as a smoothed version of applying weights to items in the sample. To the extent that we are building a model-based adaptation or generalization of existing practice, it makes sense to take the weights seriously and consider them as being inversely proportional to the probability of inclusion in the sample, even if they are not. Similarly, the assumption of independent sampling can be viewed as an instantiation of the recommended methods in which weights are attached to individual units. As we have written elsewhere, we fit a model consistent with standard practice because we want our approach to be an improvement upon rather than merely a replacement for standard weighted analysis of sample surveys. Similar ideas can be applied using non-Bayesian methods (Morikawa et al., 2022).

Another potential concern is the use of a regression, $p(w|x)$, that implies a continuous distribution of weights in the population, even though weights in real surveys typically take on only a finite possible number of values. In the past we have considered nonparametric modeling of survey weights (Si et al., 2015), but this adds enough complexity to the analysis that we have avoided it here. In practice, the lognormal or lognormal-mixture regression model used in the present paper should be fine: the lognormal regression should be a reasonable fit to weights that are constructed by multiplying many individual factors, mixture modeling can capture the discreteness that can arise if weights are dominated by one or two factors, and to the extent that the weights depend on variables in x, much of their variation will be explained by the deterministic part of the weighting regression anyway.

4.2. Sensitivity to large weights

As with weighting-based methods in general, we need to be concerned about the right tail of the weight distribution, for two reasons. First, survey weights are often smoothed or trimmed to reduce their variability, which can make sense as a variance-reduction tool but complicates their interpretation. Second, large weights correspond to lower probabilities of sampling or survey response, so they represent "dark matter" in the sampling procedure: potentially large chunks of the population that are expected to appear rarely or not at all in the data. Any resolution of this problem requires strong assumptions, such as a hard cap on the maximum weight in the population or a short upper tail that limits the total proportion of the population that would have large weights. In a finite-population analysis there is also a bound on the low end, because the probability of inclusion in the sample can never exceed 1. One advantage of the bootstrap-the-residuals procedure described in Section 3.5 is that this automatically bounds the weights.

When considering various aspects of sensitivity to model assumptions, remember that the goal is to estimate the population regression function, $p(y|x)$; the weights are just a means to this end, a way of adjusting for the biases that would occur if one were to attempt to extrapolate from a fitted model without adjusting for known discrepancies between sample and population. What is relevant, then, is the dependence of $p(y|x,w)$ on w. If this model is a smooth function of w, then approximating a discrete distribution of w by a continuous distribution might not cause serious problems. If the model behaves calmly for large values of w, then the "dark matter" problem of very large weights in the population might not be such a concern. It should be possible to do some theoretical analysis, looking at the tails of the model for w along with the functional form of $p(y|x,w)$ for large w to ensure bounded influence from the unobserved items with large weights. Working with deciles of the actual weight values may also be helpful for this kind of sensitivity analysis.

4.3. Weights that are negative, zero, or positive but very small

A dataset can include observations with zero weights, which is a signal to exclude them from analysis entirely. In our procedure it makes sense to remove these data points before beginning the analysis to avoid the awkward and otherwise unnecessary step of modeling a weight variable that can be zero or positive.

There are settings where negative weights can make sense as part of a regression adjustment (Ben-Michael et al., 2023), but these cannot be interpreted as inverse probability of inclusion in the sample, and so negative weights cannot fit into the methods used in this paper. In such settings, we would either restrict to data with positive weights or try to remove the steps within the weighting process that produced negative weights.

Finally, data points with extremely tiny weights will have essentially no effect on any weighted averages, but they can interfere with our estimation procedure by driving up the estimated variance of the weights in the population. One solution here is to fit a mixture model in which one of the components captures the low weights, thus effectively "quarantining" them so as not to contaminate inferences for population averages. It can also make sense to apply a simpler approach and just exclude such extreme cases.

5. Real and simulated-data examples

The appendixes give data and code for three examples: (A) Simulated data with a simple logistic regression model and a high correlation between the sampling weight and the outcome of interest; (B) Real data from an opinion poll with a multilevel linear regression and sampling weights; (C) Real data with a multilevel logistic regression.

6. Theoretical examples

We can work through some simple simulated-data examples to understand where the method works and where it breaks. Here are some ideas:

- No background variables x, only weights w, so the goal is to estimate the population mean. How does our approach compare to the simple weighted average?

- Weights w that depend entirely on the background variables x. Our approach should be identical to unweighted MRP.

- Simple stratified sampling with weights; then perform the analysis ignoring the strata. How does this differ from the standard stratified analysis? How does it differ from a weighted-average analysis ignoring the strata?

- Finite-population sampling including a certainty stratum and thus a hard lower bound on weights.

- Poststratification weights modeled as inverse-probability weights: how much does our approach inflate the variance estimate compared to the correct poststratification analysis?

- Simple small-area estimation without or with a group-level predictor. Result will depend on the dependence between weights and expected outcome, so try different possibilities in the simulations.

- Small-area estimation with a huge number of cells so $n_j = 0$ or 1 in almost all cells and there is no observable variation in weights within each cell, but the weights still matter.

- Weights that have been estimated for non-probability samples based on quasi-randomization or doubly robust approaches (Chen et al., 2020). How well does this procedure perform in this case in which the weights are only estimates of inverse probabilities of selection?

These examples raise conceptual challenges. Consider, for example, a national survey that is poststratified by geography, in such small areas (for example, zip codes) that there are no cells with more than one respondent in the sample. Also suppose that the survey weights are not based on geography but are instead based on the number of people living in the respondent's household, a variable that is not otherwise included in the analysis. The weights will still vary by geography even though they are not defined explicitly in geographic terms. But with only one respondent per poststratification cell in the data we cannot estimate the within-cell variance in log weights (the crucial parameter σ^2 in model (6)), and the only way forward, short of including the "number of people in the household" variable in the analysis, might be to combine cells to allow the estimation of within-cell variation of the weights.

Add some structure to the problem, though, and it becomes easier to solve. Take the same example, with the same number of poststratification cells, but suppose they are formed by the intersection of several variables, for example age, sex, ethnicity, education, and congressional district. In this case, a multilevel regression of log weight on these factors will yield a nonzero residual variance, as long as the model does not include the fully-saturated interaction of all the predictors.

7. Applications

We want as soon as possible to apply our method to live applications. Two examples we have immediately at hand are the Cooperative Election Study and Pew Research pre-election polls. Both these surveys include weights, and we have applied MRP to them in the past (Lei et al., 2017). We apply our method to the Cooperative Election Study in Appendices B and C but that is more of a demonstration and test case than a live example. We could also see how our method works when including post-election weighting based on vote preference.

We can also look into the Future of Families and Child Wellbeing Study, a survey of at-risk births for which we have been involved in construction of the weights (Kennedy and Gelman, 2021).

Evaluating the method in applied examples can be difficult because for most survey questions we do not know the true population values. One setting where we do know the truth, and which we have used to evaluate MRP in the past, is U.S. election polling; however, challenges arise there too given problems of differential nonresponse (Little and Gelman, 1998, Brick and Tourangeau, 2017, Kuriwaki et al., 2023).

That said, we still think much can be learned by applying our procedure to real problems. The method could run into computational difficulties, it could give completely unreasonable results, and the model could have problems fitting to the data. More positively, we could get a sense of distributions of weights in real surveys and compare different approaches to small-area estimation and regression modeling in the presence of survey weights.

We can also conduct simulation studies by subsampling from real survey data. In that case, the "population" (a large existing survey) is completely known, we have full control over the sampling procedure, we can define weights however we want, and we can compare our inferences to the population values, checking accuracy of estimates and coverage of uncertainty intervals.

8. Conclusion

The problem being attacked in this paper is to model an outcome y given predictors x from a sample whose data are collected with unequal probabilities, and also given inverse-probability weights w. We assume the joint distribution of x in the population is known. Our approach is to first estimate the w given x in the sample, then adjust (using the assumption of inverse-probability weighting) to estimate the distribution of w given x in the population. We then fit a model predicting y given x and w. The process is completed by averaging that fitted model over w to obtain the desired goal of an estimated distribution of y given x. This can then be averaged over the population distribution of x ("poststratification") to obtain inferences for the entire population or for subsets defined by x.

