Getting the most out of surveys: Multilevel Regression And Poststratification

Joseph T. Ornstein*

Abstract

Good causal inference requires good measurement; even the most thoughtfully designed research can be derailed by noisy measures and survey error. Because policy scholars are often interested in public opinion as a key dependent or independent variable, paying careful attention to the sources of measurement error from surveys is an essential step for detecting causation. This chapter introduces multilevel regression and poststratification (MRP), an approach to adjusting public opinion estimates to account for observed imbalances between the survey sample and population of interest. It covers the history of MRP, recent advances, a running example with code, and concludes with a discussion of best practices and limitations of the approach.

TODO:

- A non-technical "narrative" section, which links the chapter to the goal of detecting causation in the social sciences. Good measurement is as central to causal inference as a good research design, both for reasons of precision (better measurement implies less noisy estimands), and external validity. So much of the data we use for causal inference in the social sciences comes from surveys. Worried about whether the survey is representative of the population you're interested in, or perhaps you're interested in a subpopulation or small area and you don't have a representative survey data for that group. MRP helps by making use of all the information available in a survey, making inferences about individuals *not* in the survey based on the characteristics of individuals *in* the survey.

Learning Objectives

By the end of this chapter, you will be able to:

- Explain the motivation for MRP and the circumstances under which it is appropriate to implement.
- Describe the two steps in producing MRP estimates: model fitting and postsratification.
- Generate MRP estimates by adapting the provided sample code.
- Implement more sophisticated variants of MRP, including stacked regression and postratification (SRP) or multilevel regression and synthetic poststratification (MrsP) where appropriate.

Introduction

Often researchers would like to measure public opinion on some policy issue, but the surveys we use to do so are unrepresentative in some important way. Perhaps their respondents come from a convenience sample (Wang et al. 2015), or non-response bias skews an otherwise random sample. Or perhaps the data is representative of some larger population (i.e. a country-level random sample), but contains too few observations to make inferences about a subgroup of interest. Even the largest US public opinion surveys do not have enough respondents to make reliable inferences about lower-level political entities like states or

*Department of Political Science, University of Georgia

municipalities. Conclusions drawn from low frequency observations – even in a large sample survey – can be wildly misleading (Ansolabehere, Luks, and Schaffner 2015).

This presents a challenge for public opinion research: how to take unrepresentative survey data and adjust it so that it is useful for our particular research question. In this chapter, I will demonstrate a method called **multilevel regression and poststratification** (MRP). Using this approach, the researcher first constructs a model of public opinion (multilevel regression) and then reweights the model's predictions based on the observed characteristics of the population of interest (poststratification). In the sections to follow, I will describe this approach in detail, and will accompany this explanation with replication code in the `R` statistical language.

MRP was first introduced by Gelman and Little (1997), and in the subsequent decades it has helped address a diverse set of research questions in political science. These range from generating election forecasts using unrepresentative survey data (Wang et al. 2015) to assessing the responsiveness of state (Lax and Phillips 2012) and local policymakers (Tausanovitch and Warshaw 2014) to their constituents' policy preferences.

In the following sections, I will illustrate how MRP can improve estimates of small area public opinion. Our running example will be drawn from the Cooperative Election Study (Schaffner, Ansolabehere, and Luks 2021), a 50,000+ respondent study of voters in the United States. The 2020 wave of the study includes a question asking respondents whether they support a policy that would "decrease the number of police on the street by 10 percent, and increase funding for other public services." Since police reform is a policy issue on which US local governments have a significant amount of autonomy, it would be useful to know how opinions on this issue vary from place to place without having to conduct separate, costly surveys in each area.

As we will see, the accuracy of our MRP estimates depends critically on whether the first-stage model makes good out-of-sample predictions. The best first-stage models are *regularized* (Gelman 2018) to avoid both over- and under-fitting to the survey data. Regularized ensemble models (Ornstein 2020) with group-level predictors tend to produce the best estimates, especially when trained on large datasets.

Running Example

To demonstrate how MRP works, we'll consider an example where we know the true population-level estimands, and can explore how various refinements to the method can improve predictive accuracy. This approach mirrors Buttice and Highton (2013), who use disaggregated ressponses from large-scale US survey of voters as the target. The Cooperative Election Study data is available here, and we'll be using a tidied version of the dataset created by the `R/cleanup-ces-2020.R` script.[1]

```
library(tidyverse)
library(ggrepel)

load('data/CES-2020.RData')
```

This tidied version of the data only includes the 33 states with at least 500 respondents. First, let's plot the percent of CES respondents who supported "defunding" the police[2] by state.

```
truth <- ces %>%
  group_by(abb) %>%
  summarize(truth = mean(defund_police))

truth %>%
  mutate(abb = fct_reorder(abb, truth)) %>%
  ggplot(mapping = aes(x=truth, y=abb)) +
  geom_point(alpha = 0.7) +
```

[1] All replication code and data will be made available on a public repository. Throughout, I will use R functions from the "tidyverse" to make the code more human-readable.

[2] Obviously that phrase means different things to different people. In this case, we'll stick with the CES proposed policy of reducing police staffing by 10% and diverting those expenditures to other priorities.

```
  labs(x = 'Percent Who Support Police Reform Policy',
       y = 'State') +
  theme_minimal()
```

