kpop: A kernel balancing approach for reducing specification assumptions in survey weighting*

Erin Hartman† Chad Hazlett‡ Ciara Sterbenz§

May 15, 2022

Abstract

With the precipitous decline in response rates, researchers and pollsters have been left with highly non-representative samples, relying on constructed weights to make these samples representative of the desired target population. Though practitioners employ valuable expert knowledge to choose what variables, X must be adjusted for, they rarely defend particular functional forms relating these variables to the response process or the outcome. Unfortunately, commonly-used calibration weights—which make the weighted mean X in the sample equal that of the population—only ensure correct adjustment when the portion of the outcome and the response process left unexplained by linear functions of X are independent. To alleviate this functional form dependency, we describe kernel balancing for population weighting (*kpop*). This approach replaces the design matrix $\mathbf{X}$ with a kernel matrix, $\mathbf{K}$ encoding high-order information about $\mathbf{X}$. Weights are then found to make the weighted average row of $\mathbf{K}$ among sampled units approximately equal that of the target population. This produces good calibration on a wide range of smooth functions of X, without relying on the user to explicitly specify those functions. We describe the method and illustrate it by application to polling data from the 2016 U.S. presidential election.

Keywords: balancing weights, calibration, non-response, survey weighting

*The authors thank Avi Feller, Kosuke Imai, Luke Miratrix, Santiago Olivella, Alex Tarr, Baobao Zhang and participants in the UCLA Causal Inference Reading Group. This work is partially supported by a grant from Facebook Statistics for Improving Insights and Decisions. The `kbal` package for the `R` computing language implements this method and is freely available.

†Assistant Professor, Department of Political Science, University of California, Berkeley (corresponding). Email: ekhartman@berkeley.edu URL: http://www.erinhartman.com.

‡Assistant Professor, Departments of Statistics and Political Science, University of California Los Angeles (corresponding). Email: chazlett@ucla.edu URL: http://www.chadhazlett.com.

§PhD Candidate, Department of Political Science, University of California Los Angeles.
Email: cster@ucla.edu.

1 Introduction

In an era of decreasing response rates, social scientists must rely on methods to adjust for the non-representative nature of survey samples. For example, Pew Research Center saw response rates to live-caller phone surveys decline from nearly one third of respondents in the late 1990s, to only 6% in 2018 (Kennedy and Hartig, 2019). The non-random nature of this "unit non-response" poses serious challenges for survey researchers and has led to greater use of non-probability sampling methods, such as panel, quota, or river sampling for online surveys (Mercer et al., 2017). The concern, whether due to non-response or non-probability sampling, is that the resulting survey respondents are not representative of the target population about which a researcher aims to draw an inference, leaving the potential for significant bias in estimates of target outcomes. Survey researchers address this concern through the use of survey design and analysis methods, and particularly, survey weighting.

Consequently, it is not so much a matter of *if* but rather *how* researchers will employ survey weights. To construct these weights, researchers must answer two important questions: (1) what variables will the researcher account for in their weighting procedure, and (2) how will they incorporate these variables in the construction of survey weights? For example, researchers have determined pollsters' failure to account for education level in survey weighting resulted in inaccurate predictions leading up to the 2016 U.S. Presidential election. Even those that did account for education level often failed to account for low levels of Midwestern, white voters with lower levels of educational attainment, i.e. the interaction of region, race, and education level (Kennedy et al., 2018). We return to this issue in our application and demonstrate how our proposed method can address these concerns.

In the survey calibration framework, which we work within, we observe multivariate covariates X and choose a feature mapping (e.g. expansion), $\phi(X)$. The calibration weights for the sampled units are chosen such that the weighted average of $\phi(X)$ in the sample equals the average in the desired target population. These weights are then applied to the outcome of interest in the sample in order to estimate the expected average outcome in the target population. It has long been known that the unbiasedness of the resulting estimates depend on assumptions regarding the linearity of the outcome or the response process in the terms $\phi(X)$ (see e.g. Särndal and Lundström, 2005; Kott and Chang, 2010), with analogous results in the causal inference setting (see e.g. Zhao and

Percival, 2016).[1] In practice, only simple feature mappings are typically employed, such as mean calibration on the observed covariates, which uses $\phi(X) = X$, or post-stratification which uses binary indicators for intersectional strata membership defined by X. Unfortunately, except in the case of full saturation (i.e. every combination of X values can be represented by an indicator), investigators are not generally in a position to argue that the outcome or response process are indeed linear in $\phi(X)$, leaving little reason to expect that such approaches would produce unbiased results. Further, though the careful investigator may be concerned that X alone may not be rich enough to support the linearity assumption and seek to explicitly include higher-order functions of X, the number of possibly important higher-order terms is typically far too large to accommodate. Leaving the choice of what higher-order terms to include in the hands of investigators, however, allows almost unlimited researcher degrees of freedom.

How then can researchers correctly choose what functions of covariates, i.e. the mapping $\phi(\cdot)$, they should use for calibration? Answering this question requires first clarifying the conditions for survey weights to unbiasedly estimate the desired outcome among the target population. We formulate the "linear ignorability" assumption, which states that survey weights unbiasedly estimate the desired outcome among the target population only when the part of the outcome not explained by a linear combination of the $\phi(X)$ is independent of the part of the sampling process not explained linearly by $\phi(X)$ within a suitable link function. This is similar in spirit to, though more general than, existing results that call for both non-parametric ignorability of selection and linearity of the outcome (or selection model) in $\phi(X)$.

Second, we propose kernel balancing population weighting (*kpop*) for weighting samples to match target populations. Delaying technical details, this approach uses a kernel to replace the original design matrix ($\mathbf{X}$) with a higher-dimensional kernel representation ($\mathbf{K}$), whose linear span captures a wide range of smooth, non-linear functions. We then find calibration weights such that the weighted average row of $\mathbf{K}$ in the sample is approximately equal to the average row of $\mathbf{K}$ in the target population. Weights chosen in this way can be understood as those that make the weighted distribution of $\mathbf{X}$ in the survey, as would be estimated by a kernel density estimator, approximately equal to that of the population. The *kpop* procedure thus offers one reasoned approach

[1] By "response" we mean that a participant was both sampled and responded, hence the "response process" regards the probability that a unit is in the observed sample.

to automatically account for important higher-order terms, but with minimal user intervention.

This approach builds on a long history of related work survey weighting. The challenges we seek to manage regarding common survey weighting techniques, particularly post-stratification and mean calibration, are well-known in the survey weighting literature (e.g. see Kalton and Flores-Cervantes, 2003, Berinsky, 2006, Hartman and Levin, 2019). Recent methods aim to parsimoniously address trade-offs between these approaches, as well as the relationship between mean calibration and inverse propensity score weighting (Linzer, 2011; Ben-Michael et al., 2021). Variable selection for weighting has addressed one aspect of feature selection (Chen et al., 2019; McConville et al., 2017; Caughey and Hartman, 2017). Here we desribe an approach that helps to reduce user discretion in the related problem of deciding what features and functions of observed covariates must be made to look similar in the sample and target population.

Finally, survey methodology and causal inference face closely related challenges: adjusting a control group to match a treated group on observables (causal inference), or adjusting a survey sample to match a target population (survey weighting). Consequently, both have benefited from the exchange of ideas and techniques. Analogous to work on survey weighting discussed below (e.g. Särndal and Lundström, 2005; Kott and Chang, 2010), recent work on calibration in the context of causal inference (i.e. covariate balancinng weights) has explored double robustness properties whereby results may be unbiased if either the outcome or selection (treatment assignment) model is (link) linear in the calibrated $\mathbf{X}$ (Zhao and Percival, 2016). We further describe a sufficient condition for unbiasedness of calibration weights in surveys, related to these ideas. Further, our approach adapts kernel-based estimation procedures that have analogously been deployed to weight control groups to treated groups in the causal inference literature to survey weighting (Wong and Chan, 2018; Yeying et al., 2018; Hazlett, 2020; Kallus, 2020; Tarr and Imai, 2021).

In what follows, Section 2 establishes notation, overviews our proposal, provides background on related approaches, and discusses assumptions. Section 3 offers our proposal in full, with our application to the 2016 U.S. Presidential election in Section 4 in which we show *kpop* effectively addresses concerns about important interactions in survey weights without requiring foreknowledge of these interactions from researchers. Simulation evidence is left to Appendix D. Section 5 discusses and concludes with limitations and future work.

2 Background

2.1 Notation

We consider a finite target population of N_{pop} units index by $i = 1, 2, \dots, N_{pop}$ and a smaller sample of survey respondents drawn (potentially with bias) from this target population consisting of only $N_s << N_{pop}$ units.[2] The indicator R_i is equal to 1 for units in the survey sample and zero otherwise. The outcome, Y_i, is available for the N_s units in the survey sample, but not for units in the target population. By contrast, the P auxiliary variables encoded in X are available for all units. When describing potential biases we will also make use of an *unobserved* factor Z. The finite target population is a collection of the form $\{X_i, Z_i, Y_i, R_i\}_{i=1}^{N_{pop}}$, with each tuple drawn independently from a common joint density $p(X, Z, Y, R)$ describing the data generating process or super-population. The auxiliary data initially encoded as X may be mapped to a richer feature expansions $\phi(X)$, with $X \mapsto \phi(X)$ from $\mathbb{R}^P \mapsto \mathbb{R}^{P'}$. In the survey setting, typically many or all dimensions of X are categorical, as in education levels, party identification, etc. We will consider such cases here, though the methods described are equally natural for continuous variables.

While inferences about $p(\cdot)$ (such as $\mathbb{E}[Y]$) may be of ultimate interest, because the target population is of finite size (e.g. all registered voters in a country), in our setting the quantity of usual interest is the mean outcome among the N_{pop} units in the target population, $\mu = \frac{1}{N_{pop}} \sum_i Y_i$. In expectation across the finite sample populations that may be drawn, $\mathbb{E}[\mu] = \mathbb{E}[Y]$. Nevertheless in a given case the goal is to estimate μ over the finite target population by taking a weighted average of the outcomes observed among the survey respondents using weights w to account for non-representative sampling.[3]

[2] In Appendix A we also consider the case in which the smaller survey sample is drawn separately, with bias, from the super-population, rather than as a subset of those already drawn in the fixed target population. This makes little difference analytically or practically.

[3] Much of the survey literature focuses on estimating the target population total of Y. Estimating the mean, however, is often of direct interest, such as in the examples here regarding vote choice. Thus we focus on mean estimation, which only slightly changes the expressions we use by introducing a factor of $1/N_{pop}$ or expressing the target moments differently. Note that we drop the subscript i in some cases to reduce notation, since each X_i (for example) is a random variable drawn from the same distribution ($p(X)$).

2.2 Overview of the proposal

Here we offer a brief overview of our proposal and its motivating logic, after which the remainder of the paper develops these ideas with greater detail and rigor.

When there are too many unique values of X to allow fully non-parametric adjustment (see *post-stratification*, below), the usual alternative is to choose functions of the data, $\phi(X)$, and find weights that equate the sample to the target on the means of these features (see *calibration* below). The sufficiency of these weights for unbiasedness depends upon the choice of $\phi(\cdot)$. As we will describe (see *linear ignorability*), unbiasedness requires that the part of the outcome and response models that cannot be systematically explained by linear functions of $\phi(X)$ be independent of each other. The defensibility of such assumptions is greater when the functions linear in $\phi(X)$ capture a wide range of flexible, plausible forms. To this end, kernel balancing for population weights (*kpop*) employs a kernel-based expansion corresponding to a choice of $\phi(\cdot)$ whose linear span includes a very wide range of smooth, flexible, non-linear function surfaces. Practically, this replaces the covariate data for unit i, X_i, with an N_s-dimensional representation, K_i, formed using a Gaussian kernel. Approximate calibration is then conducted to obtain weights on the sample units that make the weighted average K_i among the sample nearly equal to that among the population,

$$\sum_{i:R=1} w_i K_i \approx \frac{1}{N_{pop}} \sum_j K_j, \;\; s.t. \sum_i w_i = 1, \; w_i > 0, \; \forall i$$

Our approach notes that the worst-case bias resulting from approximation can be decomposed as the remaining imbalance (difference-in-means after weighting) on each (left) singular vector of $\mathbf{K}$, scaled by its singular value, where $\mathbf{K}$ is the matrix whose rows contain K_i for every unit in the sample and population, repeating those that appear in both. Thus, the top singular vectors of $\mathbf{K}$ are most important to calibrate precisely. In light of this result, we achieve approximate overall balance by seeking nearly exact balance on the first r singular vectors of $\mathbf{K}$, choosing r so as to minimize the worst-case approximation bias.

2.3 Non-parametric ignorability and post-stratification

One route to addressing unit non-response requires asserting that the observed covariates that will be used for adjustment (X) are *sufficient*, meaning that conditionally on X, units that do and do not respond have the same distribution of Y. We will refer to this assumption as non-parametric ignorability of the response conditionally on the observed covariates used for adjustment (Little and Rubin, 2019),

Assumption 1 (Non-parametric ignorability of response)

$$Y \perp\!\!\!\perp R \mid X$$

Assumption 1 is valuable because it asserts that within each stratum defined by $X = x$, the mean of Y for sample units is an unbiased estimator of the mean of Y for the target population among units. All that remains is to average these strata-wise means together according to how often each stratum of X was found in the target population. That is, while interest is in μ, ignorability allows us to identify $\mathbb{E}[\mu] = \mathbb{E}[Y]$ for a discrete-valued X using the observed sample according to

$$\mathbb{E}[Y] = \sum_{x \in \mathcal{X}} \mathbb{E}[Y|X = x]p(X = x) = \sum_{x \in \mathcal{X}} \mathbb{E}[Y|X = x, R = 1]p(X = x) \quad (1)$$

This is the *post-stratification estimand*, which can be estimated analogously by replacing the conditional expectations with conditional means and the population proportions ($p(X = x)$) with the proportion of the finite target population having $X = x$. Note that Appendix A considers how the finite nature of the target population and difference between μ and $\mathbb{E}[\mu] = E[Y]$ affects the total error and our definition of bias, with slightly different implications when the sample is nested in the target population than when it is drawn separately.

While post-stratification is attractive because it is fully justified by non-parametric ignorability, its feasibility is severely limited by the (one-sided) *positivity* assumption, namely $\Pr(R = 1 \mid X = x) > 0$ for all x found in the target population. It is thus undefined when non-empty strata in the population are empty in the sample. This becomes especially problematic when X contains any continuous variable or many multi-level categorical variables. We demonstrate such a limitation in our applied example below.

2.4 Calibration

When post-stratification is infeasible, investigators most often turn to calibration estimators, which construct weights on the sampled units such that the weighted mean of $\phi(X)$ among the sample equals the mean of $\phi(X)$ among the target population. In general form, calibration weights are defined as:

$$\min_{w} \quad D(w, q) \tag{2}$$

$$\text{subject to} \quad \sum_{i:R_i=1} w_i \phi(X_i) = T, \tag{3}$$

$$\sum_{i:R_i=1} w_i = 1, \text{ and } \; 0 \leq w_i \leq 1. \tag{4}$$

where q_i refers to a reference or base weight, $D(\cdot,\cdot)$ corresponds to a distance metric, typically greater for weights that diverge more severely from q_i.[4] The vector T describes target population moment constraints based on the mapping $\phi(X)$ – typically (and in our case), this is an average of $\phi(X)$ in the target population.

Common types of survey weighting correspond to different distance metrics $D(\cdot,\cdot)$, and are closely related to generalized regression estimation (Särndal, 2007). We use $D(w, q) = \sum_{i:R=1} w_i log(w_i/q_i)$, commonly employed in "raking" methods and variably known as exponential tilting (Wu and Lu, 2016), maximum-entropy weighting, or entropy balancing (Hainmueller, 2012). Where available, researchers should let q_i be the design weight defined by the survey sampling scheme. This will ensure that the weights are chosen to deviate as little as possible from the initial design weights.[5] Other distance metrics may also be used. For example, $D^{el}(w, q) = \sum_{i:R=1} \{q_i log(q_i/w_i) - q_i + w_i\}$, corresponds to the generalized pseudo empirical likelihood maximization (Wu and Lu, 2016). For broader reviews of calibration, see Särndal (2007), Caughey et al. (2020), or Wu and Lu (2016). The constraints in Equation (4) jointly ensure the weights fall on the simplex. We make this choice here as we wish to regard the weights as being probability-like. Relaxing this constraint allows for "extrapolation" beyond the support of the respondent sample, which increases the possibility

[4] More properly $D(\cdot,\cdot)$ is a divergence, not a distance, since it is often not symmetric in its arguments. For convenience of language we nevertheless refer to it as a distance metric, particularly since the choice of q is almost always fixed while we vary only the choice of w.

[5] Often the design weights for the sampling strategy are unknown or unavailable, in which case we typically let $q_i = 1/N_s$ be the uniform base weight for units in the respondent sample.

of severe model dependency but is employed in some techniques such as generalized regression estimators (Deville and Särndal, 1992).

The condition to be met by calibration weights is,

Condition 1 (Mean calibration constraints on $\phi(X)$) *Weights are constructed such that*

$$\sum_{i:R=1} w_i \phi(X_i) = \frac{1}{N_{pop}} \sum_{j=1}^{N_{pop}} \phi(X_j).$$

An additional assumption, which we will violate, regards the feasibility of weights that meet Condition 1. While our results are general to different choices of $D(\cdot,\cdot)$, feasibility may depend on this choice. We therefore continue under the specific case of maximum entropy weights. The *feasibility* assumption is then,

Assumption 2 (Feasibility of mean calibration on $\phi(X)$) *A set of weights that are non-negative, sum to one (i.e. Equation (4) holds), and minimize $D(w,q) = \sum_{i:R=1} w_i log(w_i/q_i)$, exist subject to the moment constraints defined in Condition 1.*

Assumption 2 can fail if the sample and target population are so different that balance cannot be achieved by weights with these constraints.[6] This concern is exacerbated as the dimensionality of $\phi(X)$ increases, a concern that will shape the strategy we propose.

Finally, the average outcome among the target population, $\mu = \frac{1}{N_{pop}} \sum_i Y_i$, is then estimated as

Estimator 1 (Calibration Estimator for Target Population Mean of Y)

$$\hat{\mu} = \sum_{i:R_i=1} w_i Y_i$$

with weights chosen subject to Condition 1 and Assumption 2.

