PLSC 40502: Problem Set 3

Solutions

February 20, 2022

This problem set is due at **11:59 pm on Friday, March 3rd**.

Please upload your solutions as a .pdf file saved as "Yourlastname_Yourfirstinitial_pset3.pdf"). In addition, an electronic copy of your .Rmd file (saved as "Yourlastname_Yourfirstinitial_pset3.Rmd") must be submitted to the course website at the same time. We should be able to run your code without error messages. In addition to your solutions, please submit an annotated version of this `.rmd` file saved as "Yourlastname_Yourfirstinitial_pset3_feedback.rmd", noting the problems where you needed to consult the solutions and why along with any remaining questions or concerns about the material. In order to receive credit, homework submissions must be substantially started and all work must be shown.

Problem 1

We'll continue with the same dataset from last problem set, the **Congressional Election Study**, an annual, large, nationally representative survey of the American population. We'll focus on the same outcome - predicting the level of support for the Section 232 Steel and Aluminum tariffs.

Be very careful in reading the variable names and definitions (feel free to use your code from last problem if it differed from the code in this problem).

The code below will load in the Common Content from the 2020 CES. Please download the file directly from `https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi%3A10.7910/DVN/E9N6PH` and place it into your data folder:

```
ces <- read_csv("data/CES20_Common_OUTPUT_vv.csv")
```

The next few code fragments will carry out the relevant pre-processing to generate bins for all four of the covariates of interest.

```
# State FIPS
ces$statefips <- str_pad(ces$inputstate, 2, pad="0")

# Outcome
ces <- ces %>% mutate(tariff = case_when(CC20_338b == "1" ~ 1,
                                         CC20_338b == "2" ~ 0,
                                         TRUE ~ NA_real_))

# Covariates
# Age
ces <- ces %>% mutate(age = 2020 - birthyr)
ces <- ces %>% mutate(age_bin = case_when(age>=18&age<=29 ~ "18-29",
                                          age>=30&age<=44 ~ "30-44",
                                          age>=45&age<=64 ~ "45-64",
                                          age>=65 ~ "65+"))
```

```r
# Gender
ces <- ces %>% mutate(gender_bin = case_when(gender == 1 ~ "Male",
                                             gender == 2 ~ "Female"))

# Race
ces <- ces %>% mutate(race_bin = case_when(race==3|hispanic==1 ~ "Hispanic",
                                           race==1 ~ "White",
                                           race ==2 ~ "Black",
                                           race == 4  ~ "Asian",
                                           TRUE ~ "Other"))

# Education
ces <- ces %>% mutate(educ_bin = case_when(educ>=1&educ<=2 ~ "H.S. or less",
                                           educ>=3&educ<=4 ~ "Some college",
                                           educ==5 ~ "College degree",
                                           educ==6 ~ "Postgraduate"))
```

Make the variables factors with the correct baselines

```r
ces$gender_bin <- relevel(as.factor(ces$gender_bin), "Male")
ces$age_bin <- relevel(as.factor(ces$age_bin), "18-29")
ces$educ_bin <- relevel(as.factor(ces$educ_bin), "H.S. or less")
ces$race_bin <- relevel(as.factor(ces$race_bin), "White")
```

Subset down to the non-missing data

```r
ces_full <- ces %>% filter(!is.na(tariff)&!is.na(age_bin)&!is.na(race_bin)&!is.na(gender_bin)&!is.na(edu
```

Part A

Using just the data from the CES survey and none of the covariates, estimate the share of residents in Rhode Island (FIPS code = 44) who support the tariff. Use the survey weights `commonweight`. Provide a 95% confidence interval for your estimate.

Using the `survey` package

```r
library(survey)
```

```
## Loading required package: grid
## Loading required package: Matrix
##
## Attaching package: 'Matrix'
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
## Loading required package: survival
##
## Attaching package: 'survey'
## The following object is masked from 'package:graphics':
##
##     dotchart
```

```
ces_survey_RI <- svydesign(~1, weights = ~commonweight, data=ces_full %>% filter(statefips == 44))
ces_mean <- svymean(~tariff, design = ces_survey_RI)
```

Calculate mean + asymptotic CI

```
ces_mean
```

```
##         mean   SE
## tariff 0.653 0.05
```

```
c(ces_mean[1] - qnorm(.975)*SE(ces_mean), ces_mean[1] + qnorm(.975)*SE(ces_mean))
```

```
## [1] 0.560 0.746
```

We estimate that about 65 percent of Rhode Island residents support the tariff with a 95% confidence interval of $[0.560, 0.746]$.