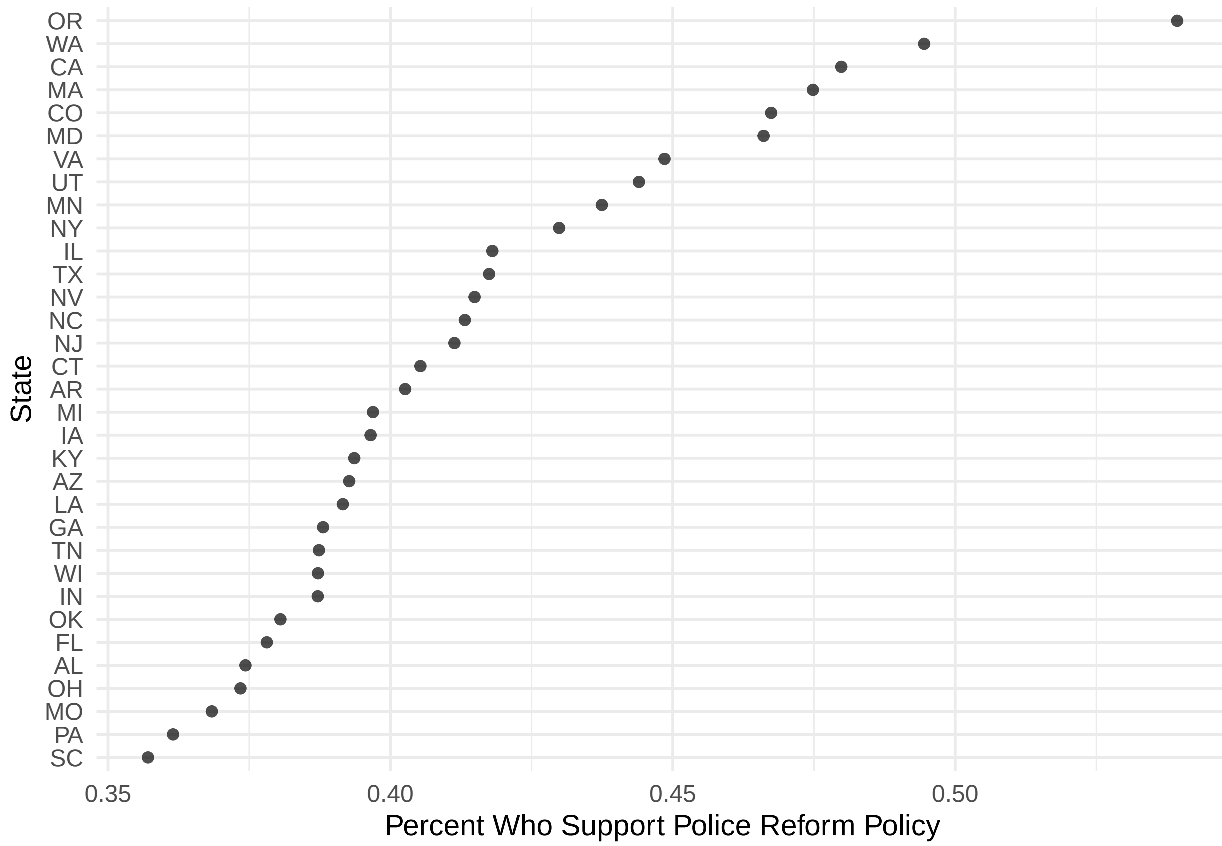

Figure 1: The percent of CES respondents in each state who support reducing police budgets. These are our target estimands.

Oregon is the only state where a majority of respondents supported this policy proposal. And note that Figure 1 likely *overstates* the percent of the total population that support such a policy, since self-identified Democrats are over-represented in the CES sample. But nevertheless, these population-level parameters will be a useful target to estimate with MRP.

Draw a Sample

Suppose that we did not have access to the entire CES dataset, but only to a random sample of 1,000 respondents. How good of a job can we do at estimating those state-level means?

```
sample_data <- ces %>%
  slice_sample(n = 1000)
```

```
sample_summary <- sample_data %>%
  group_by(abb) %>%
  summarize(estimate = mean(defund_police),
            num = n())

sample_summary
```

```
## # A tibble: 33 x 3
##    abb   estimate   num
##    <chr>    <dbl> <int>
##  1 AL       0.55     20
##  2 AR       0         4
##  3 AZ       0.438    16
##  4 CA       0.435    85
##  5 CO       0.478    23
##  6 CT       0.375     8
##  7 FL       0.402    87
##  8 GA       0.346    26
##  9 IA       0.308    13
## 10 IL       0.28     50
## # ... with 23 more rows
```

In a sample with only 1,000 respondents, there are several states with very few (or no) respondents. Notice, for example, that this sample includes only four respondents from Arkansas, of whom zero support reducing police budgets. Simply disaggregating and taking sample means is unlikely to yield good estimates, as you can see by comparing those sample means against the truth.

```r
# a function to plot the state-level estimates against the truth
compare_to_truth <- function(estimates, truth){

  d <- left_join(estimates, truth, by = 'abb')

  ggplot(data = d,
         mapping = aes(x=estimate,
                       y=truth,
                       label=abb)) +
  geom_point(alpha = 0.5) +
  geom_text_repel() +
  theme_minimal() +
  geom_abline(intercept = 0, slope = 1, linetype = 'dashed') +
  labs(x = 'Estimate',
       y = 'Truth',
       caption = paste0('Correlation = ', round(cor(d$estimate, d$truth), 2),
                        ', Mean Absolute Error = ', round(mean(abs(d$estimate - d$truth)), 3)))
}

compare_to_truth(sample_summary, truth)
```

These are clearly poor estimates of state-level public opinion. The four respondents from Arksansas simply do not give us enough information to adequately measure public opinion in that state. But one of the key insights behind MRP is that the respondents from Arkansas are not the only respondents who can give us information about Arkansas! There are other respondents in, for example, Missouri, that are similar to Arkansas residents on their observed characteristics. If we can determine the characteristics that predict support for police reform using the entire survey sample, then we can use those predictions – combined with demographic information about Arkansans – to generate better estimates. The trick, in essence, is that our estimate for Arkansas will be borrowing information from similar respondents in other states.

The method proceeds in three steps.

Step 1: Fit a Model

First, we fit a model of our outcome, using observed characteristics of the survey respondents as predictors. To demonstrate, let's fit a simple logistic regression model including only four demographic predictors: gender,

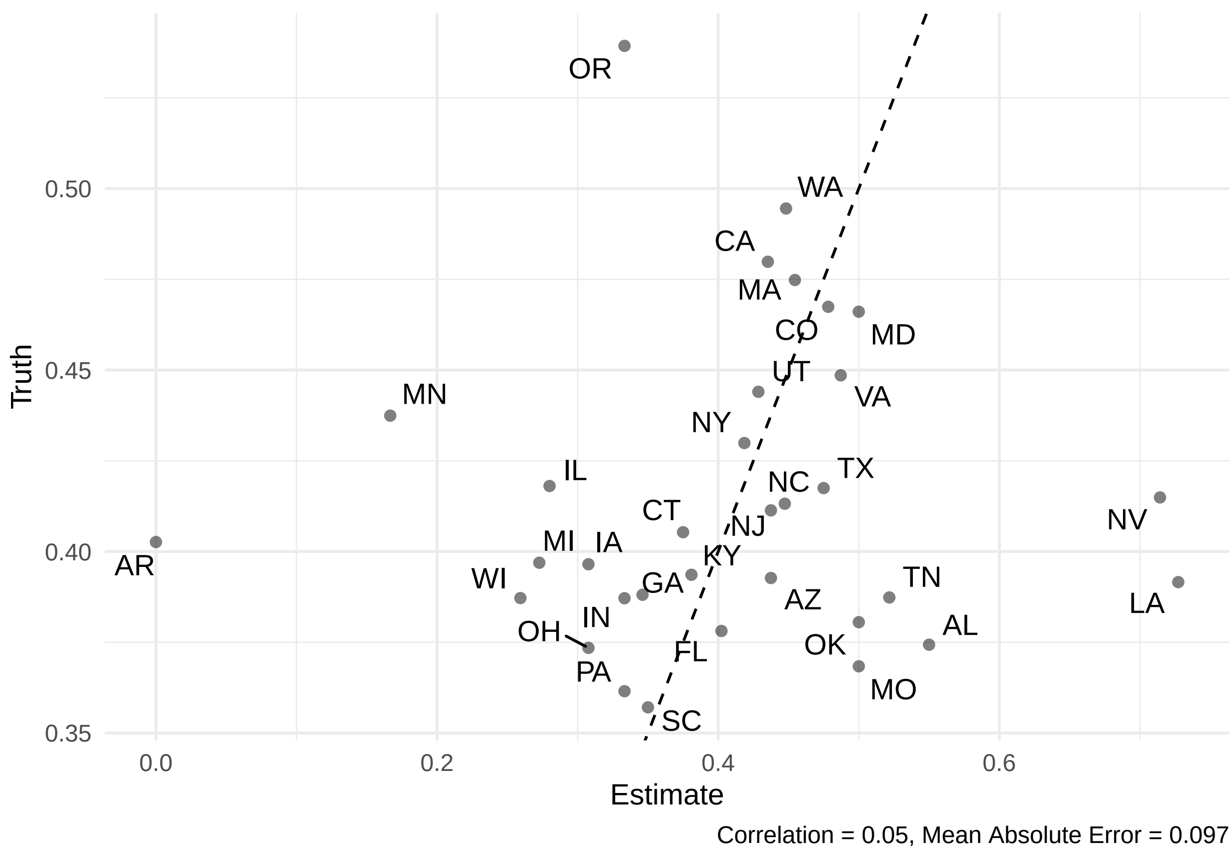

Figure 2: Esimates from disaggregated sample data

education, race, and age.

```
model <- glm(defund_police ~
               gender + educ + race + age,
             data = sample_data,
             family = 'binomial')
```

