Week 5: Surveys and Weighting

PLSC 40502 - Statistical Models

Review

Previously

- **Stan**
  - **Hamiltonian Monte Carlo** - use the gradient of the likelihood $\times$ prior to get good M-H proposals
  - Avoids proposals that are either too auto-correlated and too likely to be rejected
  - Commmon structure to statistical modeling: `data`, `parameters`, `model`
  - Frequently used models have canned routines in `rstanarm`
- **Diagnosing model fit**
  - Posterior predictive checks
  - **Information criteria** vs. **Cross-validation**
  - log-predictive density as a measure of prediction quality (how much probability mass do we place on the "correct" outcome)

This week

- **Survey weighting and post-stratification**
    - The problem of non-response
    - Using auxiliary variables to construct weights
    - Post-stratification vs. raking
- **Calibration weighting**
    - Generalizing post-stratification-style adjustments to allow for a variety of balance conditions
- **Combining multilevel regression and post-stratification**
    - Using multi-level models and population-level information to estimate quantities for **small areas**

Survey sampling

Surveys

- A common task in survey research is estimating an unknown population parameter θ from a sample of respondents $i \in \{1, 2, 3, \ldots, N\}$
  - What share of voters in Michigan plan to vote for Joe Biden in the 2020 election?
  - What is the share of adults who are vaccinated in each U.S. county?
  - What share of residents in Nevada plan to vote in the 2024 general election?
- Historically, two approaches to survey sampling design
  - **Quota sampling** - Define a set of known demographic targets and recruit respondents to match.
  - **Probability sampling** - Select respondents from a **sampling frame** at random with a known probability.
- Dominance of **probability sampling** in the late 20th century
  - Random Digit Dialing allowed for (near)-simple random samples from the U.S. adult population
  - High response rates

Decline of pure probability samples

- Two big factors have lead to the decline of exclusively probability-based sampling approaches
  - Decline in population coverage - fewer individuals have landlines!
  - Extremely high non-response rates.
- Non-random non-response can bias our estimates
  - Non-responders have different characteristics to the responders that may be correlated with the target quantity of interest.
  - Huge concern in recent efforts to poll elections (e.g. "shy tories")
- Modern polling
  - Combine probability and quota approaches
  - Weighting ex-post to match population targets.

Survey inference

- Our goal is to estimate some parameter θ_y related to a population outcome variable y (e.g. a proportion, mean, median, etc...)
- We construct an estimator $\hat{\theta}_y$
  - For population means, we'll use a sample mean, etc...
- In addition to the outcome of interest y, we observe **auxiliary** variables $\mathbf{x}$
  - Units are sampled from some unknown target population $f_{\mathcal{U}}(y, \mathbf{x})$
  - The in-sample distribution of the **responders** is denoted $g_{\mathcal{R}}(y, \mathbf{x})$
- Our auxiliary information involves some **known features** of the target distribution: $\breve{I}_{\mathbf{x}}$
  - We'll use this to construct **targets** for the population distribution $\tilde{T}_{\mathbf{x}} = \{\tilde{T}_{\mathbf{x}1}, \ldots, \tilde{T}_{\mathbf{x}M}\}$
  - For example, suppose we know the full population distribution of age, gender, education, income, and party ID
- In some settings, the auxiliary distribution maps easily to the target - but in other settings we have to use the auxiliary information to **estimate** our target
  - (e.g) we might have features of our target distribution but we don't know who will turn out to **vote** before the election
  - "Likely voter models" are a **target estimation** problem

Inference with known sampling weights

- If we know the probability that a unit is selected into the sample from the sample frame π_i, it is straightforward to construct an estimator $\hat{\theta}_y$ of the population mean θ_y.
- The **Horvitz-Thompson** estimator weights each unit by $d_i = \frac{1}{\pi_i}$, the inverse probability of being selected into the sample

$$\hat{\theta}_y^{\text{HT}} = \frac{\sum_{i=1}^{N} d_i Y_i}{\mathbb{E}[\sum_{i=1}^{n} d_i]}$$

- When sampling probabilities are equivalent, this reduces to the sample mean.
- More commonly, rather than using the expectation of the weights in the denominator, we'll use the actual observed sum of the weights, giving the Hajek estimator

$$\hat{\theta}_y^{\text{H}} = \frac{\sum_{i=1}^{N} d_i Y_i}{\sum_{i=1}^{n} d_i}$$