Our method goes beyond existing weighting methods by using a regression model for w given x, which allows us to escape the trap of what to do in small cells with only one or two observations. More generally, we can think of our procedure as a model-based adjustment for unequal sampling probabilities which plays well with MRP and other model-based approaches to small-area estimation.

Some future challenges include inference with known margins (poststratification with marginal or lower-dimensional joint distributions, for example post-election adjustments based on local vote totals); cluster sampling; inference for non-census variables (for example, religion); and generalization from sample to population in causal inference (Miratrix et al., 2013;O'Muircheartaigh and Hedges, 2014; Kennedy and Gelman, 2021).

For now, our practical recommendations depend on where you stand in the process of data preparation and analysis:

- If you have access to the raw data and relevant population information: We would not typically recommend the methods in this paper. Instead of creating weights and then incorporating them into the analysis, it should be better to just model the data directly conditional on all information that might go into weighting. This might require augmenting the poststratification table (if there are relevant non-census variables: information predictive of the outcome and predictive of inclusion in the sample that is not available at the population level), but that modeling could be done directly, with no need to create survey weights as intermediate quantities.

- If you have conducted a survey and want to create weights for others to use: In this case it could make sense to anticipate the methods discussed in the present paper when forming the weights. It could help future users of the survey if the weights contain relevant information to help the model-based analysis perform well. Some research is needed here, given that probabilities of inclusion in the sample are generally not known, only estimated, and also given that the goal is adjustment to the population, not estimation of inclusion probabilities.

- If you are analyzing a survey collected by others where the weights have been supplied: Here, we hope our theoretical and applied examples give some sense of when it would make sense to follow the approach presented here.

Our theoretical and practical challenge is to design a procedure that unifies existing design-based weighting methods and existing model-based approaches to small-area estimation. We will learn more when we try out the method on some examples.

References

Ansolabehere, S., Schaffner, B., and Luks, S. (2019). Guide to the 2018 Cooperative Congressional Election Survey. `https://cces.gov.harvard.edu/`

Bertail, P., and Combris, P. (1997). Bootstrap généralisé d'un sondage. *Annales d'Économie et de Statistique* **46**, 49–83.

Beaumont, J. F. (2008). A new approach to weighting and inference in sample surveys. *Biometrika* **95**, 539–553.

Ben-Michael, E., Feller, A., and Hartman, E. (2024). Multilevel calibration weighting for survey data. *Political Analysis* **32**, 65–83.

Bisbee, J. (2019). BARP: Improving Mister P using Bayesian additive regression trees. *American Political Science Review* **113**, 1060–1065.

Bollen, K. A., Biemer, P. P., Karr, A. F., Tueller, S., and Berzofsky, M. E. (2016). Are survey weights needed? A review of diagnostic tests in regression analysis. *Annual Review of Statistics and Its Application* **3**, 375–392.

Brick, J. M. (2013). Unit nonresponse and weighting adjustments: A critical review. *Journal of Official Statistics* **29**, 329–353.

Brick, J. M., and Tourangeau, R. (2017). Responsive survey designs for reducing nonresponse bias. *Journal of Official Statistics* **33**, 735–752.

Broniecki, P., Leemann, L., and Wüest, R. (2021). Improved multilevel regression with post-stratification through machine learning (autoMrP). *Journal of Politics* **84**, 597–601.

Chen, Ch., Duan, N., Meng, X. L., and Alegria, M. (2006). Power-shrinkage and trimming: Two ways to mitigate excess weights. *Proceedings of the Section on Survey Research Methods, American Statistical Association*, 2839–2846.

Chen, Ci., Wakefield, J., and Lumley, T. (2014). The use of sampling weights in Bayesian hierarchical models for small-area estimation. *Spatial and Spatio-temporal Epidemiology* **11**, 33–43.

Chen, Q., Elliott, M. R., Haziza, D., Sadju, Y., Ghosh, M., Little, R. J. A., Sedransk, J., and Thompson, M. (2017). Approaches to improving survey-weighted estimates. *Statistical Science* **32**, 227–248.

Chen, Q., Elliott, M. R., and Little, R. J. A. (2012). Bayesian inference for finite population quantiles from unequal probability samples. *Survey Methodology* **38**, 203–214.

Chen, Y., Li, P., and Wu, C. (2020). Doubly robust inference with nonprobability survey samples. *Journal of the American Statistical Association* **115**, 2011–2021.

Cohen, M. P. (1997). The Bayesian bootstrap and multiple imputation for unequal probability sample designs. Proceedings of the Survey Research Methods Section, American Statistical Association, 635–638.

Dever, J. A., and Valliant, R. (2010). A comparison of variance estimators for poststratification to estimated control totals. *Survey Methodology* **36**, 45–56.

Dong, Q., Elliott, M. R., and Raghunathan, T. E. (2014). A nonparametric method to generate synthetic populations to adjust for complex sampling design features. *Survey Methodology* **40**, 29–46.

DuMouchel, W. H., and Duncan, G. J. (1983). Using sample survey weights in multiple regression analysis of stratified samples. *Journal of the American Statistical Association* **78**, 535–543.

Elliott, M. R., and Little, R. J. A. (2000). Model-based alternatives to trimming survey weights. *Journal of Official Statistics* **16**, 191–209.

Elliott, M. R., and Valliant, R. (2017). Inference for nonprobability samples. *Statistical Science* **32**, 249–264.

Fay, R. E., and Herriot, R. A. (1979). Estimates of income for small places: An application of James-Stein procedures to census data. *Journal of the American Statistical Association* **74**, 269–277.

Gelman, A. (2007). Struggles with survey weighting and regression modeling (with discussion). *Statistical Science* **22**, 153–188.

Gelman, A. (2018). Regularized prediction and poststratification. Statistical Modeling, Causal Inference, and Social Science, 19 May. `https://statmodeling.stat.columbia.edu/2018/05/19/regularized-prediction-poststratification-generalization-mister-p/`

Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D., Vehtari, A., and Rubin, D. B. (2013). *Bayesian Data Analysis*, third edition. CRC Press.

Gelman, A., and Little, T. C. (1997). Poststratification into many categories using hierarchical logistic regression. *Survey Methodology* **23**, 127–135.

Gelman, A., and Little, T. C. (1998). Improving upon probability weighting for household size. *Public Opinion Quarterly* **62**, 398–404.

Gelman, A., Vehtari, A., Simpson, D., Margossian, C. C., Carpenter, B., Yao, Y., Kennedy, L., Gabry, J., Bürkner, P., and Modrák, M. (2020). Bayesian workflow. `https://arxiv.org/abs/2011.01808`

Ghitza, Y., and Gelman, A. (2013). Deep interactions with MRP: Election turnout and voting patterns among small electoral subgroups. *American Journal of Political Science* **57**, 762–776.

Gopelrud, M. (2024). Re-evaluating machine learning for MRP given the comparable performance of (deep) hierarchical models. *American Political Science Review* **118**, 529–536.

Graubard, B. I., and Korn E. L. (2002). Inference for superpopulation parameters using sample surveys. *Statistical Science* **17**, 73–96.

Haziza, D., and Beaumont, J. F. (2017). Construction of weights in surveys: A review. *Statistical Science* **32**, 206–226.

Heeringa, S. G., West, B. T., and Berglund, P. A. (2017). *Applied Survey Data Analysis*, second edition. CRC Press.

Holt, D., and Smith, T. M. F. (1979). Post stratification. *Journal of the Royal Statistical Society A* **142**, 33–46.

Kalton, G. (1983). Models in the practice of survey sampling. *International Statistical Review* **51**, 175–188.

Kennedy, L. A., and Gelman, A. (2020). Year 15 Fragile Families survey weight adjustment. Princeton Center for Research on Child Wellbeing. `https://fragilefamilies.princeton.edu/sites/g/files/toruqf2001/files/ff_const_wgtsy15.pdf`

Kennedy, L. A., and Gelman, A. (2021). Know your population and know your model: Using model-based regression and poststratification to generalize findings beyond the observed sample. *Psychological Methods* **26**, 547–558.