Step 2: Construct the Poststratification Frame

The poststratification stage requires the researcher to know (or estimate) the joint frequency distribution of predictor variables in each state. This information is stored in a "poststratification frame," a matrix where each row is a unique combination of characteristics, along with the observed frequency of that combination. Often, one constructs this frequency distribution from Census micro-data (Lax and Phillips 2009). For our demonstration, I will compute it directly from the CES.

```
psframe <- ces %>%
  count(abb, gender, educ, race, age)

head(psframe)
```

```
## # A tibble: 6 x 6
##   abb   gender educ   race    age     n
##   <chr> <chr>  <chr>  <chr> <dbl> <int>
## 1 AL    Female 2_year Black    26     1
## 2 AL    Female 2_year Black    27     2
## 3 AL    Female 2_year Black    29     1
## 4 AL    Female 2_year Black    31     1
## 5 AL    Female 2_year Black    34     2
## 6 AL    Female 2_year Black    35     2
```

Step 3: Predict and Poststratify

With the model and poststratification frame in hand, the final step is to generate frequency-weighted predictions of public opinion. For each cell in the poststratification frame, append the model's predicted probability of supporting police defunding.

```
psframe$predicted_probability <- predict(model, psframe, type = 'response')
```

Then the poststratified estimates are the frequency-weighted means of those predictions.

```
poststratified_estimates <- psframe %>%
  group_by(abb) %>%
  summarize(estimate = weighted.mean(predicted_probability, n))
```

Let's see how these estimates compare with the known values (Figure 3).

```
compare_to_truth(poststratified_estimates, truth)
```

These estimates, though still imperfectly correlated with the truth, are much better than the previous estimates from disaggregation. Notice, in particular, that the estimate for Arkansas went from 0% to roughly 39%, reflecting the significant improvement that comes from using more information than the four Arkansans in our sample can provide.

But we can still do better. In the following sections, I will show how successive improvements to the first-stage model can yield more reliable poststratified estimates.

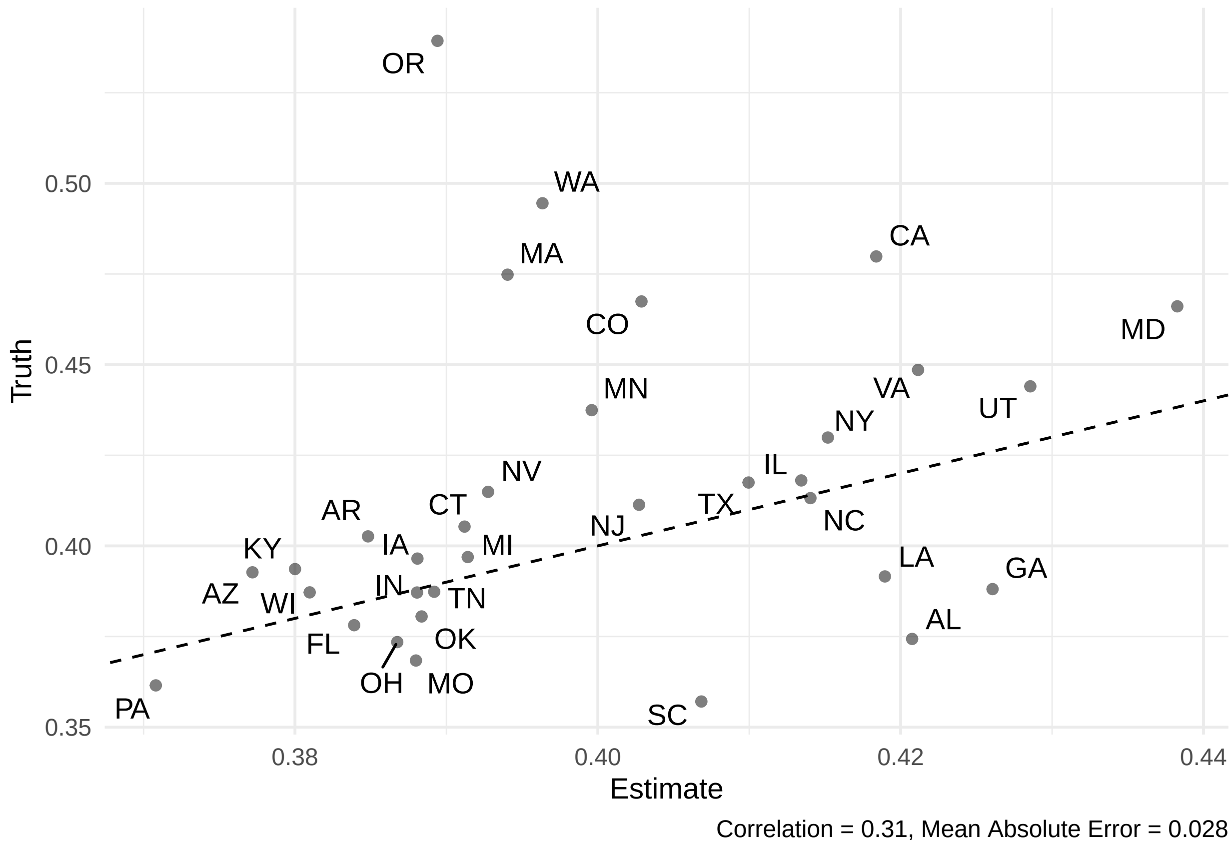

Figure 3: Underfit MRP estimates from complete pooling model

Beware Overfitting

A common instinct among social scientists building models is to take a "kitchen sink" approach, including as many explanatory variables as possible (Achen 2005). This is counterproductive when the objective is out-of-sample predictive accuracy. To illustrate, let's estimate a model with a separate intercept term for each state – a "fixed effects" model. Because our sample contains several states with very few observations, these state-specific intercepts will be over-fit to sampling variability.

```r
# fit the model
model2 <- glm(defund_police ~
              gender + educ + race + age +
                abb,
            data = sample_data,
            family = 'binomial')

# construct the poststratification frame
psframe <- ces %>%
  count(abb, gender, educ, race, age)

# make predictions
psframe$predicted_probability <- predict(model2, psframe, type = 'response')

# poststratify
poststratified_estimates <- psframe %>%
  group_by(abb) %>%
  summarize(estimate = weighted.mean(predicted_probability, n))


compare_to_truth(poststratified_estimates, truth)
```

These poststratified estimates perform about as well as the disaggregated estimates from Figure 2. Because each state's intercept is estimated separately, the over-fit model foregoes the advantages of "partial pooling" (Park, Gelman, and Bafumi 2004), borrowing information from respondents in other states. Note that the estimate for Arkansas is once again 0%.