Unknown sampling weights

- When d_i is not known, we will need to estimate **adjustment weights** using a combination of modeling assumptions and auxiliary data
    - Even when d_i is known, if non-response is high, we still don't know the probability of selection into the **observed** data ρ_i
- With adjustment weights $\tilde{w}_i$, our Hajek estimator becomes

$$\hat{\theta}_y^{\mathrm{W}} = \frac{\sum_{i=1}^{N} \tilde{w}_i Y_i}{\sum_{i=1}^{n} \tilde{w}_i}$$

- Now the weights are not necessarily known but must be obtained from our population targets $\tilde{T}_{\mathbf{x}}$

Post-stratification

- The easiest approach to adjusting a non-representative survey is to weight to match the **known joint** distribution of x in the population $f_{\mathcal{U}}(\mathbf{x})$
    - This requires a lot of auxiliary information about the target $f_{\mathcal{U}}(\mathbf{x})$ - typically obtained from high-quality census data
    - (e.g.) U.S. Census Public-Use Microdata: What is the share of Black, college educated, 30-45 year olds in Massachusetts?
- In post-stratification, we divide our sample up into C **cells** c that are mutually exclusive and exhaustive.
    - $\tilde{T}_{\mathbf{x}} = \{\tilde{P}_1, \tilde{P}_2, \dots, \tilde{P}_C\}$ is our population distribution of these cells
- Our **sampling/response model** assumes that **within** each of these cells we have a simple-random sample from that particular stratum of the population
    - There may be variation in non-response or over/under-sampling, but only **across** cells
    - "post"-stratification because the intuition is akin to a design where we actually *did* stratify ex-ante
- Our **measurement model** assumes we observe the joint distribution of auxiliary variables $\mathbf{x}$ in the target population
    - Difficult in many cases!

Post-stratification

- Two ways to think of post-stratification:
- **First** - Let $\hat{\theta}_y^c$ denote our estimator for the population mean **within** cell c (possibly using design weights d_i)
  - Then, our post-stratification estimator $\hat{\theta}_y^{PS}$ is:

$$\hat{\theta}_y^{PS} = \sum_{c=1}^{C} \tilde{P}_c \hat{\theta}_y^c$$

- Alternatively, we'll sometimes write $\tilde{P}_c = \frac{\tilde{N}_c}{\tilde{N}}$ where $\tilde{N}$ is the size of the population and $\tilde{N}_c$ is the number of units in that cell.

$$\hat{\theta}_y^{PS} = \sum_{c=1}^{C} \frac{\tilde{N}_c}{\tilde{N}} \hat{\theta}_y^c$$

- Fixed "constant" weights on the within-cell estimators.

Post-stratification

- **Second** - We can think of it as weighting individual observations using our adjustment weights $\tilde{w}_i^{PS}$
- Let $c(i)$ denote the class to which unit i belongs. Then the post-stratification weights are

$$\tilde{w}_i^{PS} = (\tilde{P}_{c(i)} / \hat{P}^S_{c(i)}) \times d_i$$

where $\hat{P}^S_{c(i)}$ is the estimated **in-sample** proportion of observations in class c

- When design weights are constant, $\hat{P}^S_{c(i)}$ is just the sample mean of the indicator of class membership $\mathbf{1}_{i \in c}$
  - More generally, you can write it as $\hat{P}^S_{c(i)} = (\sum_{i=1}^N d_i \mathbf{1}_{i \in c}) / (\sum_{i=1}^N d_i)$
- **Intuition**
  - The weights **up-weight** observations that are under-represented in the sample relative to the target population
  - The weights **down-weight** observations that are over-represented in the sample relative to the target population.

Raking

- Often post-stratification with many covariates is challenging!
  - The number of cells grows rapidly as we add more covariates
  - e.g. gender x party x state = 2 x 3 x 50 = 300 cells!
- Sometimes our population data only give us the marginal distributions but not the joint distributions
- **Raking** weights are designed to match the marginal distribution of the auxiliary covariate **in-sample**
  - No closed-form expression, but iterative algorithms exist to compute raking weights.
  - **Intuition**
  - Raking works well when things are additive
  - Doesn't work as well when things are **interactive**

Example: CCES 2020

- Let's dive in to the 2020 CCES.

```
library(survey)
cces <- read_csv("data/CCES_subset.csv") %>% filter(!is.na(trumpApprove))
```

- We'll be using a subset of the outcome data, looking at the share of respondents who state that they strongly or somewhat approve of then-President Donald Trump.
- Start by making a `svydesign` object - we'll pretend that the sampling weights don't exist for now

```
cces_surv_unwt <- svydesign(~1, weights = ~1, data=cces)
```

- In-sample, what's the proportion of respondents who approve of Trump?

```
svymean(~trumpApprove, design=cces_surv_unwt)
```

```
##               mean SE
## trumpApprove 0.383  0
```

Example: CCES 2020

- How does this compare to the properly weighted mean?

```
cces_surv_wt <- svydesign(~1, weights = ~commonweight, data=cces)
svymean(~trumpApprove, design=cces_surv_wt)
```

```
##               mean SE
## trumpApprove 0.444  0
```

- Trump's approval is a few points higher after the weighting adjustment.