Kish, L., (1992). Weighting for unequal P_i. *Journal of Official Statistics* **8**, 183–200.

Kolenikov, S., West, B. T., and Lugtig, P. J. (2020). A checklist for assessing the analysis documentation for public-use complex sample survey data sets. *Survey Statistician* **81**, 50–62.

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

Kuriwaki, S., Ansolabehere, S., Dagonel, A., and Yamauchi, S. (2023). The geography of racially polarized voting: Calibrating surveys at the district level. *American Political Science Review.*

Lei, R., Gelman, A., and Ghitza, Y. (2017). The 2008 election: A preregistered replication analysis. *Statistics and Public Policy* **4** (1), 1–8.

Léon-Novelo, L. G., and Savitsky, T. D. (2019). Fully Bayesian estimation under informative sampling. *Electronic Journal of Statistics* **13**, 1609–1645.

Li, K., and Si, Y. (2024). Embedded multilevel regression and poststratification: Model-based inference with incomplete auxiliary information. *Statistics in Medicine* **43**, 256–278.

Little, R. J. A. (1991). Inference with survey weights. *Journal of Official Statistics* **7**, 405–424.

Little, R. J. A. (1993). Post-stratification: A modeler's perspective. *Journal of the American Statistical Association* **88**, 1001–1012.

Little, R. J. A. (2004). To model or not to model? Competing modes of inference for finite population sampling. *Journal of the American Statistical Association* **99**, 546–556.

Little, R. J. A. (2015). Calibrated Bayes, an inferential paradigm for official statistics in the era of big data. *Statistical Journal of the International Association for Official Statistics* **31**, 555–563.

Little, R. J. A., and Vartivarian, S., (2003). On weighting the rates in non-response weights. *Statistics in Medicine* **22**, 1589–1599.

Little, T. C., and Gelman, A. (1998). Modeling differential nonresponse in sample surveys. *Sankhya* **60**, 101–126.

Lohr, S. (2022). *Sampling: Design and Analysis*, third edition. London: CRC Press.

Lopez-Martin, J., Phillips, J. H., and Gelman, A. (2022). Multilevel regression and poststratification case studies. `https://bookdown.org/jl5522/MRP-case-studies/`

Lumley, T. (2004). Analysis of complex survey samples. *Journal of Statistical Software* **9** (8), 1–19.

Lumley, T., and Scott, A. (2017). Fitting regression models to survey data. *Statistical Science* **32**, 265–278.

Maiti, T., Ren, H., and Sinha, S. (2014). Prediction error of small area predictors shrinking both means and variances. *Scandinavian Journal of Statistics* **41**, 775–790.

Makela, S., Si, Y., and Gelman, A. (2018). Bayesian inference under cluster sampling with probability proportional to size. *Statistics in Medicine* **37**, 3849–3868.

Meng, X. L. (1994). Multiple-imputation inferences with uncongenial sources of input. *Statistical Science* **9**, 538–558.

Miratrix, L. W., Sekhon, J. S., and Yu, B. (2013). Adjusting treatment effect estimates by post-stratification in randomized experiments. *Journal of the Royal Statistical Society B* **75**, 369–396.

Morikawa, K., Terada, Y., and Kim, J. K. (2022). Semiparametric adaptive estimation under informative sampling. `https://arxiv.org/abs/2208.06039/`.

O'Muircheartaigh, C., and Hedges, L. V. (2014). Generalizing from unrepresentative experiments: A stratified propensity score approach. *Journal of the Royal Statistical Society C* **63**, 195–210.

Pfeffermann, D. (1993). The role of sampling weights when modeling survey data. *International Statistical Review* **61**, 317–337.

Pfeffermann, D. (2011). Modelling of complex survey data: Why model? Why is it a problem? How can we approach it? *Survey Methodology* **37**, 115–136.

Pfeffermann, D., Skinner, C., Holmes, D., Goldstein, H., and Rasbash, J. (1998). Weighting for unequal selection probabilities in multilevel models. *Journal of the Royal Statistical Society B* **60**, 23–40.

Potthoff, R., Woodbury, M., and Manton, K. (1992). "Equivalent sample size" and "equivalent degrees of freedom" refinements for inference using survey weights under superpopulation models. *Journal of the American Statistical Association* **87**, 383–396.

Rabe-Hesketh, S., and Skrondal, A. (2006). Multilevel modelling of complex survey data. *Journal of the Royal Statistical Society Series A: Statistics in Society* **169**, 805–827.

Rao, J. N. K. (2003). *Small Area Estimation.* New York: Wiley.

Rubin, D. B. (1976). Inference and missing data. *Biometrika* **63**, 581–592.

Rubin, D. B. (1996). Multiple imputation after 18+ years (with discussion). *Journal of the American Statistical Association* **91**, 473–520.

Särndal, C. E. (1978). Design-based and model-based inference in survey sampling (with discussion). *Scandinavian Journal of Statistics* **5**, 27–52.

Savitsky, T. D., Gershunskaya, J., and Crankshaw, M. (2022). Joint point and variance estimation under a hierarchical Bayesian model for survey count data. `https://arxiv.org/abs/2210.14366/`.

Si, Y., Pillai, N., and Gelman, A. (2015). Bayesian nonparametric weighted sampling inference. *Bayesian Analysis* **10**, 605–625.

Si, Y., Trangucci, R., Gabry, J., and Gelman, A. (2020). Bayesian hierarchical weighting adjustment and survey inference. *Survey Methodology* **46**, 181–214.

Skinner, C. J. (1994). Sample models and weights. *Proceedings of the Section on Survey Research Methods, American Statistical Association*, 133-–142.

Solon, G., Haider, S. J., and Wooldridge, J. M. (2015). What are we weighting for? *Journal of Human Resources* **50**, 301–316.

Stanek, E. J., and Singer, J. M. (2004). Predicting random effects from finite population clustered samples with response error. *Journal of the American Statistical Association* **99**, 1119–1130.

Su, Y. S., and Gelman, A. (2023). Who wants school vouchers in America? A comprehensive study using multilevel regression and poststratification. *Social Sciences* **12**, 430.

Sugasawa, S., Tamae, H., and Kubokawa, T. (2017). Bayesian estimators for small area models shrinking both means and variances. *Scandinavian Journal of Statistics* **44**, 150–167.

Vehtari, A., Simpson, D., Gelman, A., Yao, Y., and Gabry, J. (2015). Pareto smoothed importance sampling. `https://arxiv.org/abs/1507.02646/`.

Voss, D. S., Gelman, A., and King, G. (1995). Pre-election survey methodology: Details from nine polling organizations, 1988 and 1992. *Public Opinion Quarterly* **59**, 98–132.

West, B. T., and McCabe, S. E. (2012). Incorporating complex sample design effects when only final survey weights are available. *Stata Journal* **12**, 718–725.

West, B. T., Sakshaug, J. W., and Aurelien, G. A. S. (2016). How big of a problem is analytic error in secondary analyses of survey data? *PLoS One* **11**, e0158120.

Winship, C., and Radbill, L. (1994). Sampling weights and regression analysis. *Sociological Methods and Research* **23**, 230–257.

Xie, H., Barker, L. E., and Rolka, D. B. (2020). Incorporating design weights and historical data into model-based small-area estimation. *Journal of Data Science* **18**, 115–131.

A. Simulated-data example with code: logistic regression

We first demonstrate our method with a hypothetical marketing study. The population comprises the N users who visit a webpage during a month, a small fraction of whom are given the opportunity to buy a certain product. Each user i has a customer engagement score, x_i, and the probability π_i of user i being exposed to the promotion is determined by an algorithm that depends on x_i and some other factors. The distribution of x_i is known in the population, and we know the values of π_i for the n users in the sample. Label y_i as the binary outcome of whether person i in the sample buys the product. The goal is to estimate the proportion of people in the population who would buy the product, and also to learn how this probability varies with x.