Partial Pooling

A better approach is to estimate a multilevel model (alternatively known as "varying-intercepts" or "random effects" model), including group-level covariates. In the model below, I estimate varying intercepts by US Census division, including the state's 2020 Democratic vote share as a covariate. The result is a marked improvement over Figure 3 (particularly for west coast states like Oregon, Washington, and California).

```r
library(lme4)

# fit the model
model3 <- glmer(defund_police ~  gender + educ + race + age +
                (1 + biden_vote_share | division),
                 data = sample_data,
                 family = 'binomial')

# construct the poststratification frame
psframe <- ces %>%
  count(abb, gender, educ, race, age, division, biden_vote_share)

# make predictions
psframe$predicted_probability <- predict(model3, psframe, type = 'response')
```

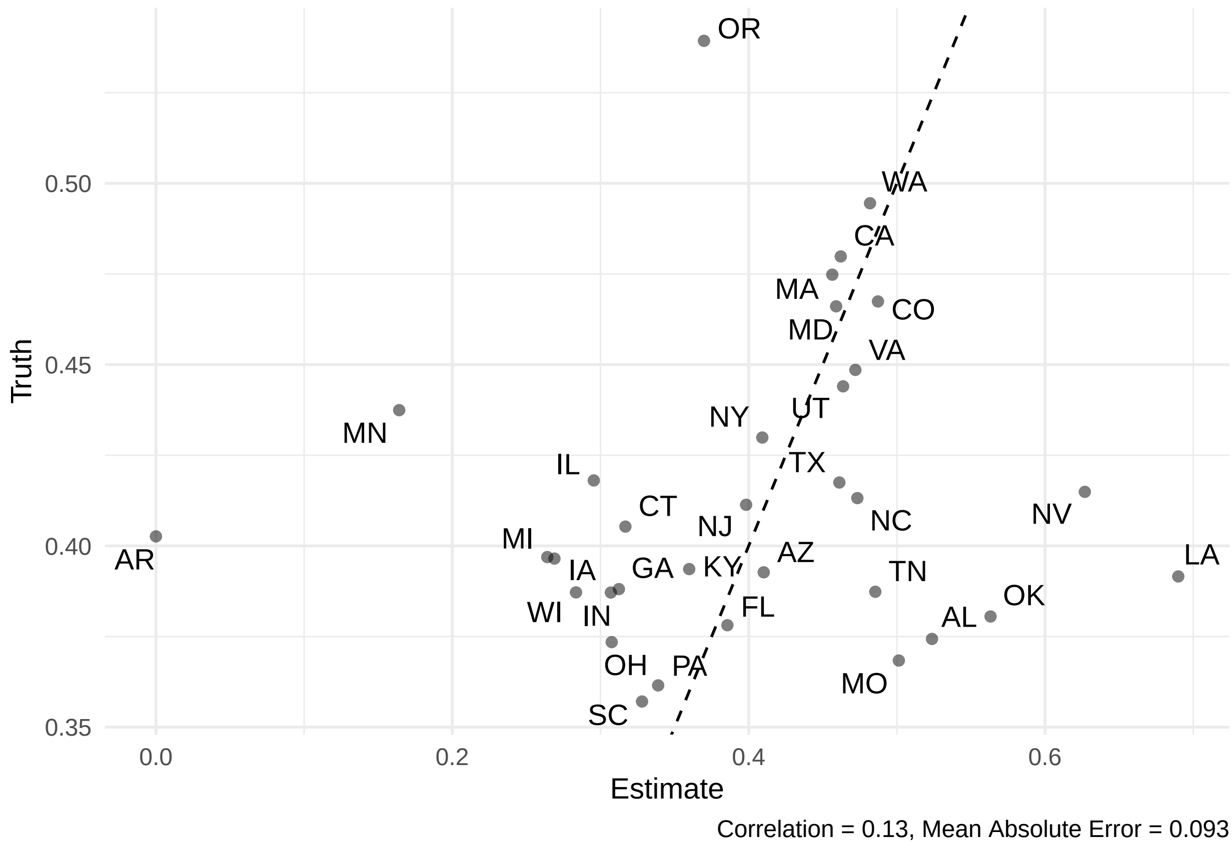

Figure 4: Overfit MRP estimates from fixed effects model

```r
# poststratify
poststratified_estimates <- psframe %>%
  group_by(abb) %>%
  summarize(estimate = weighted.mean(predicted_probability, n))

compare_to_truth(poststratified_estimates, truth)
```

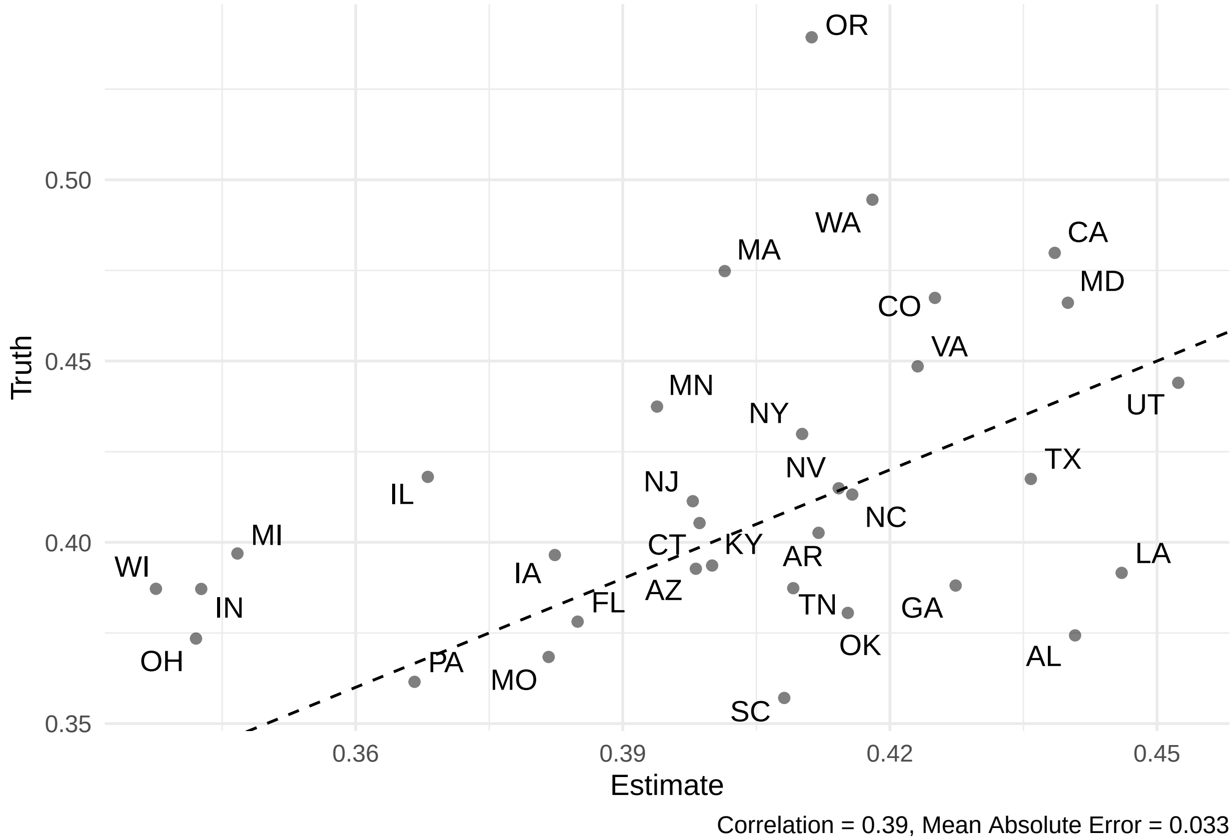

Figure 5: MRP estimates from model with partial pooling

Sample Size Is Critical

MRP's performance depends heavily on the quality and size of the researcher's survey sample. Up to now, we've been working with a random sample of 1,000 respondents, and though the resulting estimates are better than the raw sample means, their performance has been somewhat underwhelming. Suppose instead we had a sample of 5,000 respondents.

```r
sample_data <- ces %>%
  slice_sample(n = 5000)

# fit the model
model3 <- glmer(defund_police ~  gender + educ + race + age +
                (1 + biden_vote_share | division),
                 data = sample_data,
                 family = 'binomial')
```

```r
# construct the poststratification frame
psframe <- ces %>%
  count(abb, gender, educ, race, age, division, biden_vote_share)

# make predictions
psframe$predicted_probability <- predict(model3, psframe, type = 'response')

# poststratify
poststratified_estimates <- psframe %>%
  group_by(abb) %>%
  summarize(estimate = weighted.mean(predicted_probability, n))

compare_to_truth(poststratified_estimates, truth)
```