Part B

Using `brms`, fit a multilevel regression model that models the probability of supporting the tariff as a function of the four discrete covariates and a varying state-level intercept that pools to a common "us-wide" mean.

Warning: On my laptop, running 4 cores for 1000 burnin and 2000 total iterations took approximately **30 minutes** on this model so be aware of the likely run-time and prep accordingly. It may help to run fewer iterations to get a sense of the likely overall runtime on your computer.

You can and should **save** the output of a finished model in an `.Rdata` or `.rds` file to load into your workspace so that you don't have to repeatedly run the model after your first completed run.

Load the brms library

```
library(brms)
```

Estimate the model

```
set.seed(60635)
mrp_state <- brm(tariff ~ factor(gender_bin) + factor(age_bin) + factor(educ_bin) +
                   factor(race_bin) + (1 | statefips),
                           family = bernoulli(), data=ces_full,
                 cores=4, warmup=1000, iter=2000)
```

Part C

Now fit a model using `brms` that incorporates a state-level intercept that is pooled to a "census region-wide" mean instead (`region` in `ces`)

Hint: Consider a model that includes both a state and a region random intercept - in this model, the state-level intercepts can be interpreted as deviations from the "region" intercept.

Add in a region intercept for our model

```
set.seed(60635)
# Run this for more iterations since convergence at 2K is iffy
mrp_state_region <- brm(tariff ~ factor(gender_bin) + factor(age_bin) + factor(educ_bin) +
                          factor(race_bin) + (1 | region/statefips),
                           family = bernoulli(), data=ces_full, cores=4, warmup=2000,
                         iter=4000)
```

Part D

Load in the state-level post-stratification frame from the 2020 ACS as well as the state-region codebook (you'll want to merge this in to get the region codes for each frame).

```
acs_2020 <- read_csv("data/state_frame_2020_IPUMS.csv")
state_to_region <- read_csv("data/state_to_region.csv")
```

Obtain the post-stratified posterior mean estimate for the proportion supporting the tariff in each state using your model from Part B and your model for Part C. For each of the models, report the posterior mean estimate for Rhode Island and construct a 95% credible interval. Compare the results from the two models with your result from Part A.

Hint: Use `fitted` on your `brmsfit` objects to obtain predicted means for the post-stratification frame. Using the argument `summary = T` will generate the posterior means, but you may want to turn `summary = F` to obtain predictions for each MCMC draw in order to generate a credible interval for your Rhode Island estimate.

Note that you'll get slightly different numerical results due to the MCMC simulation and using a different seed.

First, merge `state_to_region` to `acs_2020` to make sure we have region codes for our post-stratification frame

```
acs_2020 <- acs_2020 %>% left_join(state_to_region, by = "fips")
acs_2020 <- acs_2020 %>% rename(statefips = fips)
```

Start by calculating the weights for each cell (grouped by state)

```
acs_2020 <- acs_2020 %>% group_by(statefips) %>% mutate(proportion = count/sum(count)) %>% ungroup()
```

Next, generate predicted probabilities for each cell.

```
state_fitted <- fitted(mrp_state, newdata=acs_2020, summary=F)
```

Now aggregate by the post-stratification frame weights

```
multiply_prop <- function(x) return(x*acs_2020$proportion) # Have to define this separately to get tidy
state_fitted_prop <- bind_cols(acs_2020 %>% select(statefips), as_tibble(t(state_fitted)) %>% mutate_al
state_mcmc <- state_fitted_prop %>% group_by(statefips) %>% summarize_all(sum) %>% ungroup() %>% select
```

Now get the state-level posterior means and 95% credible intervals (equal tailed is fine)

```
state_results <- acs_2020 %>% group_by(statefips) %>% summarize(statefips = statefips[1])
state_results$posterior_mean <- apply(state_mcmc, 1, mean)
state_results$ci_95_lower <- apply(state_mcmc, 1, function(x) quantile(x, .025))
state_results$ci_95_upper <- apply(state_mcmc, 1, function(x) quantile(x, .975))
```

Print the results specifically for Rhode Island

```
state_results %>% filter(statefips == "44")
```

```
## # A tibble: 1 x 4
##   statefips posterior_mean ci_95_lower ci_95_upper
##   <chr>              <dbl>       <dbl>       <dbl>
## 1 44                 0.566       0.531       0.605
```

Our posterior mean estimate for the share of support for the tariff in Rhode Island is 56.6 percent with a 95% credible interval of $[.531, .605]$ using the state intercepts-only model. Our point estimate has been attenuated somewhat towards the national mean. Incorporating the model has substantially improved the precision of

our estimate – the width of the credible interval is about half that of the confidence interval reported in Part A.