In principle, calibration is a general and powerful tool given the flexibility of the choice of $\phi(X)$. In practice, however, most applications of calibration simply seek to match the means of X in the sample to that of the population, i.e. $\phi(X) = X$. We refer to this below as *mean calibration*. Such

[6]We note that Assumption 2 replaces the positivity assumption above required with non-parametric ignorability. This is clarifying, for example, when we have continuous variables in X, and common support is at best a theoretical proposition.

an approach holds intuitive appeal since, at minimum, pollsters and investigators seek to adjust a sample to closely match a target population on the margins, particularly on variables such as the proportion falling in some demographic or partisan group.[7]

2.5 Sufficiency of $\phi(\cdot)$: Linear ignorability

While no statistical method can make up for a poor choice of X, the choice of $\phi(\cdot)$ remains of great consequence (e.g. Särndal and Lundström, 2005; Pew Research Center, 2018). What is required of $\phi(\cdot)$? In broad terms, Särndal and Lundström (2005) note that a good feature mapping is one that (1) predicts patterns of non-response, (2) explains variation in the outcome, and (3) accounts for subgroups of interest in the data. Here we ask more precisely what is necessary of $\phi(\cdot)$ to reduce or eliminate bias. To answer this, we introduce the linear ignorability assumption, special cases of which correspond to existing forms that may be more convenient to reason about.

When non-parametric conditioning (i.e. post-stratification) is not an option and we are forced to adopt other estimation strategies (i.e. calibration), non-parametric ignorability (Assumption 1) is no longer enough to ensure unbiasedness; we must import additional or different assumptions in keeping with the parametric nature of calibration. The approach we take replaces the initial non-parametric ignorability assumption with a parametric one we refer to as linear ignorability,

Assumption 3 (Linear ignorability in $\phi(X)$) *Let* $Y_i = \phi(X_i)^\top \beta + \epsilon_i$ *and* $Pr(R_i = 1|X_i) = g(\phi(X_i)^\top \theta + \eta_i)$ *where* $g(\cdot) : \mathcal{R} \mapsto [0,1]$. *Linear ignorability holds when* $\epsilon_i \perp\!\!\!\perp \eta_i$.

In words, this requires the part of Y not linearly explainable by (i.e. orthogonal to) $\phi(X)$ to be independent of the part of the response process not linearly explained by $\phi(X)$ via a suitable link function.[8] This can be understood as an exclusion restriction: no part of the (residual) explanation for the response process can appear in (the residual part of) the outcome process.

Our first result states that under the linear ignorability assumption, mean calibration is unbiased for the target population mean.

[7] We note also that where post-stratification is feasible, mechanically it can be implemented either by constructing strata-wise probabilities $\frac{p(x)}{p(x|R=1)}$, or as a special case of calibration in which we (i) construct $\phi(X)$ to contains indicators for every intersectional stratum of X, then (ii) calibrate the survey sample to the target population using the distance metric $D^{\chi^2}(w,q) = \sum_{i:R=1}(w_i - q_i)^2/(q_i)$.

[8] Note that in our usage, ϵ is simply the component of Y orthogonal to $\phi(X)$, whereas in other settings it may represent "influences on Y other than those linear in $\phi(X)$." Accordingly, we state $Y_i = \phi(X_i)^\top \beta + \epsilon_i$ without loss of generality, simply decomposing Y into what can be linearly explained by $\phi(X)$ and a residual, orthogonal component.

Proposition 1 (Unbiasedness Mean Calibration under Linear Ignorability) *Under linear ignorability in $\phi(X)$ (Assumption 3) and feasibility of mean calibration on $\phi(X)$ (Assumption 2), the calibration estimator ($\sum_i w_i Y_i$) will be unbiased for the target population mean, μ, where the w are constructed using constraints described in Condition 1 for mean calibration on $\phi(X)$.*

Proof of Proposition 1 can be found in Appendix A.

Remarks and special cases

Linear ignorability replaces non-parametric ignorability with a statement of the "parametric ignorability" sufficient for unbiasedness, imposing a requirement on the relationship of the residuals from a selection and outcome model.

The main intuition for linear ignorability is that if calibration provides us with the same mean values of $\phi(X)$ in the sample and target population, then the part of the outcome Y that can be explained linearly by $\phi(X)$ will have the same mean regardless of the choice of β. The question is then whether the residual part of Y, ϵ, is related to the response process. If ϵ is independent of the response indicator, R, that is sufficient. Further, ϵ need not be independent of "all of R", but only the part not linearly "explainable" by $\phi(X)$, given by η. Note that η can be understood as the component of $g^{-1}(Pr(R=1|X))$ orthogonal to $\phi(X)$.

This brings us to two special cases, each requiring non-parametric ignorability plus parametric conditions on either the outcome or response process. First, is *linearity of the outcome*: if the outcome is linear in $\phi(X)$, then under non-parametric ignorability, ϵ is independent of η (and even R). Second is *link-linearity of the response model*: if $g^{-1}(Pr(R=1|X))$ is linear in $\phi(X)$ such that the residual η is random noise, then under non-parametric ignorability, η is independent of ϵ (and even Y). While either case alone is slightly stronger than linear ignorability, these are useful cases for investigators to reason with. Such assumptions are found in prior work on calibration such as Särndal and Lundström (2005); Kott and Chang (2010); Zhao and Percival (2016), though the choice of link functions in these is sometimes more restrictive than the general case we show here (see Appendix A.1).

2.6 Counterexample and bias

To clarify the commitments one makes by subscribing to the linear ignorability assumption, we illustrate how it might be violated. Consider the decomposition of Y as $\phi(X)^\top\beta + Z + \nu$ in which ν is random noise, i.e. independent of $\{X, Z, R\}$. Z is unobserved and, without loss of generality, orthogonal to $\phi(X)$ because it could equivalently be replaced by the residual from projecting Z onto $\phi(X)$. Whether linear ignorability holds is determined by Z's role in the selection process. If Z has variance zero or is random noise (independent of all other variables), then $\epsilon = Z + \nu$ will be independent of R, satisfying Assumption 3. By contrast, if Z is associated with R, then excepting knife-edge cases, $\epsilon = Z + \nu$ is also associated with R, violating Assumption 3.

More specifically, Appendix A shows that the bias due to any Z is given by $\mathbb{E}[wZ \mid R = 1] - \mathbb{E}[Z]$, i.e. *the difference between the weighted average of Z in the respondent sample and the average of Z in the target population.*[9] If Z is associated with R, then this bias is non-zero. However if Z is independent of R (and η) then (i) $\mathbb{E}[wZ|R = 1] = \mathbb{E}[Z|R = 1]$ because the weights are a function of X and R, they will be independent of Z, leaving $\mathbb{E}[Z|R = 1] = \mathbb{E}[Z]$. Hence the bias of the mean calibration estimator for μ will be zero.

Problematic variables Z could take on two forms. First, there could be important omitted variables. Unobserved factors outside of $\phi(X)$ could be relevant to both R and to Y, thus entering into both ϵ and η, causing them to be correlated. Such a case would also violate non-parametric ignorability (Assumption 1). For example, an individual's general level of interest in politics is predictive of many policy positions, and the strength of those preferences, in American politics. It is also highly predictive of response probability to political surveys, with those interested in politics over represented in respondent samples. Because political interest is not measured in many datasets used to define target populations, such as those defined by administrative records, it is an example of an unmeasured confounder Z that could violate both non-parametric ignorability and linear ignorability.[10] No adjustment technique fully eliminates bias in this scenario, but sensitivity analyses

[9] Appendix A describes the difference when the sample is drawn (non-randomly) from the super-population without being a subset of the target population. This does not change the bias expression; only the finite sample performance and the scaling defined into the weights.

[10] We emphasize that it is only the part of political interest orthogonal to the linear relationship with the included auxiliary variables in $\phi(X)$. To the degree that the included auxiliary covariates also adjust for imbalance in political interest, and mitigate bias from an unobserved Z, it is only the remaining part of bias from such an omitted confounder that is of concern.

provide a natural approach to addressing potential remaining bias from such confounders (e.g. Soriano et al., 2021).

Second, and our focus here, Z could simply represent a non-linear function of $\phi(X)$ that is relevant to both Y and R. Z would then appear in both ϵ and η, driving their association. This is of particular concern for the commonly used mean calibration in which $\phi(X) = X$, not accounting for higher-order interactions. This form of Z is difficult to rule out: investigators may suspect the outcome to "involve" X, but can rarely make strong arguments for the functional relationship to R and Y, or justify a particular link function for R. It is this concern, and the importance of selecting a sufficiently complex $\phi(X)$, that motivates the *kpop* method.

Example. We provide a simple example to illustrate the biasing potential of such a Z. Suppose a target population of interest consists of four groups in equal shares based on college education and self-identified gender: college-educated female, non-college-educated non-female, college-educated non-female, and non-college-educated non-female. A given policy happens to be supported by 80% of college-educated females, but only 20% of those in the other groups. Thus, the mean level of support in the target population should be 0.25(0.8)+0.75(0.2) = 35%. Further suppose that the sample is designed to carefully quota on gender and education, obtaining 50% female and 50% college-educated respondents. Thus, the sample already matches the target population exactly on these two margins.[11] Because attention was not given to the joint distribution of these two variables, however, among sampled females, three-quarters were college-educated (rather than half) and similarly among non-females, again three quarters were college-educated (rather than half) as displayed in Table 1. Consequently, the average level of support for this policy in the sample appears to be 42.5% rather than 35%,[12] generating a biased estimate despite perfectly matching the target population on both margins.

To meet Assumption 3, we would require $\phi(X)$ = [female, college, female & college]. If we instead use only $\phi(X) = X$ = [female, college], the problematic Z is the omitted interaction term. In this case, sampling occurred such that the two variables in X already have the same mean in the sample

[11] We use quota sampling in this example for simplicity as it allows us to have a sample already matched to the target population on the margins. The same considerations would apply, however, in a convenience sample or more generally if weighting were required to achieve mean calibration. See Caughey et al. (2020) for more details.

[12] College educated females would be three quarters of one half (3/8) of the sample and support the policy at 80%. Everybody else supports it at 20%. Mean support in the sample is thus (3/8)(0.8) + (5/8)(0.2) = 42.5%.

as in the target population making the calibration weights simply $1/N_s$ for all units. The response probability, even within gender, depends on college education, which influences the probability of response through η_i. These strata also have different mean levels of support (Y) indicating this Z remains a part of ϵ_i. This shows how the interaction is an example of a Z that violates linear ignorability, driving an association of ϵ and R.

Table 1 compares the post-stratification weights using all four categories of female $\times$ college as strata. These weights can be constructed simply as the population proportion of each stratum divided by the sample proportion. We compare this to mean calibration on X. For illustrative purposes, we have designed this example with only two binary categories so that post-stratification is feasible and therefore yields an unbiased estimate. The challenge we seek to address is that they are infeasible in many or most practical scenarios with many more intersectional strata. If successful, *kpop* would produce weights similar to those generated here by post-stratification, while also handling situations in which post-stratification is infeasible or where investigators cannot be expected to know *ex ante* what interactions or other non-linear functions of the original X are essential to include in $\phi(X)$.

Characteristics		Proportions		Outcome	Weights (times N_s)		
female	college	target population	sample	Pr(support)	unweighted	mean cal.	post-strat
1	1	1/4	3/8	0.80	1	1	2/3
1	0	1/4	1/8	0.20	1	1	2
0	0	1/4	1/8	0.20	1	1	2
0	1	1/4	3/8	0.20	1	1	2/3
		Target Population Average:		0.35			
		Weighted Average:			0.425	0.425	0.35

Table 1: Comparison of common choices of $\phi(X)$. Quota sampling ensured the sample was representative on the means of college and female. College-educated respondents are over-represented in the sample both among females and non-females, leading to a failure of mean calibration on X.

3 Proposal: Satisfying linear ignorability through kernel-based population weights (*kpop*)

The goal of *kpop* is to reduce the risk and magnitude of violating linear ignorability, with minimal user-intervention, by replacing the design matrix $\mathbf{X}$ with a kernel matrix $\mathbf{K}$ that represents a rich choice of $\phi(\cdot)$. The non-linear functions balanced by this approach will be higher-order transformations or interactions between covariates which might be explanatory of both Y and $Pr(R = 1|X)$, including Z of the non-linear type postulated above. For example, in our previous example, the goal is to have $\phi(X)$ account for much or all of the Z that was unaccounted for by mean calibration on X, i.e. the interaction between female and college. In a setting with continuous variables, a non-linear function of one or several of the observed covariates could play a similar role. Importantly, even if such a missing function of X is not entirely captured by a linear combination of $\phi(X)$, the residual variance left unexplained may be greatly reduced, shrinking the remaining bias.

3.1 A kernel-based choice of ϕ: $\mathbf{K}_i$

Many reasonable proposals are possible for how to choose $\phi(X)$ so as to achieve linear ignorability or at least reduce bias as far as possible. In plain terms, we want $\phi(X)$ to capture any (potentially non-linear) relationship between Y and X and/or R and X. With the X well accounted for in either model by a good choice of $\phi(X)$, we expunge problematic "Z" variables from ϵ and/or η, so that such a Z does not drive an association of ϵ with η.

The role of kernels in choosing $\phi(\cdot)$ emerges in the context of regularized, linear models. We could thus proceed by considering linear functions of $\phi(X)$ that explain either the outcome or the response probability (transformed as $g^{-1}(Pr(R = 1|X))$) to be linear in $\phi(X)$. For simplicity we consider the outcome. Concretely, consider the notional regularized regression problem:

$$\underset{\theta \in \mathbb{R}^{P'}}{\arg\min} \sum_i (Y_i - \phi(X_i)^\top \theta)^2 + \lambda \theta^\top \theta \tag{5}$$

This model is notional because it will not actually be estimated – it is simply a problem statement that allows us to arrive at the kernel based approach to building basis functions, $\phi(X)$. Ideally our choice of $\phi(X)$ would be one that includes very general, high-dimensional, non-linear expansions of X_i. Fortunately, certain choices of $\phi(X)$ can be high- or infinite-dimensional, yet admit an N_s-dimensional representation of the data that can then be employed in calibration. Specifically, kernel functions "compare" two observations, X_i and X_j, with the function of $k(X_i, X_j) : \mathbb{R}^P \times \mathbb{R}^P \mapsto \mathbb{R}$. For kernels that are known as "positive semi-definite",[13] the value of $k(X_i, X_j)$ corresponds to

[13]A kernel function $k(\cdot, \cdot)$ is positive semi-definite if the kernel matrix it creates, $\mathbf{K}$, satisfies $a^\top \mathbf{K} a \geq 0$ for all real

choices of $\phi(X_i)$ through the relationships $k(X_i, X_j) = \langle\phi(X_i), \phi(X_j)\rangle$. The Gaussian kernel, which we use, corresponds to an infinite-dimensional choice of $\phi(\cdot)$, such that as the number of sample points goes to infinity, every continuous function will be linear in these features (Micchelli et al., 2006).

The connection between a kernel function, $k(\cdot, \cdot)$, that compares pairs of observations, and feature map $\phi(\cdot)$ such that $k(X_i, X_j) = \langle\phi(X_i), \phi(X_j)\rangle$ may seem irrelevant at first. This linkage is greatly beneficial, however, in that, for such kernels, the solution to the loss minimization problem in (5) is also the minimizer of

$$||Y - \mathbf{K}c||^2 + \lambda c^\top \mathbf{K}c \tag{6}$$

where c is a vector of choice coefficients, and $\mathbf{K}$ is the kernel matrix with entries $\mathbf{K}_{i,j} = k(X_i, X_j)$. While this model is notional, we can understand this matrix to be constructed using the sample data, producing an N_s by N_s kernel matrix, whose columns represent a one-dimensional similarity measure between each row observation and each sample unit in the columns. When we move to apply this kernel approach toward weighting purposes, we create additional rows representing target population units over the same set of bases, again capturing similarity with sample units in the columns .

The vital feature of this result is simply that *the functions linear in $\phi(X_i)$ have been replaced with those linear in K_i*, regardless of the dimensionality of $\phi(\cdot)$. Thus, to gain all the benefits of $\phi(\cdot)$ – whether for modeling or calibration purposes – one need only work with its kernel transformation K_i. Here we employ a Gaussian kernel,

$$k(X_i, X_j) = exp(-||X_i - X_j||^2/b)$$

where $||X_i - X_j||$ is the Euclidean distance. This kernel, like many others, is readily interpretable as a distance or similarity measure between the two inputs it compares. Note that $k(X_i, X_j)$ will equal 1 only when $X_i = X_j$, i.e. two observations have the same covariate profile, and $k(X_i, X_j)$ approaches zero only as X_i and X_j differ on many covariates. The rate at which the distance approaches zero is dictated by the choice of b, which we discuss below. A linear combination of

vectors a.

the elements of K_i is thus a weighted sum of unit i's similarity to every other unit j in the sample, where similarity is measured by centering a Gaussian kernel over each X_j and measuring its height at X_i. Hainmueller and Hazlett (2014) provides further description and illustration of this function space. The take home point is that functions linear in $\mathbf{K}$ form a flexible, powerful space of functions. Thus, in a wide variety of cases, the outcome is more likely, or much more nearly, linear in $\mathbf{K}$ than in $\mathbf{X}$. Accordingly, the linear ignorability assumption is more credible, and the potential violations smaller.

3.2 The infeasible *kpop* estimator

Replacing $\phi(X_i)$ with K_i, we now seek to satisfy Condition 1 by choosing weights that achieve,

$$\sum_{i:R=1} w_i K_i = \frac{1}{N_{pop}} \sum_{j=1}^{N_{pop}} K_j,\ \ s.t. \sum_i w_i = 1,\ w_i \geq 0,\ \forall i \tag{7}$$

Note that every K_i here is a transformation of X_i that compares unit i to each of the units in the survey sample. The matrix $\mathbf{K}$ has a row for every unit in the sample and in the target population, yielding dimensions $N_s + N_{pop}$ by N_s.[14] The term on the right gives an (unweighted) average row of K_j for units in the target population. Note that each K_j is an N_s-vector, with the i^{th} element indicating how similar unit j in the target population is to unit i in the survey sample, i.e. $k(X_j, X_i)$. The term on the left is a weighted average of K_i over the survey sample. Here too each K_i is an N_s-vector, with the l^{th} element indicating how similar unit i in the survey sample is to unit l in the survey sample, i.e. $k(X_i, X_l)$.