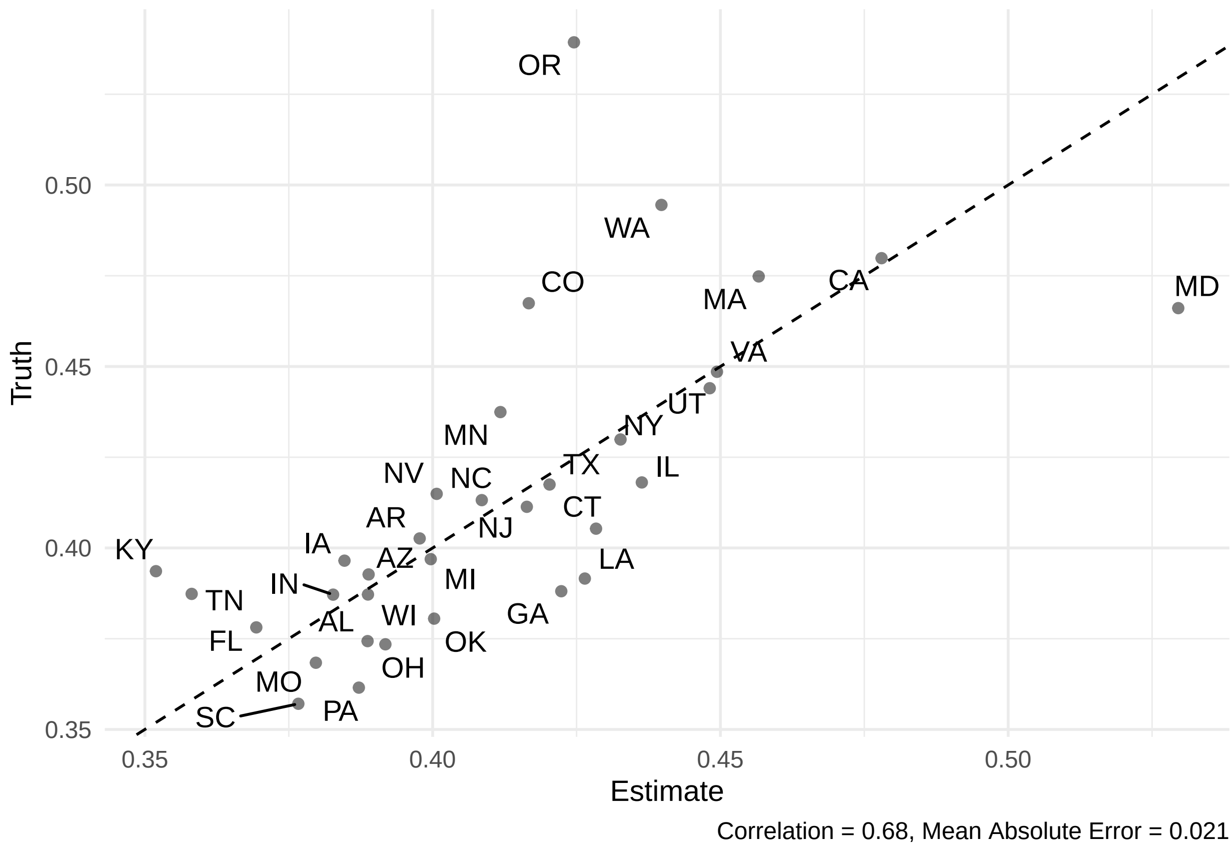

Figure 6: Poststratified estimates with a survey sample of 5,000

Now MRP really shines. With more observations, the first-stage model can better predict opinions of out-of-sample respondents, which dramatically improves the poststratified estimates.

Stacked Regression and Poststratification (SRP)

Ultimately, the accuracy of one's poststratified estimates depends on the out-of-sample predictive performance of the first-stage model. As we've seen above, the challenge is to thread the needle between over-fitting and under-fitting. Several recent papers (Bisbee 2019; Ornstein 2020; Broniecki, Leemann, and Wüest 2022) have shown that approaches from machine learning can help to automate this process, particularly with large survey samples.

In the code below, I'll demonstrate how an *ensemble* of models – using the same set of predictors but different methods for combining them into predictions – can yield superior performance to a single multilevel regression model. In particular, I will fit a "stacked regression" (Breiman 1996), which makes predictions based on a weighted average of multiple models, where the weights are assigned by cross-validated prediction performance (van der Laan, Polley, and Hubbard 2007) . The literature on ensemble models is extensive, but for good entry points I recommend Breiman (1996), Breiman (2001), and Montgomery, Hollenbach, and Ward (2012).

```
# construct the poststratification frame
psframe <- ces %>%
  count(abb, gender, educ, race, age, division, biden_vote_share)

# fit the model (an ensemble of random forest and logistic regression)
library(SuperLearner)

SL.library <- c("SL.ranger", "SL.glm")

X <- sample_data %>%
  select(gender, educ, race, age, division, biden_vote_share)

newX <- psframe %>%
  select(gender, educ, race, age, division, biden_vote_share)

sl <- SuperLearner(Y = sample_data$defund_police,
                   X = X,
                   newX = newX,
                   family = binomial(),
                   SL.library = SL.library, verbose = FALSE)
```

```
## Error : loading required package (ranger) failed
## Error : loading required package (ranger) failed
## Error : loading required package (ranger) failed
## Error : loading required package (ranger) failed
## Error : loading required package (ranger) failed
## Error : loading required package (ranger) failed
## Error : loading required package (ranger) failed
## Error : loading required package (ranger) failed
## Error : loading required package (ranger) failed
## Error : loading required package (ranger) failed
## Error : loading required package (ranger) failed
```

```
# make predictions
psframe$predicted_probability <- sl$SL.predict

# poststratify
poststratified_estimates <- psframe %>%
  group_by(abb) %>%
  summarize(estimate = weighted.mean(predicted_probability, n))

compare_to_truth(poststratified_estimates, truth)
```

The performance gains in Figure 7 reflect the improvement that comes from modeling "deep interactions" in the predictors of public opinion (Ghitza and Gelman 2013). If, for example, income better predicts partisanship in some states but not in others (Gelman et al. 2007), then a model that captures that moderating effect will produce better poststratified estimates than one that does not. Machine learning techniques like random forest (Breiman 2001) are especially useful for automatically detecting and representing such deep interactions, and stacked regression and poststratification (SRP) tends to outperform MRP in simulations, particularly for