A.1. Simulating the population and sample

We simulate data based on the following assumptions:

1. $N = 10^6$.
2. The engagement scores x_i in the population are uniformly distributed in the set $\{1, 2, \ldots, 10\}$.
3. The probability of inclusion in the sample is an increasing function of engagement score, but with variation: $\pi_i = 10^{-4} x_j \exp(\epsilon_j)$, where $\epsilon_j \sim \text{normal}(0, 0.8)$.
4. The probability of buying the product is an increasing function of the engagement score; beyond that, it increases with the probability of being in the sample, following the rule, $\Pr(y_i = 1) = \text{logit}^{-1}(0.9 + 0.1x_j + 0.5 \log \pi_j)$. We set the coefficients for x_j and π_j so that both would be relevant to the outcome, and then we set the intercept so that the buy rate in the sample is approximately 10%.
5. The survey is conducted, and the analyst is given the distribution of x in the population and the following information on the n respondents in the sample: the engagement score x, the response y, and a weight w which is proportional to $1/\pi$. The analyst is not given the values of π in the population.

Here is code to simulate the data and sampling probabilities in the populations:

```
library("arm")
library("rstanarm")
library("cmdstanr")
set.seed(123)
N <- 1e6
x <- sample(1:10, size=N, replace=TRUE)
pi <- 1e-4 * x * exp(rnorm(N, 0, 0.8))
y <- ifelse(runif(N) < invlogit(0.9 + 0.1*x + 0.5*log(pi)), 1, 0)
```

We calculate the quantity of interest, the true buy rate in the population:

```
> print(mean(y))
[1] 0.100
```

We then draw the sample:

```
in_sample <- (runif(N) < pi)
n <- sum(in_sample)
```

Here is the sample size:

```
> print(n)
[1] 771
```

We next prepare the data that would be available to the analyst:

```
w <- 1/pi                  # inverse probability of selection
w <- w/mean(w[in_sample])  # normalized to have mean 1 in the data
sample_data <- data.frame(x, y, w)[in_sample,]
```

And here is the poststratification table:

```
poststrat <- data.frame(x = as.numeric(names(table(x))), N = as.numeric(table(x)))
J <- nrow(poststrat)
```

We then label the cells in the sample data:

```
sample_data$cell <- NA
for (j in 1:J) {
  sample_data$cell[sample_data$x==poststrat$x[j]] <- j
}
```

Now it is the analyst's turn. The starting point is to compute the raw and weighted estimates from the sample:

```
> print(c(mean(sample_data$y), mean(sample_data$w*sample_data$y)/mean(sample_data$w)))
[1] 0.163 0.103
```

As expected, the raw estimate is off—in this case, it is too high, which makes sense because in this simulation, the probability π of inclusion in the sample is positively correlated with the outcome, y. The weighted average is fine as a point estimate of the population average, but the analyst is also interested in how the outcome varies with x, hence the need for a regression model that adjusts for the sampling probabilities.

A.2. Estimating the model of weights in the population

Uncertainty in our inferences for the population will come from two sources: the fitted data model (1) and the fitted weight model (2). There is also potential uncertainty in the N_j's but we are ignoring any imperfections in the poststratification in this paper.

To show the basic idea, we start with a point estimate for the model of weights in the population. We return later in this section to account for uncertainty in that part of the model.

We start with a lognormal regression for the weights, as described in Section 3.1.

```
sample_data$v <- log(sample_data$w)
fit_v <- lm(v ~ x, data=sample_data)
display(fit_v)
```

Here is the result:

```
            coef.est coef.se
(Intercept)  0.81     0.09  
x           -0.17     0.01  
---
n = 771, k = 2
residual sd = 0.82, R-Squared = 0.20
```

The residual standard deviation σ is estimated at 0.82, so, following the mathematics in Section 3.1, the estimated distribution of v in the population is shifted to the right by σ^2, yielding the estimated distribution, $v|x \sim \text{normal}(0.81 - 0.17x + 0.82^2,\, 0.82)$.

A.3. Estimating the model of $y|x, v$

We next use the sample data to estimate the regression of the outcome on the predictor x and the log weights, v. In this case the outcome is binary and we fit logistic regression. We perform Bayesian inference so that we will have posterior simulation draws that capture inferential uncertainty.

```
fit_y <- stan_glm(y ~ x + v + x:v, family=binomial(link="logit"), data=sample_data)
print(fit_v, digits=2)
```

This yields:

```
             Median MAD_SD
(Intercept) -3.09   0.43
x            0.15   0.06
v           -0.39   0.44
x:v         -0.02   0.05
```

In order to use this model to make predictions, we need the distribution of v, given x, which we estimated in Section A.2.

A.4. Averaging over the estimated population distribution of $v|x$

For each poststratification cell j, we take 1000 draws from the fitted distribution of log weights, w, and then pipe these through the uncertainty in the fitted model for $y|x, w$ as represented by the S draws from that posterior distribution:

```
sims_y <- as.matrix(fit_y)
S <- nrow(sims_y)
L <- 1000
Ey <- array(NA, c(S, J))
for (j in 1:J){
  x_j <- poststrat$x[j]
  v <- rnorm(L, coef(fit_v) %*% c(1, x_j) + sigma(fit_v)^2, sigma(fit_v))
  Ey_pop <- posterior_epred(fit_y, newdata=data.frame(x_j, v))
  Ey[,j] <- rowMeans(Ey_pop)
}
```

The result is `Ey`, an $S \times J$ matrix, which contains S draws from the posterior distribution of $\mathrm{E}(y|x_j)$ for the J cells.

We can look at the inferences for the cells individually:

```
Ey_cell_est <- colMeans(Ey)
Ey_cell_se <- apply(Ey, 2, sd)
```

This gives estimates and posterior standard deviations for the J cells.

We can also compute the posterior distribution of the poststratified population average:

```
Ey_poststrat <- (Ey %*% poststrat$N) / sum(poststrat$N)
cat(mean(Ey_poststrat), "+/-", sd(Ey_poststrat), "\n")
```

which yields,

```
0.095 +/- -.012
```

A.5. Accounting for uncertainty in the fitted model for the weights

With a bit more effort, we can propagate the uncertainty in the estimated distribution of weights. This approach could be especially useful when weights are being estimated for non-probability samples (Chen et al., 2020). This requires computing posterior simulations for the parameters in the regression model for v:

```
fit_v <- stan_glm(v ~ x, data=sample_data, refresh=0)
sims_v <- as.matrix(fit_v)
```

and then looping the predictive calculations over the S simulation draws:

```
Ey <- array(NA, c(S, J))
for (s in 1:S){
  for (j in 1:J){
    x_j <- poststrat$x[j]
    v <- rnorm(L, sims_v[s,1:2] %*% c(1, x_j) + sims_v[s,"sigma"]^2, sims_v[s,"sigma"])
    Ey_pop <- invlogit(sims_y[s,1:4] %*% rbind(1, x_j, v, x_j*v))
    Ey[s,j] <- mean(Ey_pop)
  }
}
```

The way the model is set up, the parameters of the two regression models are independent in the posterior distribution, so we could use any ordering of simulation draws when propagating uncertainty, as long as we are consistent. For simplicity in setting up the code we use the same ordering of the S draws from the two fitted models.

We can then summarize the simulations in `Ey` as before. In this case, the results are the same as before to two decimal places (0.095 ± 0.012), whcih implies that the uncertainty for these population summaries is dominated by the uncertainty in the fitted regression of y.