In cases where (known) weights $w^{(pop)}$ are used to adjust the target population itself—as in our application below—then *kpop* would instead seek weights that bring the weighted means of K_i among the sampled units to approximately equal the $w^{(pop)}$-weighted means of K_i in the target

[14] In this present setting we rely only on the observation in the sample to formulate the columns of a kernel matrix. This is because (i) if there are millions of observations in the target population, constructing a matrix with that many rows would be infeasible, and (ii) the representation of each unit based on its similarly to other units in the sample is most relevant to how we reweight members of the sample; if there were members of the population that are very different from members of the sample, then no weighting of the sample will account for this. By contrast, the original kernel balancing formulation (Hazlett, 2020) is designed for causal inference setting in which the target and reweighted groups were the treated and control (respectively). Since these groups are often of similar and more manageable sizes, in those cases it is also possible to include the observations from the target groups when formulating the columns of $\mathbf{K}$.

population. Thus the weighting condition in 7 becomes

$$\sum_{i:R=1} w_i K_i = \sum_{j=1}^{N_{pop}} w_j^{(pop)} K_j,\ s.t. \sum_i w_i = 1,\ w_i \geq 0,\ \forall i \tag{8}$$

Calibrating through this kernel transformation achieves balance on a wide range of non-linear functions of X, without requiring the researcher to pre-specify them. For example, as we will show in Section 4, *kpop* achieves balance on the interaction of education level, region, and race in a 2016 U.S. Presidential survey without requiring the researcher to have foreknowledge of its importance, much less requiring specific knowledge that Midwestern, white voters with lower levels of educational attainment must be accounted for in the survey weights to yield accurate national predictions.

A further intuition for this approach in a survey weighting framework follows from this logic of similarity: to replace X_i with K_i (and thus $\mathbf{X}$ with $\mathbf{K}$) is to describe each observation not with its covariate coordinates but with a measure of its similarity to a number of "landmarks" in the covariate space, one for each observation in the sample. Choosing weights that produce equal means of K_i for the sample and target population thus implies that the average distance of observations to sample member j is made the same in the sample and target population, for all j. For a review of this approach, see Hazlett (2020), which also discusses how approximate balance on a kernel transformation approximately equates the multivariate distribution of X in the two groups, *as it would be estimated by a corresponding kernel density estimator*. We also note closely related work on kernel-based balancing and imbalance metrics including Wong and Chan (2018); Yeying et al. (2018); Kallus (2020). Tarr and Imai (2021) consider a related approach implemented by interpreting the Lagrange coefficients estimated in a support vector machine with such a kernel as weights.

3.3 The feasible *kpop* estimator: approximate balance

Weights that achieve equal means on every element of K_i, i.e. every column of $\mathbf{K}$, are often infeasible. Even where this can be achieved within numerical tolerance, such calibration could lead to extreme weights. Instead, we use approximate calibration weights designed to minimize the worst-case bias due to remaining miscalibration. While numerous approximation approaches are possible, we use a spectral approximation. Specifically, $\mathbf{K}$ has singular value decomposition $\mathbf{K} = \mathbf{VAU}^\top$, where the columns of $\mathbf{V}$ are left-singular vectors. Even granting that the linear ignorability assumption holds, approximate balance means the calibration step is not complete, which can introduce *additional* approximation bias. The worst-case bound on this approximation bias is given by Hazlett (2020)

$$\sqrt{\gamma}||(w^{(pop)\top}\mathbf{V}_{pop} - w_s^\top \mathbf{V}_s)\mathbf{A}^{1/2}||_2 \tag{9}$$

where $\mathbf{V}_{pop}$ is the matrix containing the rows of $\mathbf{V}$ corresponding to target population units, $\mathbf{V}_s$ contains the rows of $\mathbf{V}$ corresponding to sampled units, and $\mathbf{A}$ is the diagonal matrix of singular values. In this bias bound, $w^{(pop)}$ denotes the (optional) known weights for adjusting the target population itself. The scalar γ is the (reproducing kernel Hilbert) norm on the function, equal to $c^\top \mathbf{K} c$ effectively describing how complicated or "wiggly" the chosen function is. This is an unknown constant that need not be estimated during the optimization we describe below.

We make three remarks on the form of this worst case approximation bias (9). First, the L_2 norm of the regression function ($\sqrt{\gamma}$) controls the overall scale of potential bias. Second, the imbalance on the left singular vectors of $\mathbf{K}$ after weighting, $(w^{(pop)\top}\mathbf{V}_{pop} - w_s^\top \mathbf{V}_s)$, enters directly. Third, the impact of imbalance on each singular vector is scaled by the square root of the corresponding singular value.

The third point in particular suggests the approximate balancing approach we use: calibrate to obtain nearly exact balance on the first r singular vectors (columns of $\mathbf{V}$), leaving the remaining ($r + 1$ to N_s) columns uncalibrated. The choice of r is then chosen to minimize the bias bound (Equation 9).[15] In practice, the singular values of a typical matrix $\mathbf{K}$ decrease very rapidly (see Appendix B for an illustration from the application below). Thus, balance on relatively few singular vectors achieves much of the goal, though the procedure continues beyond this to minimize the worst-case bias bound in Equation 9 directly.

[15] Note that if calibration on the first r singular vectors was numerically perfect, then the bias bound in (9) would be equal to a modified version that omits the first r columns from $\mathbf{V}$, or sets the corresponding elements of $\mathbf{A}$ to zero. However, even if we demand calibration on the first r columns of $\mathbf{V}$, small mean imbalances may exist within the tolerance set by the software employed. Further, as noted belwo we may also calibrate on the first r columns of $\mathbf{V}$ without requiring the algorithm to converge (according to some user-chosen tolerance), leaving potentially larger imbalances. In either case these imbalances can add to the approximation bias bound, and so we retain these columns of $\mathbf{V}$ and $\mathbf{A}$ in the bias expression.

Characteristics ($\mathbf{X}$)		Kernel matrix ($\mathbf{K}$)					Outcome
female	college	k(,1)	k(,2)	k(,3)	k(,4)	(repeats)	Pr(support)
1	1	1	.14	.02	.14	...	0.80
1	0	.14	1	.14	.02	...	0.20
0	0	.02	.14	1	.14	...	0.20
0	1	.14	.02	.14	1	...	0.20
⋮	⋮	⋮	⋮	⋮	⋮	⋮	
Target Population Average		0.33	0.33	0.33	0.33		0.35
Sample Average		0.42	0.23	0.23	0.42		0.425

Table 2: Kernel matrix representing each of four unique types of individuals in the sample. Each element $k(X_i, X_j)$ is equal to $exp(-||X_i - X_j||^2/1)$, where the numerator in the exponent will be equal to two times the number of features on which i and j differ. The columns provide new bases for representing the data. Because the sample over-represents both female and non-female college degree holders, the columns of $\mathbf{K}$ representing "similarity" to each of these ($k(,1)$ and $k(,4)$ has too high a mean among the sample. *kpop* chooses weights to make the sample means of each of these columns the same.

3.4 Example with *kpop* weights

We return to our simple example of a problematic Z in section 2.6 to demonstrate *kpop*. Table 2 presents the $\mathbf{K}$ for our example. Note that the entire sample could be represented (without weights) by eight unique observations: three in the female×college stratum, three in the non-female×college stratum, and one in each of the other two. We first encode categorical data with one-hot encoding resulting in four total covariates (see Section 3.5 for details and explanation). For any two units i and j with the same values on the covariates, $k(X_i, X_j) = 1$. Choosing $b = 1$ for illustrative purposes, individuals that differ on one trait but not the other will have $k(X_i, X_j) = e^{-\left((1-0)^2+(0-1)^2+(0-0)^2+(0-0)^2\right)} = e^{-2} \approx 0.14$. Individuals who differ on both characteristics will have $k(X_i, X_j) = e^{-\left((1-0)^2+(0-1)^2+(1-0)^2+(0-1)^2\right)} = e^{-4} \approx 0.02$. Table 2 shows one corner of the resulting kernel matrix.

Since there are equal numbers of individuals of each type in the target population, the average observation is equally "similar to" individuals of all types. By contrast, the over-representation of the first and fourth categories in the sample leads the average observation to be "too similar to" observations of the over-represented types and insufficiently similar to the under-represented types. Providing *kpop* with only the matrix $\mathbf{X}$ for the sample and target population, it will construct such a kernel matrix and find weights on the sample units to make the average values of K_i equal those in the target population. This produces a weight of 2/3 for each of these units in the female×college and non-female×college category, and a weight of 2 for each unit in the other two categories, identical to the post-stratification weights. Consequently, the *kpop* weighted mean level of support (35%) matches that in the target population, while mean calibration remains biased (42.5%).

3.5 Additional details

Choice of kernel and b

Note that any calibration approach can be seen as implicitly assuming a kernel. For example, mean calibration is equivalent to balancing on the dimensions of the linear kernel, $\mathbf{K} = \mathbf{X}\mathbf{X}^\top$. The *kpop* approach offfers an explicit and reasoned choice of kernel given the requirement for linear ignorability in the corresponding $\phi(X)$.

We employ the Gaussian kernel throughout and in our software implementation. Users could reasonably choose other kernels that form function spaces more suitable to their purposes. The Gaussian kernel, however, is a useful default choice for several reasons. The choice of $\phi(\cdot)$ corresponding to the Gaussian kernel is infinite dimensional, and the Gaussian is one of several "universal" kernels, able to fit any continuous function given enough observations (see e.g. Micchelli et al., 2006). More importantly, in a given finite sample scenario, the functions that are fitted by Gaussian kernels—those that can be built by rescaling and adding Gaussian curves over all the observations in the sample—form an appealing range of smooth, flexible functions suitable for many applications.

In estimating survey weights using *kpop*, several data pre-processing decisions must be made. We provide default recommendations to reduce the number of user-driven decisions. For continuous variables, we recommend rescaling so that all variables have variance one. This avoids unit-of-measure decisions from affecting the relative influence of different variables in the similarity measure between units. In our application, however, for comparability to other approaches, we focus on categorical variables. Gaussian kernels remain appropriate with categorical variables, as demonstrated in the running example, but require different data pre-processing decisions. For categorical cases we use dummy or "one-hot encoding" of categorical variables *without dropping any levels*, and we do not rescale the resulting binary variables by their standard deviations. Under these choices, the numerator of the Gaussian kernel $||X_i - X_j||^2$ is simply two times the number of variables on which units X_i and X_j differ, regardless of the number of levels attached to each of those variables.

This brings us to the choice of b in the kernel definition. This is effectively a feature extraction choice, constrained by feasibility. Obtaining balance on a kernel matrix formed with a small b implies finer balance and less interpolation, but at the cost of more extreme weights (and eventually, infeasibility). We propose a new approach to selecting b.[16] Recall that b scales the similarly measure $||X_i - X_j||^2$. Thus, a choice of b is desirable if it does not make every observation appear "too unique" (small b) or "too similar" (large b). Further, a desirable choice of $\mathbf{K}$ produces a range of values in each column, yielding meaningful variability in the similarity/difference between different pairs of units. Given that this measure is bounded between 0 and 1, we use the variance of $\mathbf{K}$ (disregarding the diagonal, which is unchanging) as a measure of the useful information available at a given choice of b, and choose the variance-maximizing value of b. We make no claim as to the optimality of this result, but note that it does guarantee a reasonable choice which is made *ex ante* by the user. In our simulations and applications, this choice produces consistently good performance, though the results are shown to be stable across a wide range of b regardless (see appendix C.5).

Practice and diagnostics

We recommend several diagnostics that can be used to better understand the resulting weights and what they achieved or failed to achieve. First, the number of dimensions of $\mathbf{K}$ optimally selected for calibration (r) should be checked. The solution will reflect only balance on r dimensions of $\mathbf{K}$, so if this is very small (e.g. 1 or 2), the user should be aware. Next, researchers should compare the weighted sample and target population margins on the original $\mathbf{X}$ and explicitly chosen functions of these variables such as interactions (as we demonstrate in our application and simulation). Third, we suggest two summary statistics to assess the degree to which multivariate balance has been improved. The first is an L_1 measure of the distance between the distribution of X for the survey and the population, summed over the units of the survey. This can be obtained both before and

[16] In related kernel methods, a common default choice for b is often $b = D$ or $b = 2D$ where D is the number of dimensions of $\mathbf{X}$ used to form $\mathbf{K}$. One reason is that, for standardized data, the expectation of the numerator in the exponent grows proportionally to D (Hainmueller and Hazlett, 2014). Hence scaling proportionally to D is reasonable, keeping the magnitude of the exponent stable as D grows. However the best constant of proportionality remains unclear.

after weights are applied to assess the reduction in multivariate imbalance (Hazlett, 2020). The second is the ratio of the approximation bias bound (Equation 9), calculated with and without the weights, to determine the proportional improvement in the degree of potential bias due to remaining imbalances on $\mathbf{K}$. Both serve to indicate to the user whether substantial improvements in multivariate balance were achieved by the weights.

Finally, it is often valuable to understand how extreme the weights are and thus how heavily the solution depends on a small number of observations. This can be done by the investigator's preferred means, such as inspecting the distribution of weights visually, or constructing statistics such as the effective sample size or the number of observations (working from the most heavily weighted towards the least) that one needs to sum to achieve 90% of the total sum of weights.

Prioritizing mean calibration on X

Researchers and pollsters may reasonably hope for exact (mean) calibration on variables of known importance to the outcome of interest, even if the means are no more important to balance than unseen higher-order moments. Further, it may be useful to know that a given estimator achieves balance on the same moments as conventional raking or mean calibration, in addition to possibly calibrating higher-order moments. To this end, we consider a "*kpop* + mean first" (*kpop+MF*) procedure, in which the weights are constrained to obtain exactly equal means (within a set tolerance) on a chosen set of variables X, in addition to calibrating on r singular vectors of $\mathbf{K}$ chosen so as to minimize the bias bound described above. *kpop+MF* eases comparisons with commonly-used calibration approaches while also providing the additional benefit of reducing bias in cases where response probability and outcomes are non-linear in X. The drawback is that *kpop+MF* will typically be able to feasibly calibrate fewer dimensions of $\mathbf{K}$ than could be calibrated without such constraints.

4 Application: 2016 U.S. Presidential Election

In the 2016 United States Presidential election, state-level polls in key states were severely biased, with polling aggregators making over-confident predictions that Donald Trump would lose. National polls correctly predicted that Hillary Clinton would lead the national popular vote, while many overstated the margin. The challenges of correctly weighting a highly non-random sample to match the national electorate likely contributed to these errors. As Kennedy et al. (2018) note, existing polls were especially likely to over-represent college-educated whites. We test whether weighting with *kpop* aids in constructing more effective weights, without foreknowledge of what functions of covariates/ intersectional strata are essential.

Because voters may have changed their mind between a given pre-election survey and the day of their vote, simply checking whether weighting the outcome of a pre-election survey produces an estimate close to the true election result would not provide a meaningful test of weighting techniques. Indeed, Kennedy et al. (2018) argue that such changes contributed to polling failures, with undecided voters breaking towards Trump late in the election cycle. To avoid this complication and focus our application solely on the question of whether each weighting approach produces representative estimates, we do not try to estimate election results using the pre-election stated vote choice, but rather turn to estimating what the average "retrospective vote choice", measured post-election, would have been in the target population, namely voters in the 2016 election. This involves (1) training a model that predicts stated retrospective vote choice as a function of X using a large post-election survey which we define as the target population; (2) applying this model to predict the "retrospective vote choice" of each individual in a pre-election survey using their covariates X; (3) constructing weights to calibrate the pre-election sample to the target population; then (4) comparing the weighted average of the predicted "retrospective vote choice" in the pre-election sample to the stated "retrospective vote choice" the target population.

4.1 Data and details

Survey sample. For the respondent sample, we use survey data from the final poll conducted by the Pew Research Center before the general election in November 2016.[17] We keep only the $N_s = 2,052$ respondents who report that they plan to vote, or have already voted, and we dispense with the nonresponse weights provided by the survey. The publicly available data do not include survey design weights, and we use $q_i = 1$ for all respondents. Our outcome of interest is candidate

[17] Pew is a high-quality, non-partisan public opinion firm. The survey was conducted from October 20-25, 2016 using telephone interviews among a national sample of 2,583 of adults. On landline phone numbers, the interviewer asked for the youngest adult currently home (647), and cell phone interviews (1,936) were done with the adult who answered the phone (Pew Research Center, 2016). Random-digit dialing was used, combined with a self-reported voter registration screen.

choice in the Presidential election; we code vote choice as being for "Republican Donald Trump", "Democrat Hillary Clinton", or "Other/Don't Know", and we include voters who "lean" towards one of the two major party candidates.

Defining the target population. Ideally we would define the target population using verified voter records from the Secretaries of State. However, we do not have access to such an administrative file. Instead, following Caughey et al. (2020), we define our target population using the common content from the post-election wave of the 2016 Congressional Cooperative Election Study (CCES) (Ansolabehere and Schaffner, 2017). The advantage of the CCES data is two fold: the CCES is a large survey that aims to be representative of all voters, and the survey weights for the post-election wave lead to an estimate of the popular vote margin between the two major parties (2.48 percentage points, D - R) that is very close to the truth (2.3 percentage points). Second, the CCES includes a number of demographic survey questions that overlap with those asked in the Pew study, which we can use for calibration. We incorporate the weights provided by the CCES ("commonweight_vv_post") into the definition of our target population, and limit to voters who stated that they "definitely voted" and for whom the outcome variable was not coded as missing, leaving $N_{pop} = 44,932$ units in our target population.

Our auxiliary data, X, are defined using all of the overlapping variables in our data sets: age, reported gender, race/ethnicity, geographic region, education level, party identification, income (5 buckets), born-again Christian, church attendance, and religion. All variables are self-reported except for region. Appendix Table C.1 summarizes the distributions of these variables and how they differ in the target population (CCES) compared to the survey sample (Pew) at base. For example, those with higher levels of education and higher income are over-represented in the Pew sample, as are older voters and Independents. By contrast Black voters and women are under-represented in the sample relative to the target population.

Modeled outcome. We use a regularized multinomial logit model to estimate the relationship between X and three-way "retrospective vote choice" (Republican, Democrat, and Other) measured by asking respondents who they voted for in the post-election CCES survey. Specifically, we include gender, 3-way party identification, race/ethnicity, 6-way education, region, 6-way income, 5-way religion, 4-way church attendance, born-again status, continuous age, age^2, gender $\times$ party identification, and age (cont.) $\times$ party identification. The degree of regularization was determined by ten fold cross-validation. We choose this set of variables such that the outcome we construct is known to include interactions. None of the subsequent weighting methods are "told" this particular choice of $\phi(X)$ had been made. Note also that the outcome can be modeled quite effectively: choosing the highest-probability outcome leads to an 85-86% correct classification rate for non-independents. This fitted post-election outcome model is then applied to the X data from every unit in the Pew pre-election survey to create their (predicted) "retrospective vote choice". Our target is the (weighted) Democratic vote margin (D - R), here 2.48 percentage points, based on reported retrospective vote choice in the CCES, using the CCES-provided weights. Additional details can be found in Appendix C.2.