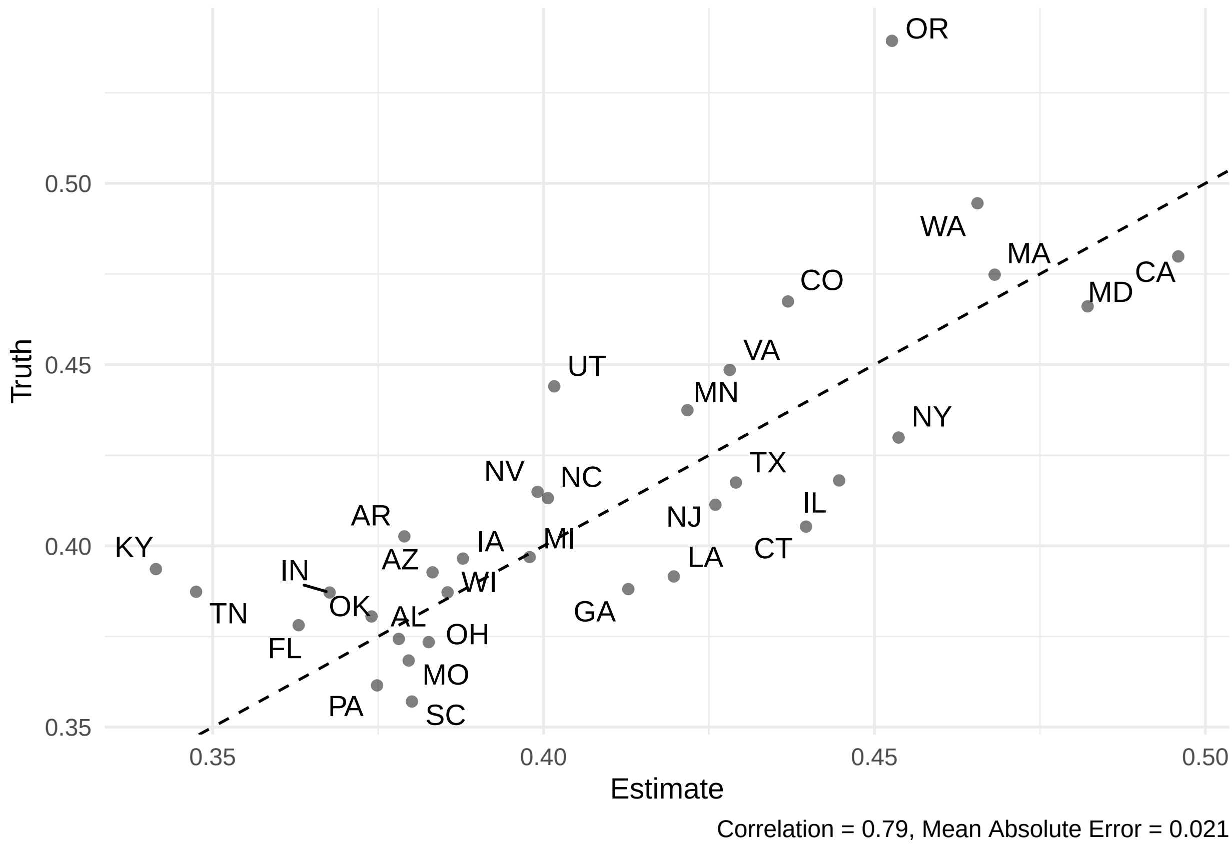

Figure 7: Estimates from an ensemble first-stage model

training data with large sample size (Ornstein 2020).

Synthetic Poststratification

Researchers rarely have access to the entire joint distribution of individual-level covariates. This can be limiting, since there may be variables that one would like to include in the first-stage model, but cannot because is not in the poststratification frame. Leemann and Wasserfallen (2017) suggest an extension of MRP, which they (delightfully) dub Multilevel Regression and Synthetic Poststratification (MrsP). Lacking the full joint distribution of covariates for poststratification, one can instead create a *synthetic* poststratification frame by assuming that additional covariates are statistically independent of one another. So long as the first-stage model is linear-additive, this approach yields the same predictions as if you knew the true joint distribution![3] And even if the first-stage model is not linear-additive, simulations suggest that the improved performance from additional predictors tends to overcome the error introduced in the poststratification stage.

Here are some CES covariates that we might want to include in our model of police reform:

- How important is religion to the respondent?
- Whether the respondent lives in an urban, rural, or suburban area
- Whether the respondent or a member of the respondent's family is a military veteran
- Whether the respondent owns or rents their home
- Is the respondent the parent or guardian of a child under the age of 18?

These variables are likely to be useful predictors of opinion about police reform, and the first-stage model could be improved by including them. But there is no dataset (that I know of) that would allow us to compute a state-level joint probability distribution over every one of them. Instead, we would typically only know the marginal distributions of each covariate (e.g. the percent of a state's residents that are military households, or the percent that live in urban areas). So a synthetic poststratification approach may prove helpful.

To create a synthetic poststratification frame, we create a set of marginal probability distributions, and multiply them together.[4]

```
# fit the model
model4 <- glmer(defund_police ~  gender + educ + race + age +
                  pew_religimp + homeowner + urban +
                  parent + military_household +
                  (1 + biden_vote_share | division),
                data = sample_data,
                family = 'binomial')

# construct the poststratification frame
psframe <- ces %>%
  count(abb, gender, educ, race, age,
        division, biden_vote_share) %>%
  # convert frequencies to probabilities
  group_by(abb) %>%
  mutate(prob = n/sum(n))

# find the marginal distribution for each new variable
marginal_pew_religimp <- ces %>%
  count(abb, pew_religimp) %>%
  group_by(abb) %>%
  mutate(marginal_pew_religimp = n/sum(n))
```

[3] See Ornstein (2020) appendix A for mathematical proof.

[4] The `SRP` package contains a convenience function for this operation (see the vignette for more information).

```r
marginal_homeowner <- ces %>%
  count(abb, homeowner) %>%
  group_by(abb) %>%
  mutate(marginal_homeowner = n/sum(n))

marginal_urban <- ces %>%
  count(abb, urban) %>%
  group_by(abb) %>%
  mutate(marginal_urban = n/sum(n))

marginal_parent <- ces %>%
  count(abb, parent) %>%
  group_by(abb) %>%
  mutate(marginal_parent = n/sum(n))

marginal_military_household <- ces %>%
  count(abb, military_household) %>%
  group_by(abb) %>%
  mutate(marginal_military_household = n/sum(n))

# merge the marginal distributions together
synthetic_psframe <- psframe %>%
  left_join(marginal_pew_religimp, by = 'abb') %>%
  left_join(marginal_homeowner, by = 'abb') %>%
  left_join(marginal_urban, by = 'abb') %>%
  left_join(marginal_parent, by = 'abb') %>%
  left_join(marginal_military_household, by = 'abb') %>%
  # and multiply
  mutate(prob = prob * marginal_pew_religimp *
           marginal_homeowner * marginal_urban *
           marginal_parent * marginal_military_household)
```

Then poststratify as normal using the synthetic poststratification frame.

```r
# make predictions
synthetic_psframe$predicted_probability <- predict(model4, synthetic_psframe,
                                                   type = 'response')

# poststratify
poststratified_estimates <- synthetic_psframe %>%
  group_by(abb) %>%
  # (note that we're weighting by prob instead of n here)
  summarize(estimate = weighted.mean(predicted_probability, prob))

compare_to_truth(poststratified_estimates, truth)
```

Best Performing

As a final demonstration, suppose we had access to the entire joint distribution over those covariates, *and* our first stage model was a Super Learner ensemble. This combination yields the best-performing estimates yet (Figure 9).

```r
# construct the poststratification frame
psframe <- ces %>%
  count(abb, gender, race, age, educ,
```