- From the CCES Guide:

> the completed cases were weighted to the sampling frame using entropy balancing. The 2019 ACS was used as the frame for weighting the common content and the team samples. The CES sample was weighted to match the distributions of the 2019 ACS on gender, age, race, Hispanic origin, and education level. The moment conditions included age, gender, education, race, plus their interactions. The resultant weights were then post-stratified by age, gender, education, race, "born again" status, voter registration status, 2016 Presidential vote choice, and 2020 Presidential vote choice as needed.

Population targets

- We'll be using the 2020 ACS 5-year Public Use Microdata Sample
  - Obtain the complete **joint** distribution of region, race, age, gender and education in the U.S.

```
acs_targets <- read_csv("data/ACS_2020_microdata_3cat.csv")
```

- What's the **marginal** distribution of education in the target population?

```
acs_targets %>% group_by(educ_bin) %>% summarize(n = sum(Count)) %>% ungroup() %>% mutate(prop
```

```
## # A tibble: 4 × 3
##   educ_bin                 n  prop
##   <chr>                <dbl> <dbl>
## 1 College graduate 48194135 0.190
## 2 H.S. or less     99042042 0.391
## 3 Postgraduate     28425804 0.112
## 4 Some college     77634550 0.306
```

Population targets

- How does it compare to the **unweighted** marginal distribution in the sample?

```
cces %>% group_by(educ_bin) %>% summarize(n = n()) %>% ungroup() %>% mutate(prop = n/sum(n))
```

```
## # A tibble: 4 × 3
##   educ_bin             n  prop
##   <chr>            <int> <dbl>
## 1 College graduate 14146 0.232
## 2 H.S. or less     18592 0.305
## 3 Postgraduate      8373 0.137
## 4 Some college     19857 0.326
```

Population targets

- What about the **joint** distribution of gender and education?
- In the **population**

```
acs_targets %>% group_by(educ_bin, gender_bin) %>% summarize(n = sum(Count)) %>% ungroup() %>%
```

```
## # A tibble: 8 × 4
##   educ_bin         gender_bin        n   prop
##   <chr>            <chr>         <dbl>  <dbl>
## 1 College graduate Female     25521371 0.101 
## 2 College graduate Male       22672764 0.0895
## 3 H.S. or less     Female     48061006 0.190 
## 4 H.S. or less     Male       50981036 0.201 
## 5 Postgraduate     Female     15043514 0.0594
## 6 Postgraduate     Male       13382290 0.0528
## 7 Some college     Female     41305976 0.163 
## 8 Some college     Male       36328574 0.143
```

Population targets

- In the **sample**

```
cces %>% group_by(educ_bin, gender_bin) %>% summarize(n = n()) %>% ungroup() %>% mutate(prop =
```

```
## # A tibble: 8 × 4
##   educ_bin          gender_bin     n   prop
##   <chr>             <chr>      <int>  <dbl>
## 1 College graduate Female      7478 0.123
## 2 College graduate Male        6668 0.109
## 3 H.S. or less     Female     12021 0.197
## 4 H.S. or less     Male        6571 0.108
## 5 Postgraduate     Female      4153 0.0681
## 6 Postgraduate     Male        4220 0.0692
## 7 Some college     Female     11536 0.189
## 8 Some college     Male        8321 0.136
```

Post-stratification

- We can also construct the post-stratification weights manually
    - Start by calculating the population proportions in each bin

```
acs_targets <- acs_targets %>% mutate(strata = str_c(gender_bin, educ_bin, age_bin, sep="-"),
                                       pop_proportion = Count/sum(Count))
```

- Do the same for the sample

```
cces <- cces %>% mutate(strata = str_c(gender_bin, educ_bin, age_bin, sep="-"))
cces_strat <- cces %>% group_by(strata) %>% summarize(n=n()) %>% ungroup() %>% mutate(samp_prop
```