A.6. Fitting a regression of log weights with normal-mixture error term

In this case, the log weights were simulated from a regression model with normal errors, but in general we would not know this. Following Section 3.2, we fit a linear regression for $v|x$ with an error term that is a mixture of three normals. We do this in Stan, and here is the program, which we call `mixture.stan`:

```
data {
  int M;
  int N;
  int K;
  vector[N] v;
  matrix[N,K] X;
}
parameters {
  vector[K] beta;
  simplex[M] lambda;
  ordered[M] mu;
  vector<lower=0>[M] sigma;
  real<lower=0> log_sigma_0;
}
model {
  vector[N] Xbeta = X*beta;
  lambda ~ lognormal(log(1./M), 1);
  mu ~ normal(0, 10);
  sigma ~ lognormal(log_sigma_0, 1);
  sum(lambda.*mu) ~ normal(0, 0.01);
  for (n in 1:N){
    vector[M] lps = log(lambda);
    for (m in 1:M){
      lps[m] += normal_lpdf(v[n] | Xbeta[n] + mu[m], sigma[m]);
    }
    target += log_sum_exp(lps);
  }
}
generated quantities {
  real mu_total = sum(lambda.*mu);
  real sigma_total = sqrt(sum(lambda.*((mu - mu_total)^2 + sigma^2)));
}
```

The model includes weakly informative priors on the parameters of the mixture components, and the line "`sum(lambda.*mu) ~ normal(0, 0.01);`" serves as a soft constraint to pin the mean of the fitted mixture model to zero, which allows the intercept of the regression to have the same interpretation as before.

We then fit the model in R and print and extract the results:

```
mixture <- cmdstan_model("mixture.stan")
M <- 3
K <- 2
mixture_data <- list(v=sample_data$v, X=cbind(rep(1,n), sample_data$x), N=n, K=K, M=M)
fit_v_mixture <- mixture$sample(data=mixture_data, seed=123, chains=4, parallel_chains=4)
print(fit_v_mixture, max_rows=20)
```

Here is the output:

```
    variable    mean  median   sd  mad      q5     q95 rhat ess_bulk ess_tail
 lp__        -948.41 -948.02 2.81 2.71 -953.58 -944.57 1.00      990     1338
 beta[1]        0.81    0.81 0.09 0.09    0.66    0.97 1.00     2137     1624
 beta[2]       -0.17   -0.17 0.01 0.01   -0.19   -0.15 1.00     2162     1697
 lambda[1]      0.30    0.23 0.24 0.22    0.04    0.78 1.01      889     1023
 lambda[2]      0.38    0.34 0.25 0.29    0.05    0.82 1.00     1177     1075
 lambda[3]      0.31    0.24 0.24 0.23    0.04    0.78 1.01     1166     1585
 mu[1]         -0.46   -0.43 0.27 0.31   -0.95   -0.09 1.00      887     1466
 mu[2]         -0.03   -0.02 0.26 0.19   -0.49    0.42 1.01     1034     1399
 mu[3]          0.50    0.47 0.31 0.34    0.09    1.04 1.00     1341     1636
 sigma[1]       0.66    0.69 0.18 0.14    0.32    0.90 1.00      931      683
 sigma[2]       0.73    0.77 0.18 0.12    0.38    0.97 1.00      907      635
 sigma[3]       0.65    0.67 0.15 0.14    0.36    0.85 1.00     1607     1809
 log_sigma_0   -0.43   -0.43 0.61 0.63   -1.46    0.56 1.00     2672     2552
 mu_total       0.00    0.00 0.01 0.01   -0.02    0.02 1.00     3888     2852
 sigma_total    0.82    0.82 0.02 0.02    0.79    0.86 1.00     4134     2662
```

A.7. Averaging over the fitted normal-mixture regression model for log weights

Next we extract the relevant parameters from the simulations and, for each simulation draw s and each poststratification cell j, we then take L draws v from the reweighted mixture model (7) and, for each, compute the expected value of y:

```
lambda_v <- as.matrix(fit_v_mixture$draws("lambda", format="df"))[,1:M]
mu_v <- as.matrix(fit_v_mixture$draws("mu", format="df"))[,1:M]
beta_v <- as.matrix(fit_v_mixture$draws("beta", format="df"))[,1:K]
sigma_v <- as.matrix(fit_v_mixture$draws("sigma", format="df"))[,1:M]
```

```
Ey <- array(NA, c(S, J))
for (s in 1:S){
  for (j in 1:J){
    x_j <- poststrat$x[j]
    v_hat_mixture <- as.numeric(beta_v[s,] %*% c(1, x_j)) + mu_v[s,]
    lambda_new <- lambda_v[s,] * exp(v_hat_mixture + 0.5*sigma_v[s,]^2)
    m <- sample(1:M, L, replace=TRUE, prob=lambda_new)
    v <- rnorm(L, v_hat_mixture[m] + sigma_v[s,m]^2, sigma_v[s,m])
    Ey_pop <- invlogit(sims_y[s,1:4] %*% rbind(1, x_j, v, x_j*v))
    Ey[s,j] <- mean(Ey_pop)
  }
}
```

As before, we can average over the cells to get S simulations of the poststratified population mean and compute its posterior mean and standard deviation:

```
Ey_poststrat <- (Ey %*% poststrat$N) / sum(poststrat$N)
cat(mean(Ey_poststrat), "+/-", sd(Ey_poststrat), "\n")
```

which yields,

```
0.095 +/- 0.012
```

This is approximately the same result as before, which makes sense given that the data were simulated from a model with a normal distribution for errors, which is approximately recovered by the fit of a mixture of three normals.

A.8. Bootstrapping residuals from the model for log weights

Finally we apply the simpler and perhaps more robust approach of Section 3.5 to simulate $v|x$ using a weighted bootstrap of the residuals:

```
fit_v <- lm(v ~ x, data=sample_data)
Ey <- array(NA, c(S, J))
for (j in 1:J){
  x_j <- poststrat$x[j]
  r_boot <- sample(resid(fit_v), L, replace=TRUE,
                   prob=exp(resid(fit_v))*ifelse(sample_data$x==x_j, n/30, 1))
  v <- predict(fit_v, newdata=data.frame(x_j)) + r_boot
  Ey_pop <- posterior_epred(fit_y, newdata=data.frame(x_j, v))
  Ey[,j] <- rowMeans(Ey_pop)
}
```

As in Section A.4, we use the point estimate of the fitted model of $v|x$—this is implied by the use of the `predict()` function for v—in order to keep the code cleaner, because we found that propagating uncertainty in that part of the model did not have any noticeable impact on the final results.

As before, we then summarize to obtain a posterior distribution for the population mean:

```
Ey_poststrat <- (Ey %*% poststrat$N) / sum(poststrat$N)
cat(mean(Ey_poststrat), "+/-", sd(Ey_poststrat), "\n")
```

which yields,

```
0.096 +/- 0.012
```

A.9. Integrated Bayesian computation

We can follow the plan described in Section 3.6 and embed all the computation inside a single Stan program, which we call `normal_logit_weighting_bootstrap.stan`. As indicated by the name of the file, this program fits a linear regression, $p(v|x, \phi)$, and a logistic regression, $p(y|x, v, \theta)$, and then in uses a weighted bootstrap to adjust for the unequal sampling probabilities in the generated quantities block:

```
data {
  int N;  // Number of data points
  int K;  // Number of regression predictors
  int J;  // Number of poststratification cells
  array[N] int<lower=0,upper=1> y;    // Binary outcome
  array[N] int<lower=0,upper=J> cell; // Poststratification cells of data
  vector<lower=0>[N] w;      // Sampling weights (data with 0 or neg weights must be removed)
  matrix[N,K] X;             // Regression predictors (including constant term)
  matrix[J,K] X_poststrat;   // Regression predictors for poststratification cells
  vector[J] N_poststrat;     // Sizes of poststratification cells
  int L;  // Number of simulations for approximating p(v|x)
}
transformed data {
  vector[N] v = log(w);
  matrix[N,2*K] Xv = append_col(X, X .* rep_matrix(v, K));
  matrix[N,J] cell_indicator;
  for (n in 1:N){
    for (j in 1:J){
      cell_indicator[n,j] = cell[n]==j;
    }
  }
}
parameters {
  vector[K] b_v;    // Coefs for regression of v on X
  vector[2*K] b_y;  // Coefs for regression of y on X interacted with v
  real<lower=0> sigma_v;  // Residual sd of regression of v
}
transformed parameters {
  vector[N] E_v = X*b_v;
}
model {
  v ~ normal(E_v, sigma_v);
  y ~ bernoulli_logit(Xv*b_y);
}
generated quantities {
  vector[J] E_y_poststrat;
  real E_y_poststrat_mean;
  {
  vector[N] resid = v - E_v;
  vector[J] v_pred = X_poststrat * b_v;
  for (j in 1:J){
    vector[N] prob_boot = exp(resid) .* (1 + (N/30.0 - 1) * cell_indicator[,j]);
    prob_boot = prob_boot/sum(prob_boot);
    vector[L] resid_boot;
    for (l in 1:L) {
      resid_boot[l] = resid[categorical_rng(prob_boot)];
    }
    vector[L] v_sim = v_pred[j] + resid_boot;
    matrix[L,K] X_sim = rep_matrix(X_poststrat[j], L);
    matrix[L,2*K] Xv_sim = append_col(X_sim, X_sim .* rep_matrix(v_sim, K));
    vector[L] E_y_sim = inv_logit(Xv_sim * b_y);
    E_y_poststrat[j] = mean(E_y_sim);
  }
  E_y_poststrat_mean = sum(N_poststrat .* E_y_poststrat) / sum(N_poststrat);
  }
}
```