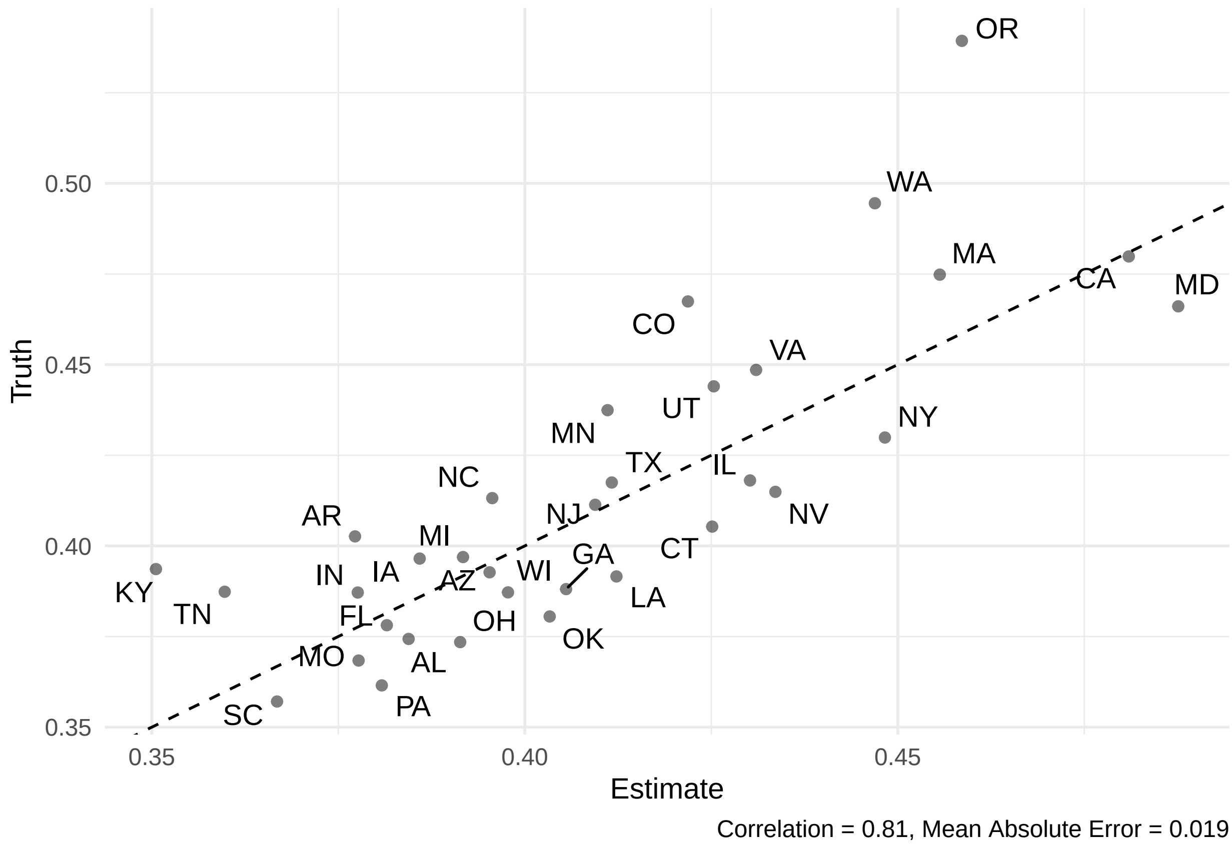

Figure 8: Estimates from synthetic poststratification, including additional covariates

```r
        division, biden_vote_share,
        pew_religimp, homeowner, urban,
        parent, military_household)

# fit Super Learner
SL.library <- c("SL.ranger", "SL.glm")

X <- sample_data %>%
  select(gender, race, age, educ,
        division, biden_vote_share,
        pew_religimp, homeowner, urban,
        parent, military_household)

newX <- psframe %>%
  select(gender, race, age, educ,
        division, biden_vote_share,
        pew_religimp, homeowner, urban,
        parent, military_household)

sl <- SuperLearner(Y = sample_data$defund_police,
                   X = X,
                   newX = newX,
                   family = binomial(),
                   SL.library = SL.library,
                   verbose = FALSE)
```

```
## Error : loading required package (ranger) failed
## Error : loading required package (ranger) failed
## Error : loading required package (ranger) failed
## Error : loading required package (ranger) failed
## Error : loading required package (ranger) failed
## Error : loading required package (ranger) failed
## Error : loading required package (ranger) failed
## Error : loading required package (ranger) failed
## Error : loading required package (ranger) failed
## Error : loading required package (ranger) failed
## Error : loading required package (ranger) failed
```

```r
# make predictions
psframe$predicted_probability <- sl$SL.predict

# poststratify
poststratified_estimates <- psframe %>%
  group_by(abb) %>%
  summarize(estimate = weighted.mean(predicted_probability, n))

compare_to_truth(poststratified_estimates, truth)
```

The results shown in Figure 9 reflect all the gains from a larger sample size, ensemble modeling, and a full set of individual-level and group-level predictors.

Conclusion

The demonstrations in this chapter suggest a number of lessons for researchers using MRP.

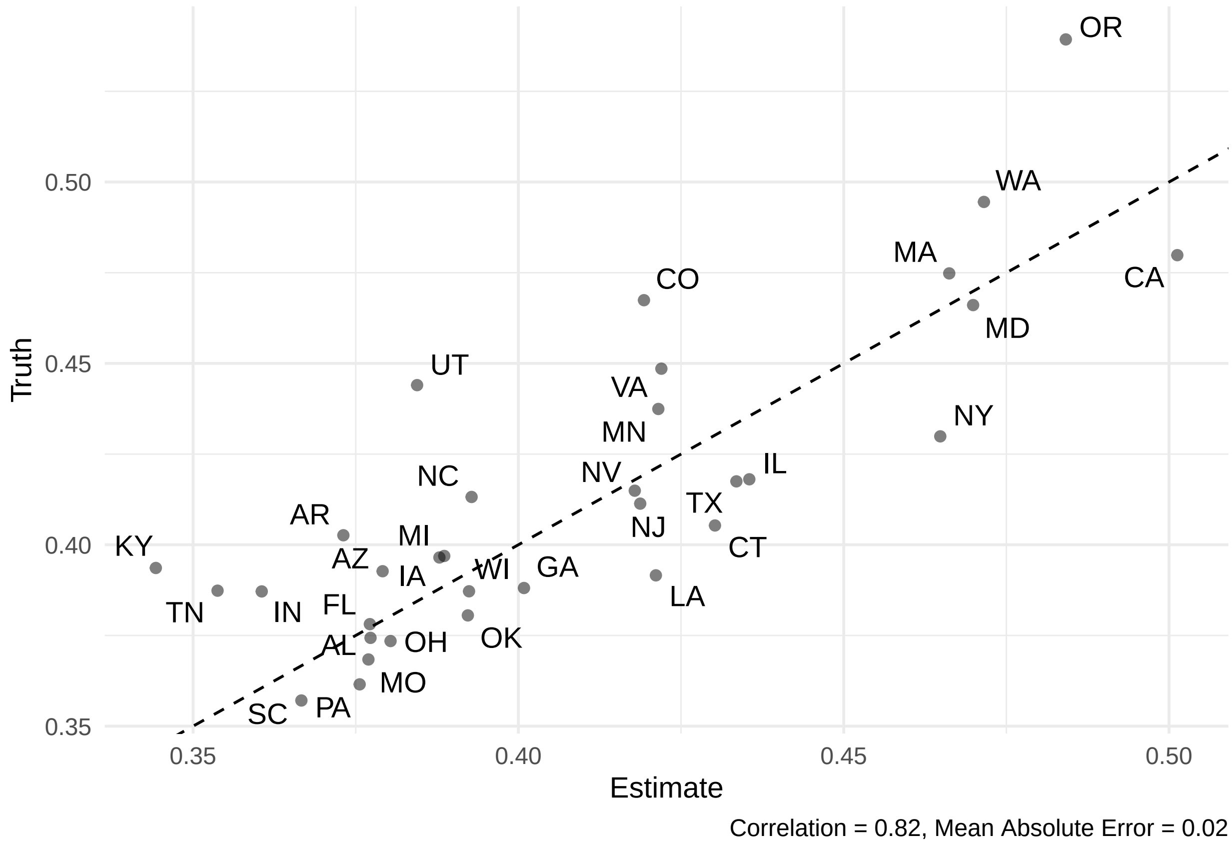

Figure 9: The best performing estimates, using a large survey sample, ensemble first stage model, and full set of predictors.