For the state and region model:

Next, generate predicted probabilities for each cell.

```
state_region_fitted <- fitted(mrp_state_region, newdata=acs_2020, summary=F)
```

Now aggregate by the post-stratification frame weights

```
state_region_fitted_prop <- bind_cols(acs_2020 %>% select(statefips), as_tibble(t(state_region_fitted))
state_region_mcmc <- state_region_fitted_prop %>% group_by(statefips) %>% summarize_all(sum) %>% ungrou
```

Now get the state-level posterior means and 95% credible intervals (equal tailed is fine)

```
state_region_results <- acs_2020 %>% group_by(statefips) %>% summarize(statefips = statefips[1])
state_region_results$posterior_mean <- apply(state_region_mcmc, 1, mean)
state_region_results$ci_95_lower <- apply(state_region_mcmc, 1, function(x) quantile(x, .025))
state_region_results$ci_95_upper <- apply(state_region_mcmc, 1, function(x) quantile(x, .975))
```

Print the results specifically for Rhode Island

```
state_region_results %>% filter(statefips == "44")
```

```
## # A tibble: 1 x 4
##   statefips posterior_mean ci_95_lower ci_95_upper
##   <chr>              <dbl>       <dbl>       <dbl>
## 1 44                 0.565       0.530       0.602
```

Our posterior mean estimate for the share of support for the tariff in Rhode Island is 56.5 percent with a 95% credible interval of $[.530, .602]$ using the state and region random intercepts model. Incorporating the pooling of the state intercepts towards region-level rather than a grand mean actually ends up improving precision somewhat without altering the point estimate substantially.