We run the Stan program from R:

```
integrated_boot <- cmdstan_model("normal_logit_weighting_bootstrap.stan")
integrated_data <- list(N=n, K=2, J=J, L=100, y=sample_data$y, w=sample_data$w,
  X=cbind(rep(1,n),sample_data$x), X_poststrat=cbind(rep(1,J),poststrat$x),
  N_poststrat=poststrat$N, cell=sample_data$cell)
integrated_boot_fit <- integrated_boot$sample(integrated_data, seed=123, chains=4,
  parallel_chains=4)
print(integrated_boot_fit, c("b_v","b_y","sigma_v","E_y_poststrat","E_y_poststrat_mean"),
  max_rows=30, digits=3)
```

When coding directly in Stan, this runs much faster than our earlier indirect approach piping the simulations of $v|x$ through the posterior prediction functions in rstanarm or brms. Here is the output:

```
          variable   mean median    sd   mad     q5    q95  rhat ess_bulk ess_tail
b_v[1]              0.816  0.815 0.094 0.092  0.663  0.975 1.003     2302     2517
b_v[2]             -0.173 -0.173 0.013 0.012 -0.194 -0.153 1.002     2362     2316
b_y[1]             -3.134 -3.123 0.441 0.437 -3.888 -2.458 1.000     1755     2110
b_y[2]              0.152  0.150 0.059 0.059  0.055  0.251 1.000     1746     2289
b_y[3]             -0.404 -0.396 0.441 0.446 -1.170  0.294 1.003     1498     1821
b_y[4]             -0.018 -0.020 0.055 0.055 -0.106  0.075 1.003     1425     1871
sigma_v             0.818  0.817 0.021 0.021  0.784  0.854 1.002     3506     2150
E_y_poststrat[1]    0.037  0.031 0.024 0.019  0.011  0.085 1.001     1416     1900
E_y_poststrat[2]    0.042  0.038 0.021 0.018  0.016  0.082 1.001     1455     1990
E_y_poststrat[3]    0.049  0.047 0.018 0.017  0.024  0.082 1.001     1572     2291
E_y_poststrat[4]    0.060  0.059 0.015 0.015  0.037  0.087 1.001     1807     2316
E_y_poststrat[5]    0.072  0.071 0.014 0.014  0.050  0.096 1.001     2332     2546
E_y_poststrat[6]    0.090  0.089 0.013 0.013  0.069  0.112 1.000     3500     2570
E_y_poststrat[7]    0.104  0.103 0.014 0.013  0.082  0.129 1.002     4366     3087
E_y_poststrat[8]    0.136  0.135 0.016 0.016  0.111  0.164 1.000     4232     3136
E_y_poststrat[9]    0.163  0.162 0.021 0.021  0.131  0.198 1.002     3495     3364
E_y_poststrat[10]   0.193  0.192 0.027 0.028  0.152  0.240 1.001     3129     3306
E_y_poststrat_mean  0.095  0.094 0.012 0.012  0.076  0.117 1.002     3066     2966
```

This gives us inference for both sets of regression parameters, the residual standard deviation of the regression of $v|x$, population averages within all the poststratification cells, and the poststratified population mean.

A.10. Estimating a population regression

Following the method explained in Section 2.2, we can use the posterior simulations of the poststratification cell averages to perform inference for the population regression of y on x. Here is the R code:

```
pop_coef <- array(NA, c(S, 2))   # 2 predictors (including the intercept)
cell_means <- integrated_boot_fit$draws(format="df", variables="E_y_poststrat")
for (s in 1:S){
  y_j <- as.numeric(poststrat$N * cell_means[s,])[1:J]
  fit <- glm(cbind(y_j, poststrat$N - y_j) ~ poststrat$x, family=quasibinomial(link="logit"))
  pop_coef[s,] <- coef(fit)
}
pop_coef_hat <- apply(pop_coef, 2, mean)
pop_coef_se <- apply(pop_coef, 2, sd)
```

The estimated intercept and slope are −3.67 and 0.22 with standard errors 0.58 and 0.07, respectively.

B. Real-data example with code: multilevel linear regression

We next show a small real-data example from our MRP case study (Lopez-Martin et al., 2022), which uses data from the 2018 Cooperative Congressional Election Study (Ansolabehere, 2019), a survey that included weights which for simplicity we had not included in our earlier case study. Here, we estimate opinion on abortion (using a composite response on a 0–6 scale which we treat as a continuous outcome), poststratifying on ethnicity, age, education, and state. The biggest difference between this and the previous example is that our goal here is inference for small areas—in this case, state-level estimates of abortion attitudes—rather than for the population average. We work with a random sample of 500 respondents so as to make the benefits of multilevel modeling more dramatic.

B.1. Simple unweighted and weighted MRP

First we set up R, read in the data, extract a subset, and renormalize the weights:

```
library("rstanarm")
source("setup_1.R")
set.seed(123)
n <- 500
subset <- sort(sample(nrow(df), n))
data <- df[subset,]
data$w <- data$w/mean(data$w)
```

We then fit the multilevel model with and without weights:

```
fit_unweighted <- stan_glmer(abortion ~ (1 | state) + (1 | eth) + (1 | educ) + (1 | age) + male +
  (1 | male:eth) + (1 | educ:age) + (1 | educ:eth) + repvote + (1 | region),
  data=data, cores=4)
fit_weighted <- stan_glmer(abortion ~ (1 | state) + (1 | eth) + (1 | educ) + (1 | age) + male +
  (1 | male:eth) + (1 | educ:age) + (1 | educ:eth) + repvote + (1 | region),
  weight=w, data=data, cores=4)
```

In addition to the poststratification variables, the model includes two state-level predictors: an indicator for region and the Republican vote share in the state in the 2016 presidential election, standardized to have zero mean and unit standard deviation among the 50 states.