First, be wary of first-stage models that are under- or over-fit to the survey data. As we saw in Figure 3, MRP estimates with too few modeled predictors tend to overshrink towards the grand mean.[5] Using such estimates in a subsequent analysis would understate the differences between regions. Conversely, models that are over-fit to survey data (e.g. Figure 4) will tend to exaggerate regional differences.

Second, new techniques like synthetic poststratification and stacked regression can help researchers manage the tradeoff between under-fitting and over-fitting. Synthetic poststratification allows for the inclusion of more relevant predictors, and regularized ensemble models help ensure that the predictions are not over-fit to noisy survey samples. The best estimates come from combining these two approaches.

Finally, recall that the most significant performance gains in our demonstration came not from more sophisticated modeling techniques, but from more data. As we saw in Figure 6, working with a larger survey yielded greater marginal improvements than any tinkering around with the first-stage modeling choices. MRP is not a panacea, and one should be skeptical of estimates produced from small-sample surveys, regardless of how they are operationalized.

In the code above I emphasize "do-it-yourself" approaches to MRP – fitting a model, building a poststratification frame, and producing estimates separately. But there are a number of `R` packages available with useful functions to help ease the process. In particular, I would encourage curious readers to explore the `autoMrP` package (Broniecki, Leemann, and Wüest 2022), which implements the ensemble modeling approach described above, and performs quite well in simulations when compared to existing packages.

Further Suggested Readings

- McElreath, Richard. 2020. *Statistical Rethinking: A Bayesian Course with Examples in R and Stan.* 2nd ed. Boca Raton: Taylor and Francis, CRC Press. (particularly chapter 13).
- Gelman, Andrew, Jennifer Hill, and Aki Vehtari. 2021. *Regression and Other Stories.* Cambridge, United Kingdom: Cambridge University Press. (particularly chapter 17).

Review Questions

1.

References

Achen, Christopher H. 2005. "Let's Put Garbage-Can Regressions and Garbage-Can Probits Where They Belong." *Conflict Management and Peace Science* 22 (4): 327–39. https://doi.org/10.1080/07388940500339167.

Ansolabehere, Stephen, Samantha Luks, and Brian F. Schaffner. 2015. "The Perils of Cherry Picking Low Frequency Events in Large Sample Surveys." *Electoral Studies* 40 (December): 409–10. https://doi.org/10.1016/j.electstud.2015.07.002.

Bisbee, James. 2019. "BARP: Improving Mister P Using Bayesian Additive Regression Trees." *American Political Science Review* 113 (4): 1060–65. https://doi.org/10.1017/S0003055419000480.

Breiman, Leo. 1996. "Stacked Regressions." *Machine Learning* 24: 49–64. https://doi.org/10.17485/ijst/2016/v9i28/98380.

———. 2001. "Random Forests." *Machine Learning* 45 (1): 5–32. https://doi.org/10.1023/A:1010933404324.

Broniecki, Philipp, Lucas Leemann, and Reto Wüest. 2022. "Improved Multilevel Regression with Poststratification Through Machine Learning (autoMrP)." *The Journal of Politics* 84 (1). https://doi.org/10.1086/714777.

Buttice, Matthew K., and Benjamin Highton. 2013. "How Does Multilevel Regression and Poststratification Perform with Conventional National Surveys?" *Political Analysis* 21 (4): 449–67. https://doi.org/10.1093/pan/mpt017.

[5] In the limit, a first-stage model with zero predictors would yield identical poststratified estimates for each state, equal to the survey sample mean.

Gelman, Andrew. 2018. "Regularized Prediction and Poststratification (the Generalization of Mister p)." *Statistical Modeling, Causal Inference, and Social Science (Blog)* May 19 (https://statmodeling.stat.columbia.edu/2018/05/19/).

Gelman, Andrew, and Thomas C Little. 1997. "Poststratification into Many Categories Using Hierachical Logistic Regression." *Survey Methodology* 23 (2): 127–35.

Gelman, Andrew, Boris Shor, Joseph Bafumi, and David Park. 2007. "Rich State, Poor State, Red State, Blue State: What's the Matter with Connecticut?" *Quarterly Journal of Political Science* 2 (June 2006): 345–67. https://doi.org/10.1561/100.00006026.

Ghitza, Yair, and Andrew Gelman. 2013. "Deep Interactions with MRP: Election Turnout and Voting Patterns Among Small Electoral Subgroups." *American Journal of Political Science* 57 (3): 762–76. https://doi.org/10.1111/ajps.12004.

Lax, Jeffrey R., and Justin H. Phillips. 2009. "How Should We Estimate Public Opinion in the States?" *American Journal of Political Science* 53 (1): 107–21. https://doi.org/10.1111/j.1540-5907.2008.00360.x.

———. 2012. "The Democratic Deficit in the States." *American Journal of Political Science* 56 (1): 148–66. https://doi.org/10.1111/j.1540-5907.2011.

Leemann, Lucas, and Fabio Wasserfallen. 2017. "Extending the Use and Prediction Precision of Subnational Public Opinion Estimation." *American Journal of Political Science* 61 (4): 1003–22.

Montgomery, Jacob M, Florian Hollenbach, and Michael D Ward. 2012. "Improving Predictions Using Ensemble Bayesian Model Averaging." *Political Analysis* 20 (3): 271–91.

Ornstein, Joseph T. 2020. "Stacked Regression and Poststratification." *Political Analysis* 28 (2): 293–301. https://doi.org/10.1017/pan.2019.43.

Park, David K., Andrew Gelman, and Joseph Bafumi. 2004. "Bayesian Multilevel Estimation with Poststratification: State-Level Estimates from National Polls." *Political Analysis* 12 (4): 375–85. https://doi.org/10.1093/pan/mph024.

Schaffner, Brian, Stephen Ansolabehere, and Sam Luks. 2021. "Cooperative Election Study Common Content, 2020." Edited by YouGov and Add your team name(s) here. https://doi.org/10.7910/DVN/E9N6PH.

Tausanovitch, Chris, and Christopher Warshaw. 2014. "Representation in Municipal Government." *The American Political Science Review* 108 (03): 605–41. https://doi.org/10.1017/S0003055414000318.

van der Laan, Mark J., Eric. C. Polley, and Alan E. Hubbard. 2007. "Super Learner." *Statistical Applications in Genetics and Molecular Biology* 6 (1).

Wang, Wei, David Rothschild, Sharad Goel, and Andrew Gelman. 2015. "Forecasting Elections with Non-Representative Polls." *International Journal of Forecasting* 31 (3): 980–91. https://doi.org/10.1016/j.ijforecast.2014.06.001.