Weighting methods. We compare our method to two common methods researchers use for constructing survey weights as discussed above, mean calibration and post-stratification. For mean calibration, we consider four specifications representing the types of choices researchers make in specifying what variables to include in weighting. In the first model (*mean calibration, demos*), we create survey weights that match the marginal distribution for the means for basic demographic variables including: age (4way), gender, race/ethnicity, geographic region, and party identification. Our second model includes education as well (*mean calibration, with education*). Our third model (*mean calibration, all*) includes all variables described in Table C.1. For post-stratification, we similarly consider age, gender, race/ethnicity, region, party identification, income, born-again Christian status, education, and an additional interaction of education $\times$ region $\times$ race/ethnicity for white respondents, but only region $\times$ race/ethnicity among non-white voters (*post-stratification*). Because empty cells are a concern when we interact these 8 auxiliary variables, we coarsen age, income, and education in to 3-, 4-, and 4-category variables respectively and do not include religion or church attendance.[18] Even with the coarsened age, income, and education level variables, 60% of strata present in the CCES population are empty in the Pew sample. Given this significant hurdle, we additionally provide the post-stratification estimator with an important, retrospectively-informed interaction among region, race, and education in order to give it the best advantage possible.

[18]For age, we collapse 18 to 50 year old respondents into one strata. For income, we collapse to the bins $< 50k$, $50 - 150k$, $> 150k$, and "prefer not to say." For education, among white voters, we collapse to those without a college degree, those with a college degree, and those with a post-graduate degree, and we do not stratify on education for non-white voters. Without this coarsening, a full 86% of the strata in CCES accounting for 30559 / 44932 units are missing from the Pew sample.

Our final model employs mean calibration but leverages retrospective, substantively informed expertise, based on the analysis of Kennedy et al. (2018) (*mean calibration, retrospective*). To address the importance of low-education white voters in the 2016 election, particularly in Midwestern states, we include the same interaction provided to post-stratification additionally interacted with party identification, namely, covariates for education level × region × party identification × race/ethnicity for white respondents, but only account for region × party identification × race/ethnicity among non-white voters. Additionally, we include mean calibration on age and gender interacted with party identification. This model "cheats", in that it is informed by knowledge gained after the election cycle. We include it only to evaluate the question of whether our proposed *kpop* method, when not given such knowledge, can perform as well as this retrospective-informed model that serves as a best-case for what expert knowledge could hope to achieve.

We compare the models described above with *kpop* (*kpop*) using all the categorical variables described in Table C.1 . We also include three models that first conduct mean calibration before proceeding to balance on the kernel matrix as discussed in 3.5, *kpop + MF (demos), kpop + MF (with education)*, and *kpop + MF (all)*, which practitioners may prefer these approaches as they produce the expected balance on the most visible margins. The other advantage of including these mean-first approaches here is that they are easily compared to the mean calibration approaches, which start with the same moment constraints but do not include the additional kernel-based constraints. For all *kpop* estimates, we show results using the kernel bandwidth, b, that produces the maximum variance in the kernel matrix. Details of data encoding follow the prescriptions in Section 3.5.

4.2 Results

Balance. We first consider the balance achieved by each method on the observed covariates. Table 3 presents the absolute error, weighted by the target population proportion for each level, for each auxiliary variable (rows 1-9) and a set of interactions (10-16). By construction, the mean calibration methods, as well as *kpop + MF* methods, perfectly match the marginal distributions for any variables that are included in the model.

All methods greatly improve representativeness of the respondent sample as indicated by the reduction in error across variables and interactions relative to the unweighted sample. As expected, *kpop* (without "mean first") achieves good but imperfect balance on the included covariates and interactions, despite not being directly constrained to achieve balance on them. We would also expect post-stratification to produce perfect balance on the included terms, however, even with coarsening, empty cells pose a significant problem, leading to a failure to get the correct margins, much less produce the non-parametric adjustment one hopes for under Assumption 1.

Table 3: Weighted Mean Absolute Error on Auxiliary Variables (percentage points)

Variable	Pew Orig	kpop	kpop + MF: (Demos)	Mean Calib: (Demos)	kpop + MF: (D+Edu)	Mean Calib: (D+Edu)	kpop + MF: (All)	Mean Calib: (All)	Mean Calib: Retro	PS Reduc
female	3.65	0.11	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.06
pid (3way)	2.53	0.28	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.45
age (4way)	4.85	0.22	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.95
race/ethnicity	1.54	0.16	0.00	0.00	0.00	0.00	0.00	0.00	1.25	3.50
region	1.50	0.02	0.00	0.00	0.00	0.00	0.00	0.00	0.58	2.51
educ (6way)	8.64	0.32	0.09	8.68	0.00	0.00	0.00	0.00	1.67	2.09
income (6way)	4.35	0.35	0.24	4.04	0.15	3.35	0.00	0.00	3.12	1.78
born	1.72	0.14	0.07	2.95	0.10	0.18	0.00	0.00	0.62	4.95
relig (5way)	6.42	0.48	0.26	5.24	0.19	6.66	0.00	0.00	7.77	6.17
pid×race	0.71	0.50	0.26	0.86	0.15	0.68	0.63	1.15	0.42	1.20
educ×pid	6.22	0.36	0.30	6.25	0.29	0.45	0.38	0.57	1.56	1.49
educ×pid×race	3.81	0.45	0.58	3.87	0.34	0.44	0.67	0.58	0.46	0.90
race×educ×region	2.91	0.24	0.37	2.97	0.27	0.48	0.70	0.49	0.32	0.87
educ×white	10.80	0.28	0.64	10.64	0.21	0.25	0.09	0.42	1.00	2.69
midwest×white×edu	1.54	0.44	0.24	0.96	0.38	0.47	0.35	0.27	0.35	0.56
midwest×educ×race	1.68	0.10	0.08	0.69	0.04	0.06	0.04	0.05	0.88	1.01

Note: Absolute error in the distribution of categorical variables, weighted by the target population proportion for each level. Gray numbers indicate the variable was included as a calibration constraint, and so imbalances very near zero are expected. Note that all interactions with education us a three-way education coding.

In the lower rows of Table 3 we investigate the balance on important interactions, including an interaction that Kennedy et al. (2018) deemed important: midwest × education level × race (bottom row). Without explicitly incorporating knowledge about the importance of these variables, *kpop* significantly improves the balance of this interaction, reducing the mean absolute error from the initial value of 1.68 down to 0.1 percentage point, a much greater reduction than any of the non-*kpop* estimators. When incorporating the mean first requirements *kpop + MF* also effectively addresses this interaction, reducing absolute bias to between 0.04 to 0.08 percentage points. We see similar patterns of improvements in performance of the *kpop* methods across a number of important interactions. Notably, the *kpop + MF* methods outperform their mean calibration counterparts, emphasizing how *kpop* is more robust to user-specified constraints.

Weights and diagnostics. Across all *kpop* methods, the distribution of weights are markedly similar, while mean calibration methods produce more variable distributions (see Appendix C.4). Compared to both of these sets of approaches, post-stratification produced weights that were skewed toward larger, more extreme values.

The additional constraints solved by *kpop* weights can lead to reduced effective sample sizes, however. For example, *post-stratification* and *mean calibration, all* have effective sample sizes of 983 and 1101, respectively, while *kpop* and *kpop + MF (all)* have effective sample sizes of 789 and 750 respectively. Similarly, the (minimum) number of observations required to arrive at 90% of the total weight is 1235 and 1362 for *post-stratification* and *mean calibration, all* respectively, but 1193 and 1223 for *kpop* and *kpop + MF (all)*. We consider this a fairly modest price to pay in variance if it increases our confidence in low bias.

Weighting diagnostics. Finally, beyond investigating the quality of balance obtained on selected margins, *kpop*ś ability to improve balance on $\mathbf{K}$ can be assessed using the diagnostics described above. The L_1 gap between the kernel-based estimates of multivariate density fell from 0.0318 prior to weighting to 0.0011 or below under all *kpop* estimates, roughly a 29-fold improvement. Similarly the bias bound showed 5-6 fold improvements under each set of weights as compared to the unweighted bias bound. Additional diagnostic and descriptive results can be found in Appendix C.4.

Estimates. Recall that our target is a two-way vote margin on reported "retrospective vote choice" (D-R) of 2.48 percentage points and that, for each weighting model, we evaluate the average of the predicted "retrospective vote choice". The Pew survey, without weights, is extremely non-representative, with an unweighted average Clinton two-party vote margin of -0.194 [-3.24, 2.85] percentage points (95% confidence interval in brackets). Mean calibration on basic demographics, excluding education level, exacerbates the difference but flips the signs of the estimate, producing an estimated margin to 6.49 [3.25, 9.74] percentage points. Mean calibration including education performs well, with an estimate of 3.04 [-0.610, 6.69]. This is consistent with the findings of Kennedy et al. (2018) that education level was a significant driver of both non-response and voting for Donald Trump. Moving to mean calibration on all auxiliary variables, the estimate moves farther from the truth to 4.31 [0.16, 8.47]. Finally, mean calibration on the retrospectively informed choice of variables and the potentially important interaction of region and education among white voters generated an improved point estimate of 3.30 [-0.48, 7.07] substantially closer to the truth. The *kpop* estimates are both stable and close to the truth across different specifications. Using *kpop* alone results in a weighted estimate of 2.50 [-2.37, 7.37] percentage points, the closest to the truth of all nine methods tested. $kpop + MF$ with additional mean first calibration produces point estimates of 2.88 [-2.49, 8.24] (*demos*), 3.09 [-2.83, 9.01] *(with education)*, and 3.19 [-1.81, 8.19] (*all*) percentage points, all close to the target margin in the CCES. Table C.3 summarizes these results across all weighting methods.

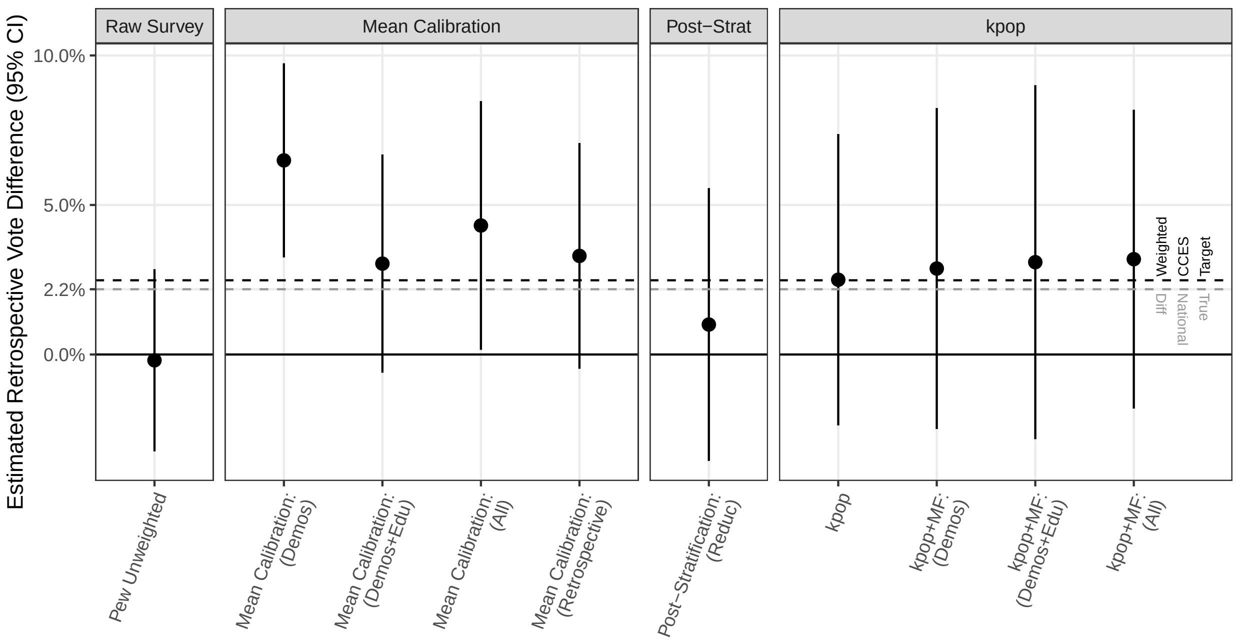

Figure 1: Comparison of weighting methods on Pew survey data to weighted CCES Target. Points represent the estimated vote margin from the predicted "retrospective vote choice" using Pew survey data and corresponding weighting scheme. The dashed black line indicates the target, the reported two-way vote margin in the CCES. The dashed grey line indicates the true values from national vote returns.

The standard errors used here assume that the sampling weights correspond to the survey weights, which is known to be conservative (Lumley, 2011). As is common with survey weighting, the weighted estimates have larger standard errors than the unweighted estimate (Little and Vartivarian, 2005), and we see that even with the mean calibration methods, balancing on more covariates inflates the variance. *kpop* balances on higher order interactions, reducing bias at a cost to variance. For example the standard errors for *kpop* are about 29% larger than the retrospective mean calibration weights. However, the smaller standard errors of the mean calibration models should not increase a reader's confidence in the results without further justification of the implied linear ignorability assumption.

Simulation. While *kpop* performs well in terms of achieving balance and low bias estimates in this application, we turn to a simulation setting in Appendix D to more fully investigate performance on both bias and variability. To construct a realistic simulation, we employ the same setting above, but draw samples from the (unweighted) CCES using a model based probability partially learned from the data. Details can be found in AppendixD and E . We find that *kpop* outperforms standard mean calibration models in terms of bias, and that unlike our example, it does not always come at a cost of increased variance relative to simple mean calibration methods. Individual *kpop* estimates tend to be closer to the truth and thus also more stable.

5 Discussion

We note several limitations and areas for future work. First, the implementation described here makes no use of outcome data when constructing weights. This allows users to choose weights blind to the outcome to protect themselves against data "snooping". Further, these weights would be appropriate for estimating any outcome, as several are often of interest in a given survey. The downside, however, is that this leaves possible gains in efficiency and bias reduction on the table if there are functions of X that predict response (and are thus imbalanced) but that do not predict the outcome and so need not be calibrated to achieve linear ignorability. Still, recognizing such variables and choosing not to calibrate on them could lead to improved calibration on the remaining variables, possibly resulting in less extreme weights. Such an approach remains an option worth exploring in future work.

While we suggest the use of the Gaussian kernel with a variance maximizing choice of b, optimal selection of b remains an area of ongoing research. Fortunately, our empirical results are not sensitive to the choice of b (see Appendix C.5), but this may not always be the case. Further, while the present example focused on discrete X for comparability to other approaches, a benefit of our approach, and the Guassian kernel, is that it applies well—and perhaps more naturally—with continuous X. Nevertheless other choices of kernels, and a means of choosing among them, is a fruitful areas of future work.

Next, in our implementation, we use only the sample observations to form the columns of **K**, a decision driven by feasibility constraints. Since the number of units in the target population is typically very large, using all sample and population units to form the columns of **K** is typically infeasible. Future work could consider ways of augmenting the columns of **K** by selecting population units that are poorly represented by sample units and using these additionally in the formation of the bases for calibration. Note that strictly non-parametric approaches such as post-stratification require, correspondingly, strict positivity, meaning that such "missing types" would have to be dropped entirely from the target population, changing the estimand. In approaches that instead imply an underlying function space, such as mean calibration methods, balance on the selected bases allows for interpolation or extrapolation across these gaps (depending on where the units fall). That said, including new columns that represent population units unlike any sample unit as additional bases will make feasibility of the weights more difficult to achieve, especially when these "missing types" lie far from the sample and require extrapolation. Attempting to calibrate these cases may prove infeasible, yield unsatisfactory bias bound ratios, or produce extreme weights. Such stumbling blocks are useful warning signs that the user may be demanding more extrapolation than is desirable, requiring careful reconsideration the target population and its suitability given the sample at hand.

In summary, *kpop* is a kernel-based approach for weighting samples to be representative of target populations, while reducing reliance on user discretion and domain expertise to determine what covariates—and functions of those covariates—should be used for calibration. It does so by estimating a flexible, non-linear set of basis functions through a kernel transformation and achieving approximate balance on this representation of the covariates. As shown in our application to the 2016 U.S. Presidential election, this method stands to greatly reduce bias in non-representative samples.

References

Ansolabehere, S. and Schaffner, B. F. (2017), 'CCES Common Content, 2016'.
URL: *https://doi.org/10.7910/DVN/GDF6Z0*

Ben-Michael, E., Feller, A. and Hartman, E. (2021), 'Multilevel calibration weighting for survey data'.

Berinsky, A. J. (2006), 'American public opinion in the 1930s and 1940s: The analysis of quota-controlled sample survey data', *International Journal of Public Opinion Quarterly* **70**(4), 499–529.

Caughey, D., Berinskey, A. J., Chatfield, S., Hartman, E., Schickler, E. and Sekhon, J. S. (2020), *Target Estimation and Adjustment Weighting for Survey Nonresponse and Sampling Bias*, Elements in Quantitative and Computational Methods for the Social Sciences, Cambridge University Press.

Caughey, D. and Hartman, E. (2017), 'Target selection as variable selection: Using the lasso to select auxiliary vectors for the construction of survey weights', *Available at SSRN 3494436* .

Chen, J. K. T., Valliant, R. L. and Elliott, M. R. (2019), 'Calibrating non-probability surveys to estimated control totals using lasso, with an application to political polling', *Journal of the Royal Statistical Society: Series C (Applied Statistics)* **68**(3), 657–681.

Deville, J.-C. and Särndal, C.-E. (1992), 'Calibration Estimators in Survey Sampling', *Journal of the American Statistical Association* **87**(418), 376–382.

Hainmueller, J. (2012), 'Entropy balancing for causal effects: A multivariate reweighting method to produce balanced samples in observational studies', *Political analysis* pp. 25–46.

Hainmueller, J. and Hazlett, C. (2014), 'Kernel regularized least squares: Reducing misspecification bias with a flexible and interpretable machine learning approach', *Political Analysis* **22**(2), 143–168.

Hartman, E. and Levin, I. (2019), Accounting for complex survey designs: Strategies for post-stratification and weighting of internet surveys, *in* E. Suhay, B. Grofman and A. H. Trechsel, eds, 'Oxford Handbook of Electoral Persuasion', Oxford University Press.

Hazlett, C. (2020), 'Kernel balancing: A flexible non-parametric weighting procedure for estimating causal effects', *Statistica Sinica* **30**, 1155–1189.

Kallus, N. (2020), 'Generalized optimal matching methods for causal inference.', *Journal of Machine Learning Research* **21**(62), 1–54.

Kalton, G. and Flores-Cervantes, I. (2003), 'Weighting Methods', *Journal of Official Statistics* **19**(2), 81–97.