We look at the fitted models to make sure they make sense:

```
> print(fit_unweighted)
            Median MAD_SD
(Intercept)  2.6    0.5
male        -0.1    0.4
repvote      0.4    0.1
```

```
Auxiliary parameter(s):
      Median MAD_SD
sigma 1.9    0.1

Error terms:
 Groups   Name        Std.Dev.
 state    (Intercept) 0.22
 educ:age (Intercept) 0.53
 educ:eth (Intercept) 0.46
 male:eth (Intercept) 0.64
 age      (Intercept) 0.36
 educ     (Intercept) 0.39
 region   (Intercept) 0.47
 eth      (Intercept) 0.66
 Residual             1.90

> print(fit_weighted)
stan_glmer
 family:       gaussian [identity]
 formula:      abortion ~ (1 | state) + (1 | eth) + (1 | educ) + (1 | age) +
   male + (1 | male:eth) + (1 | educ:age) + (1 | educ:eth) +
   repvote + (1 | region)
 observations: 500
------
            Median MAD_SD
(Intercept)  2.6    0.5
male        -0.1    0.4
repvote      0.1    0.2

Auxiliary parameter(s):
      Median MAD_SD
sigma 2.6    0.1

Error terms:
 Groups   Name        Std.Dev.
 state    (Intercept) 0.19
 educ:age (Intercept) 0.32
 educ:eth (Intercept) 0.42
 male:eth (Intercept) 0.50
 age      (Intercept) 0.51
 educ     (Intercept) 0.42
 region   (Intercept) 0.72
 eth      (Intercept) 0.56
 Residual             2.62
```

Next we poststratify each fitted model to obtain inferences for the 50 states, which we graph vs. Republican vote share in the previous election:

```
poststrat_state_graph <- function(Ey, poststrat_df, statelevel_predictors_df, y_range,
    ylab="", main="") {
  S <- nrow(Ey)
  states <- names(table(poststrat_df$state))
  n_state <- length(states)
```

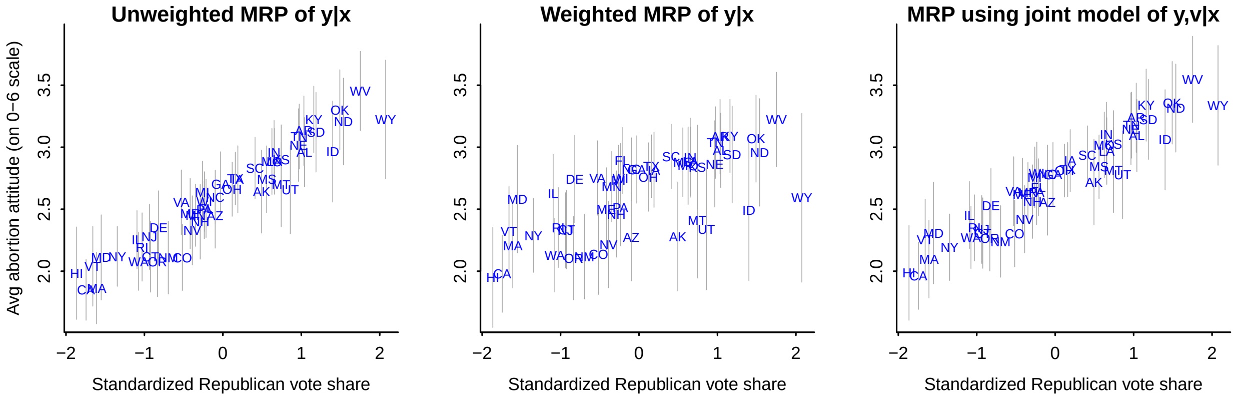

Figure 1: *Posterior estimates ±1 standard deviation for state-level opinion averages based on three different multilevel regression and poststratification (MRP) analyses fit to a sample of 500 respondents: (a) MRP applied to the unweighted data, (b) MRP using the weights as powers of the likelihood factors, (c) our recommended approach using a joint model for weights and outcomes.*

```
  mrp_state <- array(NA, c(S, n_state))
  for (i in 1:n_state){
    keep <- poststrat_df$state==states[i]
    mrp_state[,i] <- Ey[,keep] %*% poststrat_df$N[keep] / sum(poststrat_df$N[keep])
  }
  mrp_state_est <- apply(mrp_state, 2, mean)
  mrp_state_se <- apply(mrp_state, 2, sd)
  plot(range(statelevel_predictors_df$repvote), y_range,
    xlab="Standardized Republican vote share", ylab=ylab, bty="l", type="n", main=main)
  for (i in 1:n_state){
    lines(rep(statelevel_predictors_df$repvote[i], 2),
      mrp_state_est[i] + c(-1,1)*mrp_state_se[i], col="darkgray", lwd=.5)
  }
  text(statelevel_predictors_df$repvote, mrp_state_est, states, cex=.8, col="blue")
}
poststrat_state_graph(posterior_epred(fit_unweighted, newdata=poststrat_df),
  poststrat_df, statelevel_predictors_df, c(1.6,3.9),
  ylab="Avg abortion attitude (on 0-6 scale)", main="Unweighted MRP of y|x")
poststrat_state_graph(posterior_epred(fit_weighted, newdata=poststrat_df),
  poststrat_df, statelevel_predictors_df, c(1.6,3.9),
  ylab="", main="Weighted MRP of y|x")
```

The left and center plots on Figure 1 show the results. In this case, the difference between unweighted and weighted MRP inferences appear to come from estimation of the coefficients for region in the fitted models.

B.2. Estimating the model of weights in the population

We checked and this survey has no data with zero or negative weights, so we proceeded by fitting a regression of log weights on the predictors used in the MRP model:

```
data$v <- log(data$w)
fit_v <- stan_glmer(v ~ (1 | state) + (1 | eth) + (1 | educ) + (1 | age) + male +
  (1 | male:eth) + (1 | educ:age) + (1 | educ:eth) + repvote + (1 | region),
  data=data, cores=4)
print(fit_v)
```

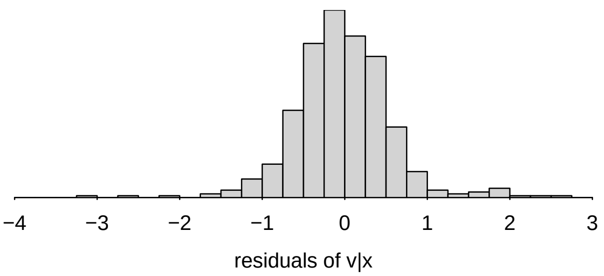

Figure 2: *Histogram of log weights for the MRP example. The distribution is unimodal with wider-than-normal tails.*

which yields,

```
            Median MAD_SD
(Intercept) -0.1    0.3
male         0.4    0.2
repvote     -0.1    0.0

Auxiliary parameter(s):
      Median MAD_SD
sigma 0.6    0.0

Error terms:
 Groups   Name        Std.Dev.
 state    (Intercept) 0.135
 educ:age (Intercept) 0.110
 educ:eth (Intercept) 0.305
 male:eth (Intercept) 0.252
 age      (Intercept) 0.078
 educ     (Intercept) 0.633
 region   (Intercept) 0.167
 eth      (Intercept) 0.265
 Residual             0.609
```

Figure 2 displays a histogram of the residuals from the regression of log weights. Recall that here we are working with a random sample of 500 survey respondents. When looking at the full survey with data from 60,000 people, the distribution of weights has a second mode, corresponding to less than 1% of the data, of very small weights, less than -8 on the log scale. These data points will be negligible in any weighted analysis; however, because of their distance from the large mass of the data, they would have some influence on the coefficients for v in the model of $y|x, v$, and we would recommend removing these cases with extremely low weights before proceeding.

B.3. Estimating the model of $y|x, v$

We begin the joint modeling by fitting a multilevel regression of the outcome given predictors, interacting everything with the log-weight variable, v:

```
fit_y <- stan_glmer(abortion ~ (1 + v | state) + (1 + v | eth) + (1 + v | educ) + (1 + v | age) +
  (1 + v | male:eth) + (1 + v | educ:age) + (1 + v | educ:eth) + v*repvote + (1 + v | region),
  data=data, cores=4)
print(fit_y)
```

which yields,

```
             Median MAD_SD
(Intercept)  2.6    0.3
v            0.1    0.3
male        -0.2    0.3
repvote      0.4    0.1
v:male      -0.3    0.4
v:repvote   -0.1    0.2

Auxiliary parameter(s):
      Median MAD_SD
sigma 1.9    0.1

Error terms:
 Groups   Name        Std.Dev. Corr
 state    (Intercept) 0.20
          v           0.32     -0.26
 educ:age (Intercept) 0.48
          v           0.29     -0.08
 educ:eth (Intercept) 0.39
          v           0.24     0.15
 male:eth (Intercept) 0.43
          v           0.35     0.15
 age      (Intercept) 0.22
          v           0.19     0.21
 educ     (Intercept) 0.22
          v           0.23     -0.01
 region   (Intercept) 0.30
          v           0.28     0.19
 eth      (Intercept) 0.40
          v           0.41     -0.02
 Residual             1.88
```

If there is interest, it would be possible to study the varying coefficients in this model to see which particular age categories, ethnic groups, ethnicities, regions, and states are associated with higher weights, which should comport with general understanding of which people are less likely to participate in surveys (Voss et al., 1995). Perhaps surprisingly, younger respondents do not have higher weights; perhaps the survey put in special effort to reach young people. The residual standard deviation of 1.89 is essentially unchanged from the 1.90 from the regression that did not include v (see the fitted model `fit_unweighted` on page 29), so in this case the weights are not highly predictive of the outcome. This is fine: our goal here is to demonstrate an approach that can be applied in general, not just to outcomes where the weighting makes a big difference.