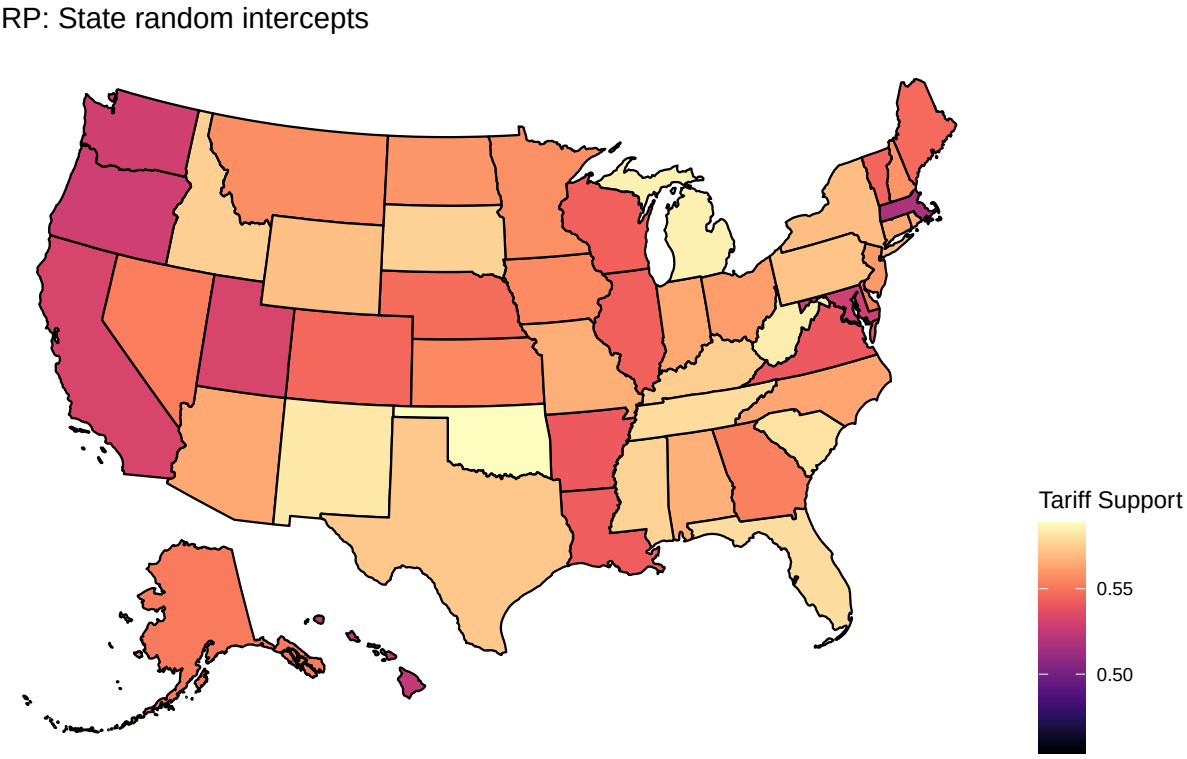
Part E

Using the `usmap` package, generate two maps of your posterior mean MRP estimates - one for the state-varying intercepts model and another for the state-/region- intercepts model. Compare the results. What do you notice about the output of the two models?

```
library(usmap)
library(viridis) # For the scales
```

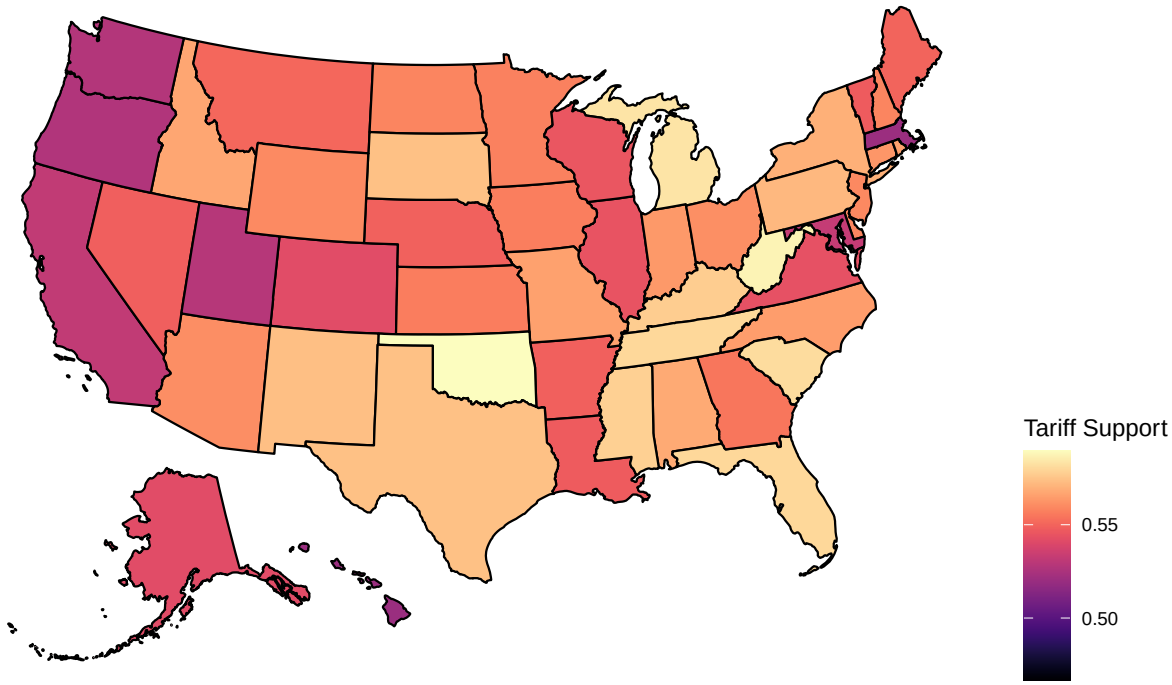
First the state-level model

```
plot_usmap(regions="states", data= state_results %>% mutate(fips=statefips), values="posterior_mean") +
  theme(legend.position = "right") + ggtitle("MRP: State random intercepts")
```



Next, the state and intercept level model

```
plot_usmap(regions="states", data= state_region_results %>% mutate(fips=statefips), values="posterior_m
  theme(legend.position = "right") + ggtitle("MRP: State and region random intercepts")
```



Consistent with the results we see for Rhode Island, it does not appear that the maps are too different

in the point estimates, though incorporating region random effects does appear to attenuate some of the more extremely high state-level estimates downward, such as in Wyoming and in New Mexico. We see some of the benefit from pooling towards the regional mean rather than grand mean – support in the West is noticeably lower than in other states and this allows the noisier estimates for states like Wyoming to be primarily informed by their regional neighbors rather than by all of the states in the sample. Conversely, the west coast estimates (California, Washington, and Oregon) are somewhat lower as they are not being pulled to the national mean but rather a regional mean.