Kennedy, C., Blumenthal, M., Clement, S., Clinton, J. D., Durand, C., Franklin, C., McGeeney, K., Miringoff, L., Olson, K., Rivers, D. et al. (2018), 'An evaluation of the 2016 election polls in the united states', *Public Opinion Quarterly* **82**(1), 1–33.

Kennedy, C. and Hartig, H. (2019), "'Response rates in telephone surveys have resumed their decline"'.
URL: *https://www.pewresearch.org/fact-tank/2019/02/27/response-rates-in-telephone-surveys-have-resumed-their-decline/*

Kott, P. S. and Chang, T. (2010), 'Using Calibration Weighting to Adjust for Nonignorable Unit Nonresponse', *Journal of the American Statistical Association* .

Linzer, D. A. (2011), 'Reliable inference in highly stratified contingency tables: Using latent class models as density estimators', *Political Analysis* pp. 173–187.

Little, R. J. and Rubin, D. B. (2019), *Statistical analysis with missing data*, Vol. 793, John Wiley & Sons.

Little, R. J. and Vartivarian, S. (2005), 'Does weighting for nonresponse increase the variance of survey means?', *Survey Methodology* .

Lumley, T. (2011), *Complex surveys: a guide to analysis using R*, Vol. 565, John Wiley & Sons.

McConville, K. S., Breidt, F. J., Lee, T. C. and Moisen, G. G. (2017), 'Model-assisted survey regression estimation with the lasso', *Journal of Survey Statistics and Methodology* **5**(2), 131–158.

Mercer, A. W., Kreuter, F., Keeter, S. and Stuart, E. A. (2017), 'Theory and Practice in Nonprobability Surveys', *Public Opinion Quarterly* **81**(S1), 250–271.

Micchelli, C. A., Xu, Y. and Zhang, H. (2006), 'Universal kernels', *Journal of Machine Learning Research* **7**(Dec), 2651–2667.

Pew Research Center (2016), As Election Nears, Voters Divided Over Democracy and 'Respect', Technical report, Pew Research Center.

Pew Research Center (2018), For weighting online opt-in Samples, what matters most?, Technical report, Pew Research Center.

Särndal, C.-E. (2007), 'The calibration approach in survey theory and practice', *Survey Methodology* **33**(2), 99–119.

Särndal, C.-E. and Lundström, S. (2005), *Estimation in surveys with nonresponse*, John Wiley & Sons.

Soriano, D., Ben-Michael, E., Bickel, P. J., Feller, A. and Pimentel, S. D. (2021), 'Interpretable sensitivity analysis for balancing weights'.

Tarr, A. and Imai, K. (2021), 'Estimating average treatment effects with support vector machines', *arXiv preprint arXiv:2102.11926* .

Wong, R. K. and Chan, K. C. G. (2018), 'Kernel-based covariate functional balancing for observational studies', *Biometrika* **105**(1), 199–213.

Wu, C. and Lu, W. W. (2016), 'Calibration Weighting Methods for Complex Surveys', *International Statistical Review* **84**(1), 79–98.

Yeying, Z., Debashis, G. et al. (2018), 'A kernel-based metric for balance assessment', *Journal of Causal Inference* **6**(2).

Zhao, Q. and Percival, D. (2016), 'Entropy balancing is doubly robust', *Journal of Causal Inference* **5**(1).

Supplementary Materials for "*kpop*: A kernel balancing approach for reducing specification assumptions in survey weighting"

A Proofs

To first review notation, we consider covariates X_i, the outcome of interest, Y_i, and an indicator for having completed data on Y_i, $R_i \in \{0,1\}$, accounting for both unit i being in a sample and completing the question. In addition, we consider an *unobserved* factor Z_i. Independent observations of the form $\{X_i, Z_i, Y_i, R_i\}$ are realized from a common joint density $p(X_i, Z_i, Y_i, R_i)$. Investigators also have a smaller survey respondent sample, consisting only of N_s units. We consider first the case where these N_s sampled units are drawn, with bias, from the N_{pop} units in the target population data, with $R_i = 1$ indicating sampled units, who thus have non-missing values of Y_i. Below we also consider the case when the sample is drawn disjointly from the target population.

A.1 Outcome linearity

While the full linear ignorability assumption defined in the paper regards the independence of (link) linear residuals from both the outcome and response models, to develop the underlying ideas we will first consider only the residual in the outcome model and how it relates to the full response process, not just the residual in the response process.

The decomposition of Y into $\phi(X)^\top\beta + \epsilon$ is always possible, and so the assumptions embodied by this decomposition are borne out by further assumptions on ϵ. Here, unlike in many regression settings, the concern is not the relationship between $\phi(X)$ and an ϵ defined as other causes of Y, but rather between the ϵ that is constructed simply to be orthogonal to X, and the sample-presence indicator, R.

While non-parametric ignorability asserts that $Y \perp\!\!\!\perp R \mid X$, suppose we replace this with the notion that the variation in Y not explained linearly by $\phi(X)$, namely ϵ, is independent of R: $\epsilon \perp\!\!\!\perp R$. Note that (i) the non-parametric conditioning on $\mathbf{X}$ has been replaced here by a specific parametric adjustment, $Y - \phi(X)^\top\beta$, and (ii) we have not yet removed the linear relationship between $\phi(X)$ and R, and doing so is complicated by R's potential link-linear relationship to $\phi(X)$, but will consider that shortly.

Without loss of generality, consider the decomposition of Y as

$$Y = \phi(X)^\top\beta + Z + \nu$$

in which Z can take various forms that will affect whether linear ignorability holds and such that ν contains merely idiosyncratic noise independent of $\phi(X)$, Z, and R. The ϵ above is now composed of $Z + \nu$. Note we can assume Z is orthogonal to $\phi(X)$ because, if it wasn't, one could replace the original Z with the component not in the span of $\phi(X)$, $Z - (\phi(X)^\top\phi(X))^{-1}\phi(X)^\top Z$. This will change the β of the best fitting model, but that is not a concern here.

The weighted average outcome in the sample will then be,

$$\mu_s(w) = \frac{1}{N_s}\sum Y_i = \frac{1}{N_s}\sum w_i \phi(\mathbf{X})_i^\top \beta + \frac{1}{N_s}\sum w_i Z_i + \frac{1}{N_s}\sum w_i \nu_i$$

while the mean outcome in the target population is

$$\mu = \overline{\phi(X)}_{pop}^\top \beta + \frac{1}{N_{pop}}\sum Z_i + \nu_i$$

which will not equal $\mathbb{E}[Y]$ exactly, but $p(\cdot)$ is constructed such that $\mathbb{E}[\mu] = \mathbb{E}[Y]$. The bias under exact mean calibration on X is then

$$\begin{aligned}
\text{bias} &= \mathbb{E}[\mu_s(w) - \mu] \\
&= \mathbb{E}\left[\frac{1}{N_s}\sum w\phi(X_i)^\top \beta + \frac{1}{N_s}\sum w_i Z_i + \frac{1}{N_s}\sum w_i \nu_i - \left(\overline{\phi(X)}_{pop}^\top \beta + \frac{1}{N_{pop}}\sum Z_i + \frac{1}{N_{pop}}\sum \nu_i\right)\right] \\
&= \mathbb{E}\left[\overline{\phi(X)}_{pop}^\top \beta + \frac{1}{N_s}\sum w_i Z_i + \frac{1}{N_s}\sum w_i \nu_i - \left(\overline{\phi(X)}_{pop}^\top \beta + \frac{1}{N_{pop}}\sum Z + \frac{1}{N_{pop}}\sum \nu_i\right)\right] \\
&= \mathbb{E}\left[\frac{1}{N_s}\sum w_i Z_i - \left(\frac{1}{N_{pop}}\sum Z_i\right)\right] + \mathbb{E}\left[\frac{1}{N_s}\sum w_i \nu_i - \frac{1}{N_{pop}}\sum \nu_i\right] \\
&= \mathbb{E}[wZ \mid R = 1] - \mathbb{E}[Z] + 0
\end{aligned}$$

Hence if Z (and thus $\epsilon = Z + \nu$) is not orthogonal to R, the bias is simply *the difference between the weighted average of Z in the sample and the average of Z in the target population.*[19]

So far this result does not depend on whether the sample units $(i : R_i = 1)$ are a subset of the target population units. Note that the term $\left[\frac{1}{N_s}\sum_{i:R=1} w_i \nu_i - \frac{1}{N_{pop}}\sum_i \nu_i\right]$ equals zero in expectation so disappears from the bias but is one source of error in a finite sample. In the case where the sample is a subset of the target population, the second term in this expression can be rewritten as a combination of the sample units and the units in the sample not in the population, leaving us with

[19]Note that while $\mathbb{E}[\cdot]$ here integrates over p (the full data-generating process), this is an expectation of the bias as defined by the difference between a fixed target population mean (i.e. the μ we would have versus our estimate of it from a smaller weighted sample, $\mu_s(w)$.

$$= \frac{1}{N_s} \sum_{i:R=1} w_i \nu_i - \left(\frac{N_s}{N_{pop}} \right) \frac{1}{N_s} \sum_{i:R=1} w_i \nu_i + \left(\frac{N_{pop} - N_s}{N_{pop}} \right) \frac{1}{N_{pop} - N_s} \sum_{i:R_i=0} \nu_i \tag{10}$$

$$= \frac{1}{N_s} \sum_{i:R=1} w_i \nu_i - \frac{1}{N_s} \sum_{i:R=1} w_i \nu_i + \frac{1}{N_{pop}} \sum_{i:R=0} \nu_i \tag{11}$$

$$= \frac{1}{N_{pop}} \sum_{i:R=0} \nu_i \tag{12}$$

Hence, this particular source of finite sample error is smaller when the sample is a subset of the target population: because the sample and population share units, they differ on ν only for the observations that are in the target population and not the sample.

When we consider a $Z \perp\!\!\!\perp R$ then $\epsilon = Z + \nu$ is independent of R. Accordingly, when $Z \perp\!\!\!\perp R$, $\mathbb{E}[Z|R = 1] = \mathbb{E}[Z]$. Further, as the weights depend upon only R and X, $Z \perp\!\!\!\perp X$ and $Z \perp\!\!\!\perp R$ implies $Z \perp\!\!\!\perp w$. Hence, when Assumption 3 holds, the final line above becomes $\mathbb{E}[w|R = 1]\mathbb{E}[Z|R = 1] - \mathbb{E}[Z] = \mathbb{E}[Z] - \mathbb{E}[Z] = 0$, concluding our proof.

A.2 Link-linearity in the sampling process

The full linear ignorability assumption (3) is met not only when $\epsilon \perp\!\!\!\perp R$, but more weakly when $\epsilon \perp\!\!\!\perp \eta$, where η is the part of the sampling probability that cannot be explained (link) linearly by $\phi(X)$. Derivations working with η are complicated by the fact that, as a binary variable, we do not expect to model R directly as a linear function of $\phi(X)$ but rather through a link function. Suppose we impose a general link-linear decomposition, $Pr(R = 1|X) = g(\phi(X)^\top \theta + \eta)$, where $g(\cdot)$ is some (inverse) link function $g : \mathbb{R} \mapsto [0, 1]$. We now consider the assumption of independence only between the linear residual of the outcome and "link-linear" residual for the selection, are independent, i.e. that $\epsilon \perp\!\!\!\perp \eta$.

In the "nested" case used in the text, where the sample contains a subset of the units in the target population, mean calibration weights solve the moment conditions,

$$\sum_{x \in X} \phi(x) w(x) p(X = x \mid R = 1) = \sum_{x \in X} \phi(x) p(X = x) \tag{13}$$

where $p(X = x)$ is the distribution among the target population. Note that though $\phi(X)$ has an infinite-dimensional representation in the case of the Gaussian kernel, the virtue of the kernel representation is that this can be achieved by calibrating on K_i instead of $\phi(X_i)$. The number of constraints to be solved (neglecting the approximation) is thus the number of dimensions of K_i, namely N_s. Further, the same moment conditions would be solved were we to choose weights according to the function $w(x) = \frac{p(X=x)}{p(X=x|R=1)}$. We thus assume the empirical weights solving Equation 13 converge to those described by $w(x)$ and can thus be expressed as

$$w(x) = \frac{p(x)}{p(x \mid R = 1)} = \frac{p(x)p(R = 1)}{p(R = 1|x)p(x)} = \frac{p(R = 1)}{p(R = 1|x)} \tag{14}$$

producing the well-known "inverse probability of selection" weights. In particular the sample weights become

$$w_i = \frac{N_s/N_{pop}}{g(\phi(X_i)^\top\theta + \eta_i)} \tag{15}$$

The "non-nested' case, where the sample ($R = 1$) is drawn not as a subset of the target population but as a separate group, is similar. Here, mean calibration for $\phi(x)$ solves

$$\sum_{x\in X} \phi(x)w(x)p(X = x \mid R = 1) = \sum_{x\in X} \phi(x)p(X = x \mid R = 0)$$

where $R = 0$ designates units in the target population alone and $R = 1$ are those in the sample alone. Expression (14) is then replaced by

$$w(x) = \frac{p(x \mid R = 0)}{p(x \mid R = 1)} = \frac{p(R = 0|x)p(R = 1)}{p(R = 1|x)p(R = 0)} = \frac{p(R = 0|x)}{p(R = 1|x)}\frac{N_s}{N_{pop}} \tag{16}$$

so that $w(X_i)$ is just a rescaling of $\frac{1-g(\phi(X_i)^\top\theta+\eta_i)}{g(\phi(X_i)^\top\theta+\eta_i)}$, i.e. the inverse "odds of selection" rather than probability of selection. In either case, the weights are a function of $X_i^\top\theta + \eta_i$, which we rename $h(X_i^\top\theta + \eta_i)$. For example, supposing $g(\cdot)$ was the inverse link function for the logit, then in the non-nested case,

$$w_i = h(\phi(X_i)^\top\theta + \eta_i) = \frac{1 - g(\phi(X_i)^\top\theta + \eta_i)}{g(\phi(X_i)^\top\theta + \eta_i)}\frac{N_s}{N_{pop}} = e^{-(\phi(X_i)^\top\theta+\eta_i)}\frac{N_s}{N_{pop}}$$

However the logit link is just one example; we can proceed for any choice of $g(\cdot) : \mathbb{R} \mapsto [0, 1]$. As derived above, the bias from using any weights w to approximate the mean outcome in the target population is given by $\mathbb{E}\left[\sum_i w_iZ_i\right] - \mathbb{E}[Z]$. Under linear ignorability, this is

$$\begin{aligned}
\text{bias} &= \mathbb{E}\left[\sum_i w_iZ_i\right] - \mathbb{E}[Z] && (17)\\
&= \mathbb{E}\left[\sum_i h(\phi(X_i)^\top\theta + \eta_i)Z_i\right] - \mathbb{E}[Z] && (18)\\
&= \mathbb{E}[Z]N_s\mathbb{E}\left[h(\phi(X_i)^\top\theta + \eta_i)\right] - \mathbb{E}[Z] && (19)\\
&= \mathbb{E}[Z]N_s\mathbb{E}[w_i] - \mathbb{E}[Z] && (20)\\
&= 0 && (21)
\end{aligned}$$

where (19) follows from the prior line because (i) Z is orthogonal to $\phi(X)$ and thus any function $\phi(X)^\top\theta$, and (ii) linear ignorability ensures that $Z \perp\!\!\!\perp \eta$.

B Singular value decomposition of kernel matrix

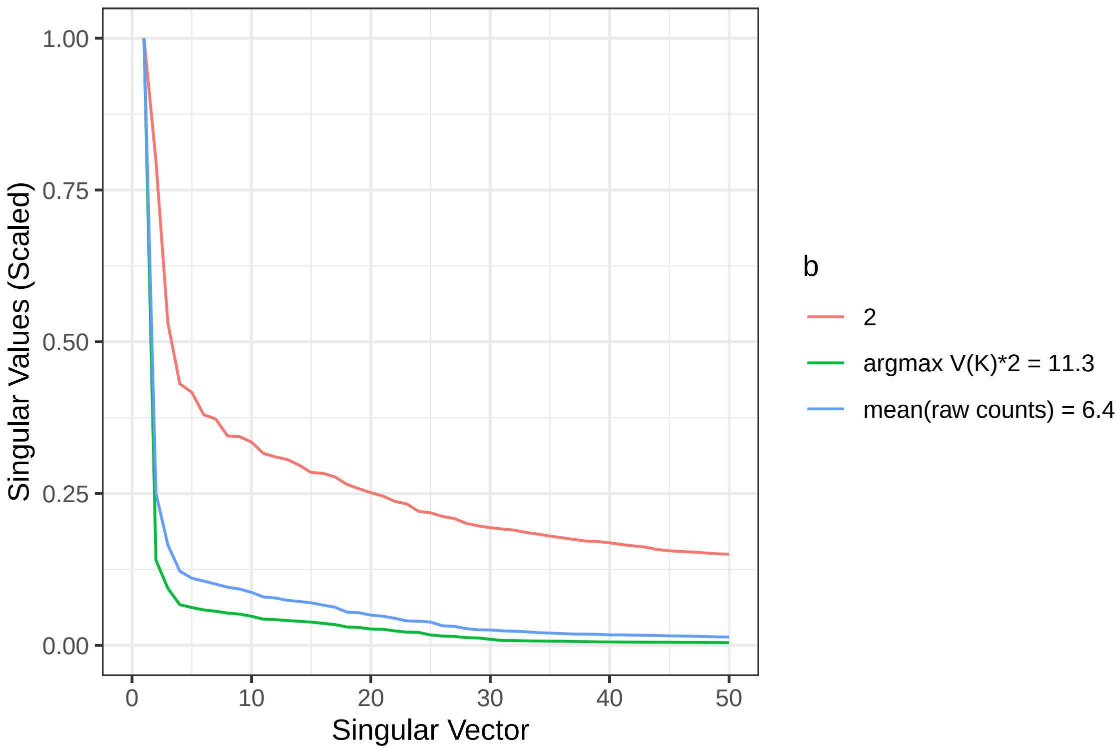

Figure B.1: Screeplot of first 50 singular values of kernel matrix for different choices of b (normalized by maximum for comparability). Note that mean(raw counts) refers to the average number of "misses" or differences between two units across all categorical variables.