- Join the datasets

```
cces <- cces %>% left_join(acs_targets%>% select(strata, pop_proportion), by="strata")
cces <- cces %>% left_join(cces_strat %>% select(strata, samp_proportion), by="strata")
```

Post-stratification

- Construct the post-stratification weights

```
cces <- cces %>% mutate(postStratWt = pop_proportion/samp_proportion)
```

- Take the weighted average to estimate Trump Approval

```
weighted.mean(cces$trumpApprove, cces$postStratWt)
```

```
## [1] 0.397
```

Population targets

- Did the weights equalize the distributions? Let's look again at the joint distribution of gender and education
- In the **population**

```
acs_targets %>% group_by(educ_bin, gender_bin) %>% summarize(n = sum(Count)) %>% ungroup() %>%
```

```
## # A tibble: 8 × 4
##   educ_bin         gender_bin        n   prop
##   <chr>            <chr>         <dbl>  <dbl>
## 1 College graduate Female     25521371 0.101
## 2 College graduate Male       22672764 0.0895
## 3 H.S. or less     Female     48061006 0.190
## 4 H.S. or less     Male       50981036 0.201
## 5 Postgraduate     Female     15043514 0.0594
## 6 Postgraduate     Male       13382290 0.0528
## 7 Some college     Female     41305976 0.163
## 8 Some college     Male       36328574 0.143
```

Population targets

- In the re-weighted **sample**

```
cces %>% group_by(educ_bin, gender_bin) %>% summarize(n = sum(postStratWt)) %>% ungroup() %>% m
```

```
## # A tibble: 8 × 4
##   educ_bin         gender_bin      n   prop
##   <chr>            <chr>       <dbl>  <dbl>
## 1 College graduate Female      6143. 0.101 
## 2 College graduate Male        5457. 0.0895
## 3 H.S. or less     Female     11568. 0.190 
## 4 H.S. or less     Male       12271. 0.201 
## 5 Postgraduate     Female      3621. 0.0594
## 6 Postgraduate     Male        3221. 0.0528
## 7 Some college     Female      9942. 0.163 
## 8 Some college     Male        8744. 0.143 
```

Post-stratification.

- Create the post-stratification weights using `postStratify` in `survey`

```
cces_postStrat <- postStratify(cces_surv_unwt, strata=~gender_bin + age_bin + educ_bin,
                               population = acs_targets %>% select(gender_bin, age_bin, educ_bi
```

- Then use `svymean`

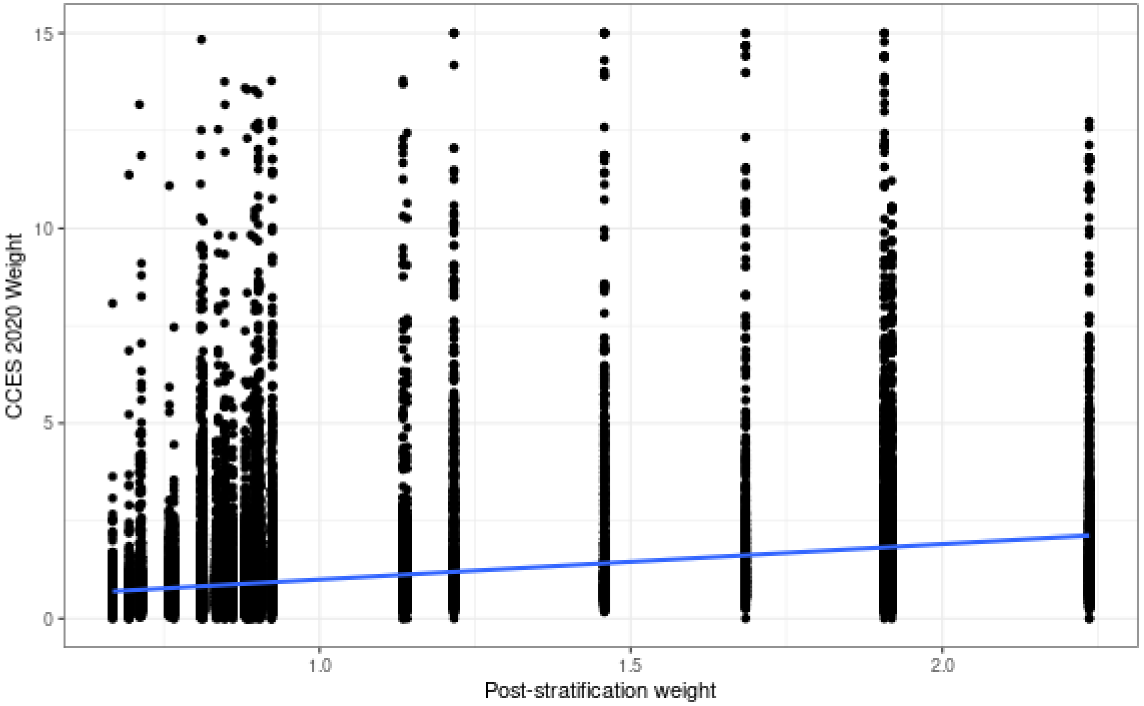
```
svymean(~trumpApprove, cces_postStrat)
```

```
##                mean SE
## trumpApprove 0.397  0
```