B.4. Bootstrapping residuals from the model for log weights

We can simulate the distribution of v within each poststratification cell using a weighted bootstrap of residuals as described in Section 3.5. In this case, though, the model of $y|x, w$ is linear in v, hence all we need is $\mathrm{E}(v|x)$, so we can replace the bootstrap sample of residuals by the appropriate weighted average for each cell:

```
r <- resid(fit_v)
r_pop_mean <- rep(NA, J)
for (j in 1:J){
  in_cell <- data$state==poststrat_df$state[j] & data$eth==poststrat_df$eth[j] &
```

```
    data$educ==poststrat_df$educ[j] & data$age==poststrat_df$age[j] &
    data$male==poststrat_df$male[j]
  w <- exp(r)*ifelse(in_cell, n/30, 1)
  r_pop_mean[j] <- sum(w*r)/sum(w)
}
```

I need to fix this code to do the reweighting of residuals as shown in equation (10). — AG

We then pipe this thorough our fitted model to obtain posterior simulation draws of the expected outcomes within all J cells:

```
v_pop_mean <- colMeans(posterior_epred(fit_v, newdata=poststrat_df)) + r_pop_mean
Ey <- posterior_epred(fit_y, newdata=data.frame(poststrat_df, v=v_pop_mean))
```

And then we plot the inferences for the 50 states. The result is the rightmost plot of Figure 1.

B.5. Integrated Bayesian computation

We can follow the plan described in Section 3.6 and embed all the computation inside a single Stan program, as demonstrated for our earlier example in Section A.9. We have not done so here just because the resulting Stan program with all the multilevel components was getting tangled. It would make more sense to build upon the `rstanarm` or `brms` package, which convert multilevel regressions directly into Stan code, and then augment the resulting Stan program with a generated quantities block to perform the sampling of $v|x, \phi$, the calculation of $\mathrm{E}(y|x, v, \theta)$, and the poststratification, as with the Stan program in Section A.9.

C. Real-data example with code: multilevel logistic regression

Section B demonstrated our procedure in the context of MRP with a linear model fit to data from the 2018 Cooperative Election Study. We now illustrate a logistic model using a binary outcome from the same survey. The procedure is mostly unchanged, but at the end there is a challenge: $\mathrm{E}(y|x, v)$ is now a nonlinear function of v, so we cannot just summarize the distribution of $v|x$ by its expectation. Instead, within each poststratification cell j, we create 1000 simulations $v^l|x_j$, then compute $\mathrm{E}(y|x_j, v^l)$ for each, then average over these to compute a Monte Carlo estimate of $\mathrm{E}(y|x, v_j)$. All this is conditional on the parameter vector θ^s; we average over the distribution induced by the S simulations to compute posterior mean and standerd deviations. As before, we condition on a point estimate of ϕ, the parameters governing the model for $v|x$, because uncertainty in ϕ seems to be a very minor contributor to posterior uncertainty in $\mathrm{E}(y|x)$.

First we define a binary outcome and fit a multilevel logistic regression predicting it given x and v:

```
data$abortion3 <- data$abortion >= 3
fit_y_binary <- stan_glmer(abortion3 ~ (1 + v | state) + (1 + v | eth) +
  (1 + v | educ) + (1 + v | age) + v*male + (1 + v | male:eth) + (1 + v | educ:age) +
  (1 + v | educ:eth) + v*repvote + (1 + v | region),
  family=binomial(link="logit"), data=data, cores=4)
```

We then average over the fitted distribution of v within each poststratification cell. Because of the additional storage and computing requirements, we only simulate $L = 20$ values of v within each cell. We concatenate these into JL points to be predicted using the fitted model for $y|x, v$. Some annoying code is required to put these together as a matrix:

```
L <- 20
Ey_alt <- array(NA, c(S, J))
Ev0 <- colMeans(posterior_epred(fit_v, newdata=poststrat_df))
poststrat_rep <- NULL
v_rep <- NULL
for (l in 1:L){
```

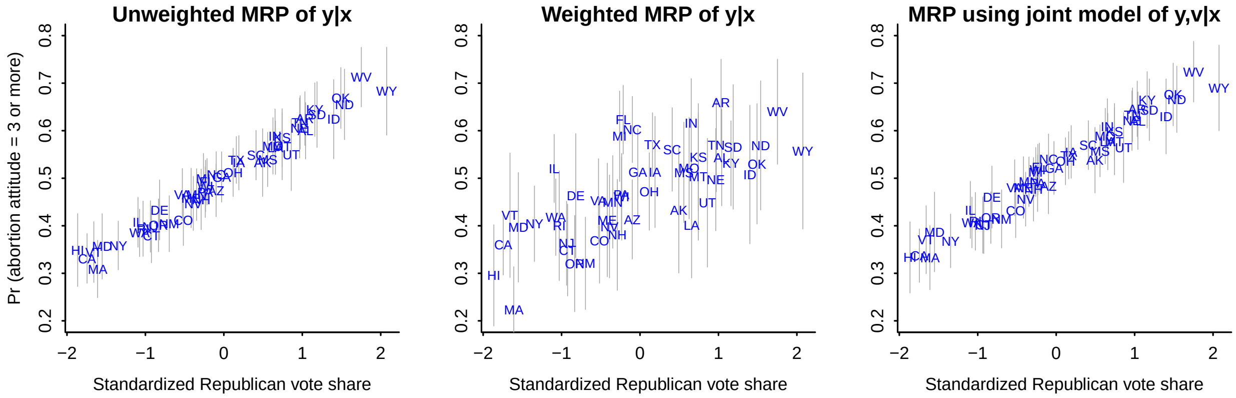

Figure 3: *Posterior estimates ±1 standard deviation for state-level opinion for a binary outcome based on three different multilevel regression and poststratification (MRP) analyses fit to a sample of 500 respondents: (a) MRP applied to the unweighted data, (b) MRP using the weights as powers of the likelihood factors, (c) our recommended approach using a joint model for weights as demonstrated for our earlier example in Section A.9. We have not done so here just because the re and outcomes.*

```
    poststrat_rep <- rbind(poststrat_rep, poststrat_df)
    v_rep <- c(v_rep, Ev0 + sample(r, L, replace=TRUE, prob=exp(r)))
  }
```

I need to fix this code to do the reweighting of residuals as shown in equation (10). — AG

We then compute S posterior draws of $\mathrm{E}(y|x)$ for the JL cells:

```
  poststrat_rep <- data.frame(poststrat_rep, v=v_rep)
```

We then average the L points within each poststratification cell to obtain an $S \times J$ matrix representing S posterior draws of average opinion within the J cells:

```
  Ey_pop <- posterior_epred(fit_y_binary, newdata=poststrat_rep)
  for (j in 1:J){
    indexes <- j + (0:(L-1))*J
    Ey_alt[,j] <- rowMeans(Ey_pop[,indexes])
  }
```

From there, we can poststratify within states to get S posterior draws of average opinion within the 50 states. Figure 3 shows the result, compared to MRP and weighted MRP of the data. As with the continuous-data models in Figure 1, the joint model gives similar results to the unweighted analysis. More generally, though, the weights can make a difference, in which case we want to apply them appropriately.

As discussed in Section B.5, the best way to proceed would be to embed all the computation inside a single Stan program, extending `rstanarm` or `brms` to perform the necessary sampling of v, inference for $\mathrm{E}(y|x, v, \theta)$, and poststratification in a generated quantities block, at which point the results could be plotted as above.