C Application: Additional Details

C.1 Description of target population

Table C.1: Marginal distribution, in percentage points, of important demographics for the target population and the unweighted sample

	Target (CCES)	Unweighted Pew
Party Identification		
Dem	38.1	34.5
Ind	32.3	35.0
Rep	29.5	30.5
4-way Age Bucket		
18 to 35	28.7	19.1
36 to 50	21.3	21.4
51 to 64	29.9	31.0
65+	20.1	28.5
Gender		
Female	50.9	47.2
Male	49.1	52.8
Race/Ethnicity		
Black	11.7	8.8
Hispanic	6.5	7.6
Other	6.8	7.1
White	75.0	76.5
Region		
Midwest	23.4	22.1
Northeast	19.7	18.3
South	35.4	37.9
West	21.4	21.7
Education Level		
No HS	6.8	2.6
High school	30.6	19.7
Some college	23.0	15.2
2-year	10.6	11.4
4-year	18.7	29.1
Post-grad	10.4	22.1
Income		
Less than 20k	10.5	9.2
20-50k	29.9	25.3
50-100k	32.5	27.2
100-150k	11.5	15.2
More than 150k	5.5	14.8
Prefer not to say	10.1	8.3

Table C.1: Marginal distribution in percentage points (Cont.)

	Target (CCES)	Unweighted Pew
Religious Affiliation		
Catholic	21.3	21.6
Jewish	2.3	3.5
Not religious	32.4	22.2
Other	3.4	4.9
Protestant	40.5	47.8
Born-Again Christian		
No	67.4	69.1
Yes	32.6	30.9
Church Attendance		
Never	49.6	30.3
Weekly	27.9	34.9
Monthly	8.2	15.7
Yearly	14.3	19.2

C.2 Details of predictive outcome model

We run a regularized multinomial logistic model run on the CCES data to predict (weighted) three-way vote choice and include all variables in Table C.1 and a handful of interactions among them. Specifically, the model includes gender, 3-way party id, race/ethnicity, 6-way education, region, 6-way income, 5-way religion, 4-way church attendance, born-again status, continuous age, age^2, gender $\times$ party id, and finally age (cont.) $\times$ party id. We train this model on an 80% sample of the CCES target population. We use 10-fold cross-validation to select λ, the regularization parameter, choosing the value which produces cross-validation error that is within one standard error of the the minimum.When we classify vote choice as the highest predicted probability outcome for an individual, this model accurately classifies 79% of vote choice in the test set with the following confusion matrix (in percent):

		Reference	
	Democrat	Other	Republican
Predicted			
Democrat	86.530	47.098	14.049
Other	0.067	0.792	0.000
Republican	13.403	52.111	85.951

Table C.2: Lasso multinomial model confusion matrix (in percent) for test set observations when classifying their predicted outcome among Democrat, Republican and Other by the highest modeled probability.

When we classify test observations by their outcome with highest modeled probability, we have the most trouble accurately predicting those who choose "Other" which make up only 9% of responses. This is of minimal concern, however, because we focus on outcomes which consider only the vote share between "Democrat" and "Republican." In particular, we focus on the weighted average individual vote margin defined simply as the difference between the model's predicted probability of voting Democrat and Republican, $p(D) - p(R)$. We project this model onto the Pew sample data to predict three-way retrospective vote choice and subsequently compare the performance of different methods in weighting the sample's projected outcomes to our target. Recall that our target is the reported retrospective vote choice reported by respondents to the CCES.

C.3 Performance on the modeled vote margin outcome

Table C.3: Performance on Modeled p(D) - p(R) Vote Margin among Various Weighting Methods

	Estimate	SE	CI lower	CI upper
CCES Target	2.48	0.70	1.12	3.85
Unweighted Pew	-0.19	1.55	-3.24	2.85
Mean Calibration: Demographics	6.49	1.66	3.25	9.74
Mean Calibration: D+Educ	3.04	1.86	-0.61	6.69
Mean Calibration: All	4.31	2.12	0.16	8.47
Mean Calibration: Retrospective	3.30	1.93	-0.48	7.07
Post-Stratification: Reduced	1.00	2.33	-3.56	5.56
kpop	2.50	2.49	-2.37	7.37
kpop + MF: Demographics	2.88	2.74	-2.49	8.24
kpop + MF: D+Edu	3.09	3.02	-2.83	9.01
kpop + MF: All	3.19	2.55	-1.81	8.19

C.4 Distribution of Weights and Diagnostic Measures

Note that for ease of interpretability, all of the following use weights that sum to N_s. In this application that corresponds to 2052 pew units.

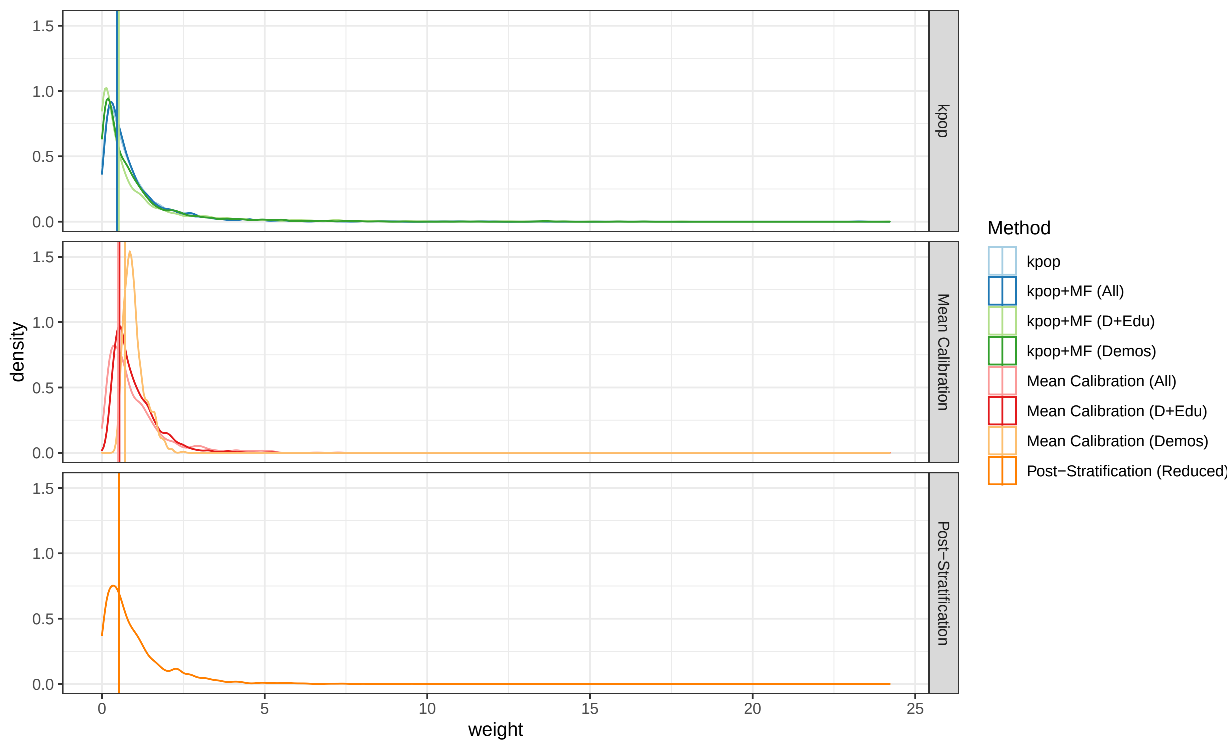

Figure C.2: Distribution of weights (for Pew sample) across the various weighting methods discussed in Section 4. Vertical lines indicate where 90% of the total sum of the weights ordered from largest to smallest occurs. Thus, the area to the right of these lines accounts for 90% of the mass of the weights.

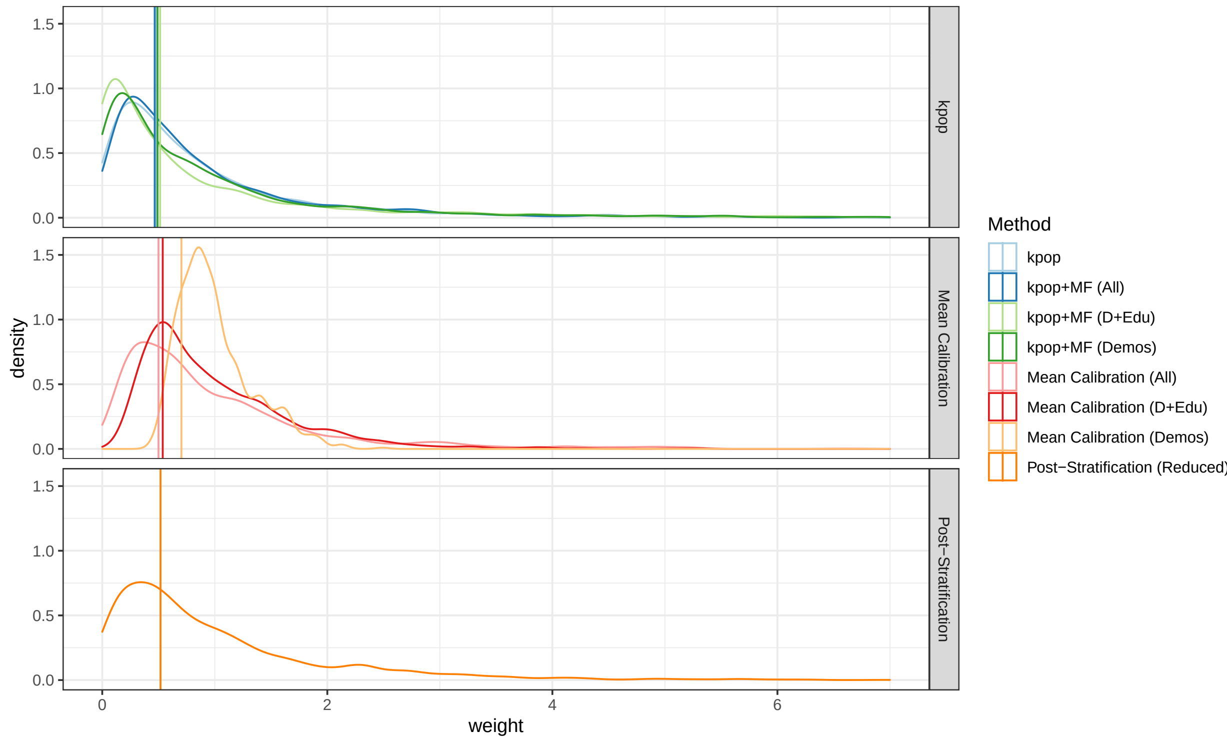

Figure C.3: Distribution of weights (for Pew sample) across the various weighting methods discussed in Section 4, focusing on a smaller range of weight magnitudes (horizontal axis). Vertical lines indicate where 90% of the total sum of the weights ordered from largest to smallest occurs. Thus, the area to the right of these lines accounts for 90% of the mass of the weights.

Table C.4: Numerical Summary of Distribution of Survey Weights

	Variance	Max	Min	IQR	Effective SS	No. Units to 90% Sum of Total (2052)
kpop	1.6009	15.2294	0.0103	0.9226	789.1844	1193
kpop+MF (Demos)	2.2313	16.6525	0.0006	0.9638	635.2498	1050
kpop+MF (D+Edu)	2.9855	24.2144	0.0001	1.0090	515.0601	901
kpop+MF (All)	1.7382	23.2793	0.0232	0.9067	749.6189	1223
Mean Calib. (Demos)	0.1082	2.4930	0.4367	0.4186	1851.8182	1721
Mean Calib. (D+Edu)	0.4404	5.2809	0.1515	0.7804	1424.8507	1535
Mean Calib. (All)	0.8639	7.3218	0.0516	0.8806	1101.1832	1362
Post-Strat (Reduced)	1.0896	9.5190	0.0050	0.9606	982.7328	1235

Table C.5: L1 and Biasbound Results for all Kpop Methods in 2016 Election Application

	L1 Original	L1 Optimal	Biasbound Original	Biasbound Optimal	Biasbound Ratio
kpop	0.0318	0.0011	0.0446	0.0069	6.4236
kpop + MF (Demos)	0.0318	0.0005	0.0446	0.0076	5.8954
kpop + MF (D+Edu)	0.0318	0.0005	0.0446	0.0078	5.7174
kpop + MF (All)	0.0318	0.0010	0.0446	0.0085	5.2332

Note: Original refers to the L1 and biasbound before any balancing is conducted. Optimal refers to the final L1 and biasbound after balance is conducted on the optimal number of dimensions of the left singular vectors of $\mathbf{K}$. The biasbound ratio reports the proportion of the original biasbound to the final optimal biasbound.

C.5 Performance Across a Range of b

Examining results across a range of values of b, we first observe that once b becomes extremely large, we have coarsened the information in the kernel matrix to reduce essentially to a linear space such that *kpop* begins to perform very similarly to mean calibration approaches. Until we reach this point, however, *kpop* results are highly consistent across a wide range of values for b.

Notably, at smaller values of b, we see how feasibility constraints directly lead to a trade off between achieving calibration on the mean first constraints and seeking balance on the left singular vectors of $\mathbf{K}$. Because these mean first results prioritize mean calibration on all available variables, for small values of b, *kpop + MF* struggles to find weights that additionally balance on more than the first handful of left singular vectors of $\mathbf{K}$. The resulting estimates therefore perform similarly to mean calibration approaches.

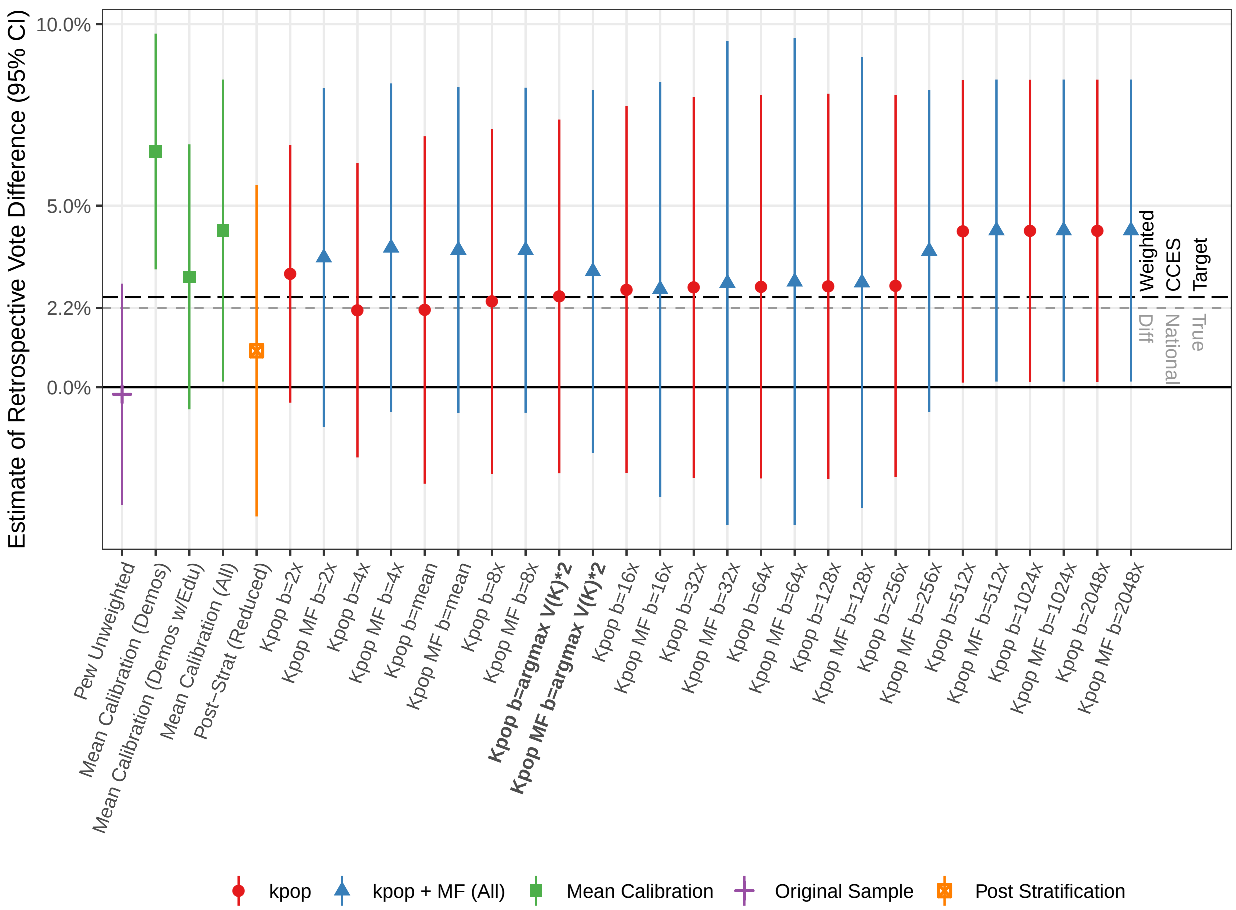

Figure C.4: Performance of Kpop across a range of choices of b with and without ebalance convergence and mean first constraints. Note that mean first results append all columns of the data and prioritize first achieving mean calibration on these columns before proceeding to balance on the left singular vectors of $\mathbf{K}$.

D Simulation evidence

While *kpop* achieves good balance and low bias estimates in our application, we employ a simulation setting to more fully investigate performance on both bias and variability. To construct a realistic simulation, we employ the same setting above, but draw samples from the (unweighted) CCES using a model-based probability partially learned from the data, and again focusing on a modeled outcome variable (vote choice).

We describe the procedure briefly here, with extensive details and code for the simulation procedure given in Supplement E. We first concatenate the CCES and Pew data, constructing an indicator for being in the (Pew) sample. We then use a regularized probit regression to estimate a probability of inclusion in the Pew sample for each unit. This model includes continuous age, age^2, the interaction of gender and party identification, the interaction of age and party identification, and finally the interaction of race, region, party identification, and 3-way education among white voters. The simulation then proceeds by taking Bernoulli draws from the CCES data to reconstruct artificial Pew-like samples. We then construct a simulated outcome by modeling two-way vote choice in the CCES data using this same model. In so doing, we ensure that the outcome is a non-linear function of our covariates and that the correlation between the simulated outcome and sample inclusion probability will result in bias when these non-linearities are not accounted for.

We then use the same nine models described in Section 4 to estimate the target population (CCES) simulated vote margin from the Pew-like samples, repeating the procedure 500 times. Results are summarized in Table D.6 and Figure D.5. Additionally, we provide a table evaluating balance on auxiliary variables in Table D.7. Mean calibration with the true selection model is shown not as a feasible estimator, but as an ideal oracle for benchmarking purposes. In terms of bias, besides mean calibration with the true selection model, the best performers are *kpop*, followed by the increasingly complicated *kpop + MF* models. This suggests that the *kpop* methods, even with mean calibration constraints, can accurately capture important higher-order interactions. Mean calibration on basic demographics alone, however, performs very poorly. Mean calibration with education is more successful, but still has more than four times the bias and over ten times the MSE of *kpop*. The small remaining bias of *kpop* is not surprising given the deliberate mis-specification in this example: both the sampling and outcome models are drawn from a different function space than the one represented linearly in $\mathbf{K}$. The goal however is to greatly reduce bias by obtaining balance over a function space that can much more nearly fit Y, R, or both.