Post-stratification

- Plotting our post-stratification weights against the actual CCES weights

```
cces %>% ggplot(aes(x=postStratWt, y=commonweight)) + geom_point() + geom_smooth(method="lm") +
  xlab("Post-stratification weight") + ylab("CCES 2020 Weight")
```



Post-stratification

- Note that we can combine survey design weights with additional post-stratification weights.
  - For example, if we want to take a national survey and re-weight it to different demographic targets
- What happens if we combine our post-stratification weights with the CCES design weights

```
cces_postStrat_wt <- postStratify(cces_surv_wt, strata=~gender_bin + age_bin + educ_bin,
                              population = acs_targets %>% select(gender_bin, age_bin, educ_bi
```

- Our estimated Trump approval is closer to that original 44 percent (as expected, since our post-stratification variables are a subset of all the covariates that go into the CCES weights)

```
svymean(~trumpApprove, cces_postStrat_wt)
```

```
##               mean SE
## trumpApprove 0.443  0
```

Raking

- Sometimes we only know the marginals and not the joint distributions. We can still construct weights that get us balance on the *marginals*.
- Start by generating the marginal counts (in the real world, these would be **all** that we have).

```
acs_marginals <- list()
acs_marginals[["gender_bin"]] <- acs_targets %>% group_by(gender_bin) %>% summarize(Freq = sum(
acs_marginals[["age_bin"]] <- acs_targets %>% group_by(age_bin) %>% summarize(Freq = sum(Count)
acs_marginals[["educ_bin"]] <- acs_targets %>% group_by(educ_bin) %>% summarize(Freq = sum(Cour
```

- Make the raking design

```
cces_rake_unwt <- rake(cces_surv_unwt, sample.margins = list(~gender_bin, ~age_bin, ~educ_bin),
                       population.margins=acs_marginals)
cces$rakeWt <- weights(cces_rake_unwt)
```

- In this case, we actually do about as well as when we have the full joint distribution!

```
svymean(~trumpApprove, cces_rake_unwt)
```

```
##               mean SE
## trumpApprove 0.399  0
```

Raking

- Note that raking will **not** guarantee balance on the full joint distribution

- In the **population**

```
acs_targets %>% group_by(educ_bin, gender_bin) %>% summarize(n = sum(Count)) %>% ungroup() %>%
```

```
## # A tibble: 8 × 4
##   educ_bin         gender_bin        n   prop
##   <chr>            <chr>         <dbl>  <dbl>
## 1 College graduate Female     25521371 0.101
## 2 College graduate Male       22672764 0.0895
## 3 H.S. or less     Female     48061006 0.190
## 4 H.S. or less     Male       50981036 0.201
## 5 Postgraduate     Female     15043514 0.0594
## 6 Postgraduate     Male       13382290 0.0528
## 7 Some college     Female     41305976 0.163
## 8 Some college     Male       36328574 0.143
```

Raking

- In the re-weighted **sample**

```
cces %>% group_by(educ_bin, gender_bin) %>% summarize(n = sum(rakeWt)) %>% ungroup() %>% mutate
```

```
## # A tibble: 8 × 4
##   educ_bin         gender_bin         n   prop
##   <chr>            <chr>          <dbl>  <dbl>
## 1 College graduate Female     21885820. 0.0864
## 2 College graduate Male       26308315. 0.104
## 3 H.S. or less     Female     56712040. 0.224
## 4 H.S. or less     Male       42330002. 0.167
## 5 Postgraduate     Female     12005947. 0.0474
## 6 Postgraduate     Male       16419857. 0.0648
## 7 Some college     Female     39328059. 0.155
## 8 Some college     Male       38306491. 0.151
```