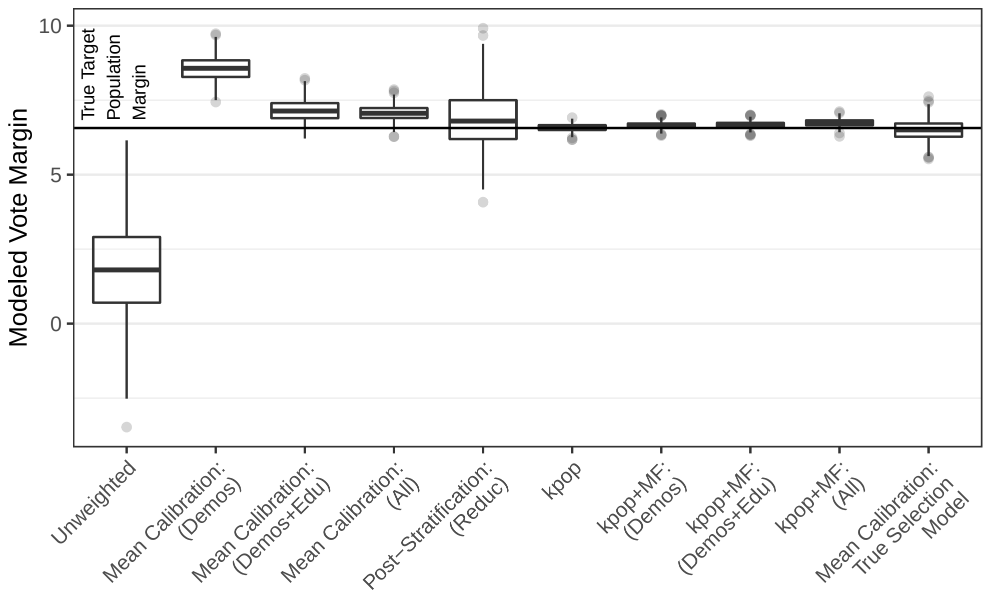

Figure D.5: Results of the simulated analysis of 2016 vote margin, based on 510 draws of Pew-like samples from the CCES data.

Table D.6: Simulation Results (arranged by MSE)

	Bias	SE	MSE	Bias Reduction
kpop	0.02	0.12	0.01	1.00
kpop + MF: (Demos)	0.09	0.11	0.02	1.02
kpop + MF: (Demos+Edu)	0.10	0.11	0.02	1.02
kpop + MF: (All)	0.18	0.12	0.05	1.04
Mean Calibration: True Selection Model	-0.07	0.34	0.12	0.98
Mean Calibration: (All)	0.50	0.26	0.32	1.11
Mean Calibration: (Demos+Edu)	0.58	0.37	0.47	1.12
Post-Stratification: (Reduc)	0.29	0.95	0.98	1.06
Mean Calibration: (Demos)	2.01	0.40	4.20	1.42
Unweighted	-4.76	1.57	25.15	0.00

Note: Bias, standard deviation, mean squared error, and bias reduction relative to the unweighted sample on our modeled vote outcome across 510 simulations of Pew-like draws from the CCES. Rows are arranged in order of increasing MSE.

Table D.7: Simulations: Average Weighted Mean Abs Error across 510 simulations, b = argmax V(K), No Pop Weights

Variable	Pew Orig	kpop	kpop + MF: (Demos)	Mean Calib: (Demos)	kpop+MF: (D+Edu)	Mean Calib: (D+Edu)	kpop+MF: (All)	Mean Calib: (All)	PS Reduc	Mean Calib: Truth
female	3.41	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.20	0.02
pid (3way)	0.90	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.14	0.01
age (4way)	0.65	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.42	0.63
race/ethnicity	0.17	0.02	0.00	0.00	0.00	0.00	0.00	0.00	1.12	0.00
region	0.29	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.45	0.00
educ (6way)	0.70	0.00	0.00	0.72	0.00	0.00	0.00	0.00	0.33	0.01
income (6way)	0.16	0.00	0.00	0.14	0.00	0.02	0.00	0.00	0.43	0.04
born	0.09	0.02	0.03	0.50	0.03	0.13	0.00	0.00	2.29	0.11
relig (5way)	0.19	0.01	0.02	0.04	0.02	0.02	0.00	0.00	0.26	0.03
pid× race	0.16	0.00	0.00	0.04	0.00	0.04	0.01	0.04	0.12	0.00
educ× pid	0.27	0.00	0.00	0.26	0.00	0.05	0.00	0.05	0.03	0.01
educ× pid× race	0.05	0.00	0.00	0.05	0.00	0.02	0.00	0.02	0.02	0.00
race× educ× reg	0.03	0.00	0.00	0.03	0.00	0.01	0.00	0.01	0.01	0.00
educ× white	1.43	0.06	0.05	1.44	0.04	0.39	0.07	0.39	0.76	0.01
midwest× white × edu	0.32	0.02	0.00	0.15	0.00	0.08	0.02	0.08	0.34	0.00
midwest× edu× race	0.08	0.00	0.00	0.02	0.00	0.01	0.00	0.01	0.02	0.00
age^2	126.41	13.41	11.76	14.54	11.66	11.69	10.58	11.46	40.65	36.16

Note: Absolute error in the distribution of categorical variables, weighted by the target population proportion for each level. Gray numbers indicate the variable was included as a calibration constraint, and so imbalances very near zero are expected. Note that all interactions with education us a three-way education coding.

E Simulations Code

In Section D we briefly described our simulation study. In this section of the supplementary materials we describe the simulations in more detail and provide replication code.

```
set.seed(9345876)

library(MASS)
library(tidyverse)
library(survey)
library(kbal)
library(parallel)
library(knitr)
library(kableExtra)
library(glmnet)
library(caret)
```

```
###### SET PARAMS  ###############
set.seed(9345876)
path_data= "/Users/Ciara/Dropbox/kpop/Updated/application/data/"
#run sims with or without population weights
POPW = FALSE
#parameters for parallel
cores_saved = 23
nsims = (detectCores()-cores_saved)*20
#ebalance parameters
tolerance = 1e-4
maxit = 500
#for runtime, increment searched dimensions of svd(K) by 5 and set min and max start poitns
increment = 5
min_num_dims = 140
max_num_dims = 500
#if want to load simulations and run only results
eval_sims = FALSE
```

First, we define the models used in the raking and post-stratification function.

```
###################### Formulas for Non Kpop Methods###############
formula_rake_demos_noeduc <- ~recode_age_bucket + recode_female + recode_race +
    recode_region + recode_pid_3way

formula_rake_demos_weduc <- ~recode_age_bucket + recode_female +
    recode_race + recode_region + recode_educ_3way + recode_pid_3way

formula_rake_all_vars <- ~recode_age_bucket + recode_female +
    recode_race + recode_region + recode_pid_3way + recode_educ +
    recode_income_5way + recode_relig_6way + recode_born + recode_attndch_4way
```

```r
formula_ps <- ~recode_age_3way + recode_female + recode_race +
    recode_region + recode_educ_wh_3way + recode_pid_3way

formula_ps_reduc <- ~recode_age_3way + recode_female +
    recode_race + recode_region + recode_pid_3way  +
    recode_income_3way + recode_born + recode_educ_wh_3way

#let's coarsen income and religion and attend church for all:
formula_ps_all <- ~recode_age_3way + recode_female +
    recode_race + recode_region + recode_pid_3way + recode_educ_3way +
    recode_income_3way + recode_relig_6way + recode_born + recode_attndch_bin
```

Next, we define our selection model. This is the formula passed to our lasso logistic regression below which models Pew sample selection. We use this model to create simulated selection probability for CCES units. We use this probability vector to sample bernoulli draws from the CCES creating Pew-like samples for the simualtions.

```r
#### (4) Selection model (also rake_demo_truth)
selection_model = as.formula(~recode_female:recode_pid_3way +
                               recode_age:recode_pid_3way +
                               recode_pid_race +
                               recode_race_reg_wh_educ +
                               recode_educ_wh_3way +
                               poly(recode_age, 3))
```

And, finally, we create some helpful functions for estimation.

```r
create_targets <- function (target_design, target_formula) {
    target_mf <- model.frame(target_formula, model.frame(target_design))
    target_mm <- model.matrix(target_formula, target_mf)
    wts <- weights(target_design)

    return(colSums(target_mm * wts) / sum(wts))
}

### Post-stratification function
## For now assumes that strata variable is already created and in
## the data set and called "strata"
postStrat <- function(survey, pop_counts, pop_w_col, strata_pass) {
    survey_counts <- survey %>%
        group_by(!!as.symbol(strata_pass)) %>%
        summarize(n = n()) %>%
        ungroup() %>%
        mutate(w_survey = n / sum(n))

    pop_counts <- pop_counts %>%
```

```
        rename(w_pop = matches(pop_w_col))

    post_strat <- pop_counts %>%
        left_join(survey_counts, by = strata_pass) %>%
        filter(!is.na(w_survey)) %>%
        ## Normalizes back to 1 after dropping
        ## empty cells
        mutate(w_pop = w_pop * 1/sum(w_pop),
               w = w_pop / w_survey) %>%
        dplyr::select(!!as.symbol(strata_pass), w)

    survey <- survey %>%
        left_join(post_strat)

    return(survey)
}


est_mean <- function(outcome, design) {
  svymean(as.formula(paste0("~", outcome)), design, na.rm = TRUE)[1]
}

var_K= function(b, n_d, diag_length){
  d <- n_d[,1] %>% pull()
  n_d <- as.vector(n_d[,2] %>% pull())
  #removing diagonal 0's from consideration
  n_d[1] <- n_d[1] - diag_length
  p_d <- n_d/ sum(n_d)
  mean_k = sum(exp(-1*d/b)*p_d)
  var_k = sum((exp(-1*d/b)-mean_k)^2 * p_d)
  return(var_k)
}
```

E.1 Data

Following our application, we use the Pew study described in Section 4.1 of the manuscript. We define the target population using the 2016 CCES, with the same restrictions described in Section 4.1, weighted using the `commonweight_vv_post`.

```
############# Load Data #####################
#these data have been cleaned already see app_modeled for how it was done
## Load Pew
pew <- readRDS(paste0(path_data, "pew_lasso_061021.rds"))
### Load Target Data
cces <- readRDS(paste0(path_data, "cces_lasso_061021.rds"))
```

```
######### Make STRATA variable in CCES and Pew  for different PS specifications ############
cces <- bind_cols(cces, cces %>%
                    unite("strata", all.vars(formula_ps), remove = FALSE) %>%
                    unite("strata_reduc", all.vars(formula_ps_reduc),
                          remove = FALSE) %>%
                    unite("strata_all", all.vars(formula_ps_all)) %>%
                    dplyr::select(strata, strata_reduc, strata_all))

pew <- bind_cols(pew, pew %>%
                   unite("strata", all.vars(formula_ps), remove = FALSE) %>%
                   unite("strata_reduc", all.vars(formula_ps_reduc),
                         remove = FALSE) %>%
                   unite("strata_all", all.vars(formula_ps_all)) %>%
                   dplyr::select(strata, strata_reduc, strata_all))
```

E.2 Generate Sampling Probabilities

In an effort to simulate a data generating process similar to our application, we create a sampling probability that resembles the relationship between the Pew study and the weighted CCES. This means our sampling probabilities, π_i are modeled as:

$$\begin{aligned}
\pi_i = \text{logit}^{-1}[&\alpha + \beta_j(\text{gender} \times \text{PID})_j + \\
&\beta_k(\text{age} \times \text{PID})_k + \\
&\beta_l(\text{educ} \times \text{white})_l + \\
&\beta_m(\text{race} \times \text{region} \times (\text{educ} \times \text{white}))_m + \\
&\gamma\text{age} + \gamma_2(age^2) + \gamma_3(age^3)]
\end{aligned} \tag{22}$$

where specifically, the term $(\text{race} \times \text{region} \times (\text{educ} \times \text{white}))$ is an interaction of region and race among non-white respondents and an interaction of region and three way education among white voters.

We construct the coefficients by:

1. Run a regularized logitic model, using the CCES weights, using the selection model defined above.
2. Select the value of the regularization parameter λ through 10-fold cross-validation. We use the value that is within one standard error of the minimum cross-validation error.
3. Project this model onto the CCES data, again using the CCES weights, to get sample selection probabilities for each CCES unit.

```
##################### LASSO: Selection #############################
# Stack data with S = 1 indicating Pew
stack_data <- data.frame(bind_rows(pew, cces),
                         S = c(rep(1, nrow(pew)), rep(0, nrow(cces))))
```

```r
#build data
mod <- model.matrix(selection_model, data = stack_data)
## Remove columns where Pew missing strata (none but good check)
mod <- mod[, apply(mod[stack_data$S == 1, ], 2, sum) != 0]
## Remove columns where CCES missing Strata
mod <- mod[, apply(mod[stack_data$S == 0, ], 2, sum) != 0]


#Run lambda from 10 fold default CV
#Note: had to remove the intercept in mod and have glmnet add it back
#other wise even with intercept false it was adding two intercepts
lasso_lambda <- cv.glmnet(x= mod[,-1],
                          y = as.matrix(stack_data$S),
                          alpha = 1,
                          family = "binomial",
                          intercept = TRUE)

lasso_include <- glmnet(x= mod[,-1],
                        y = as.matrix(stack_data$S),
                        alpha = 1,
                        lambda = lasso_lambda$lambda.min,
                        weights = if(POPW){ c(rep(1, nrow(pew)),
                                              cces$commonweight_vv_post)} else {NULL},
                        family = "binomial",
                        intercept = TRUE)

#take a look at coefs
lasso_include_coefs <- coef(lasso_include)
res <- as.matrix(lasso_include_coefs)
#View(res)
#how many regularized?
sum(lasso_include_coefs == 0)/ncol(mod)
```

```
## [1] 0.2741935
```

```r
## Getting Selection Probability
#probability of being in pew for cces
lasso_pinclude = predict(lasso_include,
                         s= lasso_lambda$lambda.min,
                         type = "response",
                         newx = mod[stack_data$S == 0,-1])
p_include <- lasso_pinclude
#this will be roughly our sample size in the simulations
sum(p_include)
```

```
## [1] 1946.247
```

```
########### Examine Sampling Inclusion Results
mean(p_include); max(p_include); min(p_include)
```

```
## [1] 0.04331539

## [1] 0.2255599

## [1] 0.01279477
```

```
#correlation with outcome
cor(cces$diff_cces_on_cces, p_include)
```

```
##               1
## [1,] -0.1673515
```

We see that sampling probabilities are bounded away from 0 and 1, and that the mean response rate is close to what we typically see in surveys (i.e. around 5%).

E.3 Simulations

The target population is constant across simulations, so we first create the population targets for raking and post-stratification.

```
#################### Targets ###################
if(POPW) {
  cces_svy <- svydesign(ids = ~1, weights = ~commonweight_vv_post, data = cces)
} else {
  cces_svy <- svydesign(ids = ~1, data = cces)
}
```

```
## Warning in svydesign.default(ids = ~1, data = cces): No weights or probabilities
## supplied, assuming equal probability
```

```
targets_rake_demos_noeduc <- create_targets(cces_svy,
                                            formula_rake_demos_noeduc)
targets_rake_demos_weduc <- create_targets(cces_svy, formula_rake_demos_weduc)
targets_rake_all_vars <- create_targets(cces_svy,
                                        formula_rake_all_vars)
targets_demo_truth <- create_targets(cces_svy, selection_model)


## Make table of Population Counts for post-stratification for manual ps function
cces_counts <- cces %>%
  group_by(strata) %>%
  summarize(n = if(!POPW) {n()} else {sum(commonweight_vv_post, na.rm = TRUE)}) %>%
  ungroup() %>%
  mutate(w = n / sum(n, na.rm = TRUE))

cces_reduc_counts <- cces %>%
```

```r
  group_by(strata_reduc) %>%
  summarize(n = if(!POPW) {n()} else {sum(commonweight_vv_post, na.rm = TRUE)}) %>%
  ungroup() %>%
  mutate(w_reduc = n / sum(n, na.rm = TRUE))

cces_all_counts <- cces %>%
  group_by(strata_all) %>%
  summarize(n = if(!POPW) {n()} else {sum(commonweight_vv_post, na.rm = TRUE)}) %>%
  ungroup() %>%
  mutate(w_all = n / sum(n, na.rm = TRUE))
```

In each simulation, we do the following:

1. Create a simulated survey sample of respondents using the sampling probability (π_i, `p_include`) defined above. We estimate the unweighted margin.
2. Run the following five survey weighting models:
   - `rake_demos_noeduc`: Raking on the margins of: `recode_age_bucket`, `recode_female`, `recode_race`, `recode_region`, `recode_pid_3way`.
   - `rake_demos_weduc`: Raking on the margins of: `recode_age_3way`, `recode_female`, `recode_race`, `recode_region`, `recode_pid_3way`, `recode_educ_3way`.
   - `rake_all`: Raking on the margins of: `recode_age_3way`, `recode_female`, `recode_race`, `recode_region`, `recode_pid_3way`, `recode_educ`, `recode_income_5way`, `recode_relig_6way`, `recode_born`, `recode_attndch_4way`
   - `post_stratification`: Post-stratification on intersectional strata of: `recode_age_3way`, `recode_female`, `recode_race`, `recode_region`, `recode_pid_3way`, `recode_income_3way`, `recode_born`, `recode_educ_wh_3way`.
   - `rake_truth`: Raking on the true selection model.
3. Run kpop methods:
   - `kpop` (X matrix given all available variables as described in C.1),
   - `kpop+MF (Demos)` (same X matrix, with additional MF constraints on the variables in `rake_demos_noeduc`)
   - `kpop+MF (Demos + Edu)` (same X matrix, with additional MF constraints on the variables in `rake_demos_weduc`)
   - `kpop+MF (All)` (same X matrix, with additional MF constraints on the variables in `rake_all`)
4. Record weighted outcome estimates as well as margins across a large number of variables and interactions. Recall that the outcome of interest is restrospective vote difference $p(D) - p(R)$ as described in 4.1 and C.2.

```r
########################### RUN SIMS ##################################
#save mem
rm(pew, lasso_pinclude, lasso_include,lasso_lambda,
   stack_data, mod)


margins_formula <- ~recode_vote_2016 +
```

```
  mod_cces_on_cces_pR + mod_cces_on_cces_pD + mod_cces_on_cces_pO+
  diff_cces_on_cces + margin_cces_on_cces +
  recode_pid_3way +
  recode_female + recode_race +recode_region + recode_educ + recode_relig_6way +
  recode_born + recode_attndch_4way + recode_income_5way +
  recode_age_bucket + recode_age + recode_agesq + recode_agecubed + recode_age_factor +
  recode_race_educ_reg + recode_educ_wh_3way +
  recode_educ_pid_race +
  recode_pid_race +
  recode_educ_pid +
  recode_midwest_edu_race +
  recode_midwest_wh_edu


system.time({
  sims <- mclapply(1:nsims, function(nsim) {

    cat(paste("====================  SIM:",nsim,
              "==================== \n"))
    sample <- rbinom(nrow(cces), 1, p_include)

    survey_sim <- cces[sample == 1, ]
    survey_design <- suppressWarnings(svydesign(ids = ~1, data = survey_sim))

    ############################################
    ## Unweighted estimate
    ############################################

    unweighted <- est_mean("diff_cces_on_cces", survey_design)
    ############################################
    ## Sample size
    ############################################
    n <- sum(sample)

    ############################################
    ## Raking on demographics (no education)
    ############################################
    rake_demos_noeduc_svyd <- calibrate(design = survey_design,
                                        formula = formula_rake_demos_noeduc,
                                        population = targets_rake_demos_noeduc,
                                        calfun = "raking")

    rake_demos_noeduc <- est_mean("diff_cces_on_cces", rake_demos_noeduc_svyd)
    ############################################
    #### Raking on demographics (with education)
```

```r
############################################
rake_demos_weduc_svyd <- calibrate(design = survey_design,
                                   formula = formula_rake_demos_weduc,
                                   population = targets_rake_demos_weduc,
                                   calfun = "raking")

rake_demos_weduc <- est_mean("diff_cces_on_cces", rake_demos_weduc_svyd)


############################################
#### Raking on everything
############################################
rake_all_svyd <- calibrate(design = survey_design,
                           formula = formula_rake_all_vars,
                           population = targets_rake_all_vars,
                           calfun = "raking")

rake_all <- est_mean("diff_cces_on_cces", rake_all_svyd)


############################################
## Post-stratification: Old Formula
############################################
post_stratification_svyd = svydesign(~1, data = postStrat(survey_sim,
                                                          cces_counts, "w",
                                                          strata_pass = "strata"),
                                     weights = ~w)

post_stratification <- est_mean("diff_cces_on_cces", post_stratification_svyd)

############################################
## Post-stratification: Reduced
############################################
post_strat_reduc_svyd = svydesign(~1, data = postStrat(survey_sim,
                                                       cces_reduc_counts, "w_reduc",
                                                       strata_pass = "strata_reduc"),
                                  weights = ~w)

post_strat_reduc <- est_mean("diff_cces_on_cces", post_strat_reduc_svyd)

############################################
## Post-stratification: All
############################################
post_strat_all_svyd = svydesign(~1, data = postStrat(survey_sim,
                                                     cces_all_counts, "w_all",
```

```
                                                strata_pass = "strata_all"),
                                weights = ~w)

  post_strat_all <- est_mean("diff_cces_on_cces", post_strat_all_svyd)

  ############################################
  #### Raking on true model
  ############################################
  #very messy error catching for the moment just a stop gap to see how things look
  rake_truth_svyd <- try(calibrate(design = survey_design,
                                   formula = selection_model,
                                   population = targets_demo_truth,
                                   calfun = "raking",
                                   epsilon = .009), silent = T)

    rake_truth <- est_mean("diff_cces_on_cces", rake_truth_svyd)
    truth_margins = svymean(margins_formula, rake_truth_svyd)

  ############################################
  ## Kpop: Categorical Data + b = argmax V(K)
  ############################################
# Select the covariates for use in Kbal: updated cat data no cont age
#one-hot coded for cat kernel
kbal_data <- bind_rows(survey_sim %>% dplyr::select(recode_age_bucket,
                                                    recode_female,
                                                    recode_race,
                                                    recode_region,
                                                    recode_pid_3way,
                                                    recode_educ,

                                                    recode_income_5way,
                                                    recode_relig_6way,
                                                    recode_born,
                                                    recode_attndch_4way),
                       cces %>% dplyr::select(recode_age_bucket,
                                              recode_female,
                                              recode_race,
                                              recode_region,
                                              recode_pid_3way,
                                              recode_educ,

                                              recode_income_5way,
                                              recode_relig_6way,
                                              recode_born,
                                              recode_attndch_4way))
```

```r
kbal_data <- data.frame(lapply(kbal_data, as.factor))
kbal_data <- model.matrix(~ ., kbal_data,
                          contrasts.arg = lapply(kbal_data,
                                                 contrasts, contrasts=FALSE))
kbal_data <- kbal_data[, -1]
kbal_data_sampled <- c(rep(1, nrow(survey_sim)), rep(0, nrow(cces)))

K <- makeK(kbal_data, b=2, useasbases = kbal_data_sampled,
           linkernel = FALSE, scale = FALSE)
raw_counts <- -log(K)

#### now get max var(K) b
#run optim: much faster than table

n_d <- data.frame(diff = c(raw_counts)) %>% group_by(diff) %>% summarise(n())
res = optimize(var_K, n_d, length(diag(K)),
                 interval=c(0,2000), maximum=TRUE)
#note with one hot encoding this needs to be * by 2
b_maxvar <- res$maximum

##### Demos Constraint
rake_demos_constraint <- bind_rows(survey_sim %>% dplyr::select(recode_age_bucket,
                                                                recode_female,
                                                                recode_race,
                                                                recode_region,
                                                                recode_pid_3way),
                                   cces %>% dplyr::select(recode_age_bucket,
                                                          recode_female,
                                                          recode_race,
                                                          recode_region,
                                                          recode_pid_3way))%>%
  model.matrix(as.formula("~."), .)
rake_demos_constraint <- rake_demos_constraint[,-1]
rake_demos_constraint <- scale(rake_demos_constraint)

rake_demos_wedu_constraint <- bind_rows(survey_sim %>% dplyr::select(recode_age_bucket,
                                                                     recode_female,
                                                                     recode_race,
                                                                     recode_region,
                                                                     recode_pid_3way,
                                                                     recode_educ),
                                        cces %>% dplyr::select(recode_age_bucket,
                                                               recode_female,
                                                               recode_race,
                                                               recode_region,
```

```
                                                              recode_pid_3way,
                                                              recode_educ))%>%
  model.matrix(as.formula("~."), .)

rake_demos_wedu_constraint <- rake_demos_wedu_constraint[,-1]
rake_demos_wedu_constraint <- scale(rake_demos_wedu_constraint)

rake_all_constraint <- bind_rows(survey_sim %>% dplyr::select(recode_age_bucket,
                                                             recode_female,
                                                             recode_race,
                                                             recode_region,
                                                             recode_pid_3way,
                                                             recode_educ,
                                                             recode_income_5way,
                                                             recode_relig_6way,
                                                             recode_born,
                                                             recode_attndch_4way),
                                 cces %>% dplyr::select(recode_age_bucket,
                                                       recode_female,
                                                       recode_race,
                                                       recode_region,
                                                       recode_pid_3way,
                                                       recode_educ,
                                                       recode_income_5way,
                                                       recode_relig_6way,
                                                       recode_born,
                                                       recode_attndch_4way)) %>%
  model.matrix(as.formula("~."), .)
rake_all_constraint <- rake_all_constraint[,-1]
rake_all_constraint <- scale(rake_all_constraint)

#option to check more b values
#b <- c(b_mean, b_maxvar*2, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048)
b <- c(b_maxvar*2)
#save memory
rm(K, res, raw_counts)
#run kpop with specified choice(s) of b
kpop <- lapply(1:length(b), function(i) {
  #### DEFAULT ######
  kbal_est <- kbal(allx=kbal_data,
                   sampled = kbal_data_sampled,
                   b = b[i],
                   scale_data = FALSE,
                   drop_multicollin = FALSE,
                   incrementby = increment,
```

```
                    meanfirst = FALSE,
                    ebal.tol = tolerance,
                    ebal.maxit = maxit,
                    minnumdims = min_num_dims,
                    maxnumdims = max_num_dims,
                    linkernel = if(TEST){TRUE} else{ FALSE},
                    sampledinpop = FALSE,
                    fullSVD = TRUE)

kpop_svyd <- svydesign(~1, data = survey_sim,
                          weights = kbal_est$w[kbal_data_sampled ==1])

kpop <- est_mean("diff_cces_on_cces", kpop_svyd)

#save memory by saving only the svd to re use
svdK = kbal_est$svdK
numdims = kbal_est$numdims
biasbound_r = kbal_est$biasbound.orig/kbal_est$biasbound.opt
biasbound = kbal_est$biasbound.opt

#CONVERGED
dist_record = data.frame(t(kbal_est$dist.record))
min_converged = dist_record[which.min(dist_record[dist_record$Ebal.Convergence ==1,"BiasBou

rm(kbal_est)

#### CONVG ###
kbal_est_conv <- kbal(allx=kbal_data,
                       K.svd = svdK,
                       sampled = kbal_data_sampled,
                       numdims = min_converged,
                       ebal.tol = tolerance,
                       ebal.maxit = maxit,
                       minnumdims = min_num_dims,
                       maxnumdims = max_num_dims,
                       scale_data = FALSE,
                       drop_multicollin = FALSE,
                       incrementby = increment,
                       meanfirst = FALSE,
                       sampledinpop = FALSE,
                       ebal.convergence = TRUE)
  kpop_svyd_conv <- svydesign(~1, data = survey_sim,
                               weights = kbal_est_conv$w[kbal_data_sampled ==1])
  kpop_conv <- est_mean("diff_cces_on_cces", kpop_svyd_conv)
```

```r
  numdims_conv = kbal_est_conv$numdims
  biasbound_r_conv = kbal_est_conv$biasbound.orig/kbal_est_conv$biasbound.opt
  biasbound_conv = kbal_est_conv$biasbound.opt
  rm(kbal_est_conv)


####### aMF #######
kbal_mf_est <- kbal(K.svd = svdK,
                    allx=kbal_data,
                    sampled = kbal_data_sampled,
                    ebal.tol = tolerance,
                    ebal.maxit = maxit,
                    minnumdims = min_num_dims,
                    maxnumdims = max_num_dims,
                    scale_data = FALSE,
                    drop_multicollin = FALSE,
                    incrementby = increment,
                    meanfirst = TRUE,
                    sampledinpop = FALSE)

kpop_mf_svyd <- svydesign(~1, data = survey_sim,
                          weights = kbal_mf_est$w[kbal_data_sampled ==1])

kpop_mf <- est_mean("diff_cces_on_cces", kpop_mf_svyd)

mfnumdims = kbal_mf_est$numdims
mf_appended_dims = kbal_mf_est$meanfirst.dims
biasbound_r_mf = kbal_mf_est$biasbound.orig/kbal_mf_est$biasbound.opt
biasbound_mf = kbal_mf_est$biasbound.opt
rm(kbal_mf_est)

#########MF: DEMOS constraint method:
kbal_demos_est <- kbal(K.svd = svdK,
                       allx=kbal_data,
                       sampled = kbal_data_sampled,
                       ebal.tol = tolerance,
                       ebal.maxit = maxit,
                       minnumdims = min_num_dims,
                       maxnumdims = max_num_dims,
                       scale_data = FALSE,
                       drop_multicollin = FALSE,
                       incrementby = increment,
                       constraint = rake_demos_constraint,
                       meanfirst = FALSE,
                       sampledinpop = FALSE)
```

```
kpop_demos_svyd <- svydesign(~1, data = survey_sim,
                             weights = kbal_demos_est$w[kbal_data_sampled ==1])

kpop_demos <- est_mean("diff_cces_on_cces", kpop_demos_svyd)

numdims_demos = kbal_demos_est$numdims
biasbound_r_demos = kbal_demos_est$biasbound.orig/kbal_demos_est$biasbound.opt
biasbound_demos = kbal_demos_est$biasbound.opt
rm(kbal_demos_est)


######### MF: DEMOS + EDU  constraint method:
kbal_demos_wedu_est <- kbal(K.svd = svdK,
                            allx=kbal_data,
                            sampled = kbal_data_sampled,
                            ebal.tol = tolerance,
                            ebal.maxit = maxit,
                            minnumdims = min_num_dims,
                            maxnumdims = max_num_dims,
                            scale_data = FALSE,
                            drop_multicollin = FALSE,
                            incrementby = increment,
                            constraint = rake_demos_wedu_constraint,
                            meanfirst = FALSE,
                            sampledinpop = FALSE)
kpop_demos_wedu_svyd <- svydesign(~1, data = survey_sim,
                                  weights = kbal_demos_wedu_est$w[kbal_data_sampled ==1])

kpop_demos_wedu <- est_mean("diff_cces_on_cces", kpop_demos_wedu_svyd)

numdims_demos_wedu = kbal_demos_wedu_est$numdims
if(is.null(numdims_demos_wedu)) {numdims_demos_wedu = c("dn_converge") }
biasbound_r_demos_wedu = kbal_demos_wedu_est$biasbound.orig/kbal_demos_wedu_est$biasbound.o
biasbound_demos_wedu = kbal_demos_wedu_est$biasbound.opt
rm(kbal_demos_wedu_est)


#########MF ALL constraint method:
kbal_all_est <- kbal(K.svd = svdK,
                     allx=kbal_data,
                     sampled = kbal_data_sampled,
                     ebal.tol = tolerance,
                     ebal.maxit = maxit,
                     minnumdims = min_num_dims,
                     maxnumdims = max_num_dims,
```

```
                         scale_data = FALSE,
                         drop_multicollin = FALSE,
                         incrementby = increment,
                         constraint = rake_all_constraint,
                         meanfirst = FALSE,
                         sampledinpop = FALSE)
kpop_all_svyd <- svydesign(~1, data = survey_sim,
                                weights = kbal_all_est$w[kbal_data_sampled ==1])

kpop_all <- est_mean("diff_cces_on_cces", kpop_all_svyd)

numdims_all = kbal_all_est$numdims
if(is.null(numdims_all)) {numdims_all = c("dn_converge") }
biasbound_r_all = kbal_all_est$biasbound.orig/kbal_all_est$biasbound.opt
biasbound_all = kbal_all_est$biasbound.opt
rm(kbal_all_est)
rm(svdK)

##### return
out = list()
out$sims = data.frame(b[i],
                      kpop,
                      kpop_mf,
                      kpop_conv,
                      kpop_demos,
                      kpop_demos_wedu,
                      kpop_all,
                      bb = biasbound,
                      bbr = biasbound_r,
                      bb_conv = biasbound_conv,
                      bbr_conv = biasbound_r_conv,
                      bb_mf = biasbound_mf,
                      bbr_mf = biasbound_r_mf,
                      bb_demos = biasbound_demos,
                      bbr_demos = biasbound_r_demos,
                      bb_demos_wedu = biasbound_demos_wedu,
                      bbr_demos_wedu = biasbound_r_demos_wedu,
                      bb_all = biasbound_all,
                      bbr_all = biasbound_r_all,
                      numdims,
                      numdims_conv,
                      mfnumdims,
                      mf_appended_dims,
                      numdims_demos,
                      numdims_demos_wedu,
```

```
                      numdims_all)
#weights
out$weights = list(
                   b[i],
                   kpop_w = weights(kpop_svyd),
                   kpop_w_conv = weights(kpop_svyd_conv),
                   kpop_mf_w = weights(kpop_mf_svyd),
                   kpop_demos_w = weights(kpop_demos_svyd),
                   kpop_demos_wedu_w = weights(kpop_demos_wedu_svyd),
                   kpop_all_w = weights(kpop_all_svyd))

######## Margins ########

out$km <- round(cbind(
                      b = b[i]/100,
                      kpop = svymean(margins_formula, kpop_svyd),
                      kpop_conv = svymean(margins_formula, kpop_svyd_conv),
                      kpop_mf = svymean(margins_formula, kpop_mf_svyd),
                      kpop_demos = svymean(margins_formula, kpop_demos_svyd),
                      kpop_demos_wedu = svymean(margins_formula, kpop_demos_wedu_svyd),
                      kpop_all = svymean(margins_formula, kpop_all_svyd)) * 100,
                4)

rm(kpop_svyd, kpop_mf_svyd, kpop_svyd_conv, kpop_demos_svyd,
   kpop_demos_wedu_svyd, kpop_all_svyd)

return(out)
})

############################################ OUTPUT
out = list()
out$sims = cbind(nsim,
               n,
               unweighted,
               rake_demos_noeduc,
               rake_demos_weduc,
               rake_all,
               post_stratification,
               post_strat_reduc,
               post_strat_all,
               rake_truth,
               lapply(kpop, `[[`,1) %>% bind_rows())


margin <- round(cbind(sample = svymean(margins_formula, survey_design),
```

```r
                        cces =  svymean(margins_formula, cces_svy),
                        rake_demos_noeduc = svymean(margins_formula,
                                                    rake_demos_noeduc_svyd),
                        rake_demos_weduc = svymean(margins_formula,
                                                   rake_demos_weduc_svyd),
                        rake_all = svymean(margins_formula,
                                           rake_all_svyd),
                        post_stratification = svymean(margins_formula,
                                                      post_stratification_svyd),
                        post_strat_reduc = svymean(margins_formula,
                                                   post_strat_reduc_svyd),
                        post_strat_all = svymean(margins_formula,
                                                 post_strat_all_svyd),
                        rake_truth = truth_margins) * 100, 5)

      #these are just means so let's not multiply by 100
      margin["recode_age",] <- margin["recode_age",]/100
      margin["recode_agesq",] <- margin["recode_agesq",]/100
      margin["recode_agecubed",] <- margin["recode_agecubed",]/100


      margin = cbind(margin,
                     as.data.frame(sapply(kpop, `[`, 3)))

      margin <- margin[,grepl("km.kpop", colnames(margin))|
                         !grepl("km.", colnames(margin)) ]

      colnames(margin)[grepl("km.kpop", colnames(margin))] <- unlist(lapply(b, function(x)
          c(paste0("kpop_b", round(x,3)),
            paste0("kpop_cvg_b", round(x,3)),
            paste0("kpop_mf_b", round(x,3)),
            paste0("kpop_demos_b", round(x,3)),
            paste0("kpop_demos_wedu_b", round(x,3)) ,
            paste0("kpop_all_b", round(x,3)) ) ))

    out$margins = margin

    return(out)

  }, mc.cores = detectCores() - cores_saved)
})


good = which(lapply(sims, function (x) return(class(x))) == "list")
save(sims,tolerance, maxit, increment, min_num_dims,
```

```
    file = paste0("./cat_sims_modeled_outcome",POPW, "_m",maxit, "_t",tolerance, "_inc",
                  increment, "mindims", min_num_dims,
                  Sys.Date(),
                  "_nsims", length(good),
                  ".RData"))
```