Problem Set 7

Nosa Lawani

4/7/2021

```
set.seed(2319)
object_1 <- kenya %>%
  filter(treatment %in% c("local + SMS", "control")) %>%
  drop_na() %>%
  rename(perc_reg = reg_byrv13) %>%
  select(-rv13, -block, -poll_station) %>%
  slice_sample(n = 100)
```

```
fit_1 <- stan_glm(data = object_1,
         formula = perc_reg ~ treatment + mean_age,
         seed = 54,
         refresh = 0)
```

#Written 1

$$y_i = \beta_0 + \beta_1 local + SMS_i + \beta_2 mean\_agei + \epsilon_i$$

Written 2

```
print(fit_1, digits = 4)
```

```
## stan_glm
##  family:       gaussian [identity]
##  formula:      perc_reg ~ treatment + mean_age
##  observations: 100
##  predictors:   3
## ------
##                        Median  MAD_SD
## (Intercept)            -0.1231  0.0617
## treatmentlocal + SMS    0.0264  0.0076
## mean_age                0.0031  0.0015
##
## Auxiliary parameter(s):
##       Median MAD_SD
## sigma 0.0375 0.0027
##
## ------
## * For help interpreting the printed output see ?print.stanreg
## * For info on the priors used see ?prior_summary.stanreg
```

The median value for the Intercept is the best estimate for beta0, which is the perc_reg, i.e. the percent of registered voters in a polling district, if no one was treated and the mean_age was 0. The median value for local + SMS is our best estimate for beta1, which represents the change in percent of registered voters treated counties. The median value for mean_age is the best estimate for beta2, which represents the change in the percent of registered voters for every increase of one year in the polling district's mean_age. For all of these estimates, we can discern our confidence in them using the MAD SD; we are 95% confident that the true value lies within two MAD SDs of the median.

```
fit_2 <- stan_glm(data = object_1,
         formula = perc_reg ~ treatment + poverty + distance + pop_density + mean_age,
         seed = 47,
         refresh = 0)
```

```
loo_1 <- loo(fit_1, k_threshold = 0.7)
```

```
## 1 problematic observation(s) found.
## Model will be refit 1 times.
```

```
##
## Fitting model 1 out of 1 (leaving out observation 72)
```

```
loo_2 <- loo(fit_2, k_threshold = 0.7)
```

```
## 2 problematic observation(s) found.
## Model will be refit 2 times.
```

```
##
## Fitting model 1 out of 2 (leaving out observation 72)
```

```
##
## Fitting model 2 out of 2 (leaving out observation 78)
```

```
object_2 <- loo_compare(loo_1, loo_2)
```

Written 3

```
print(object_2)
```

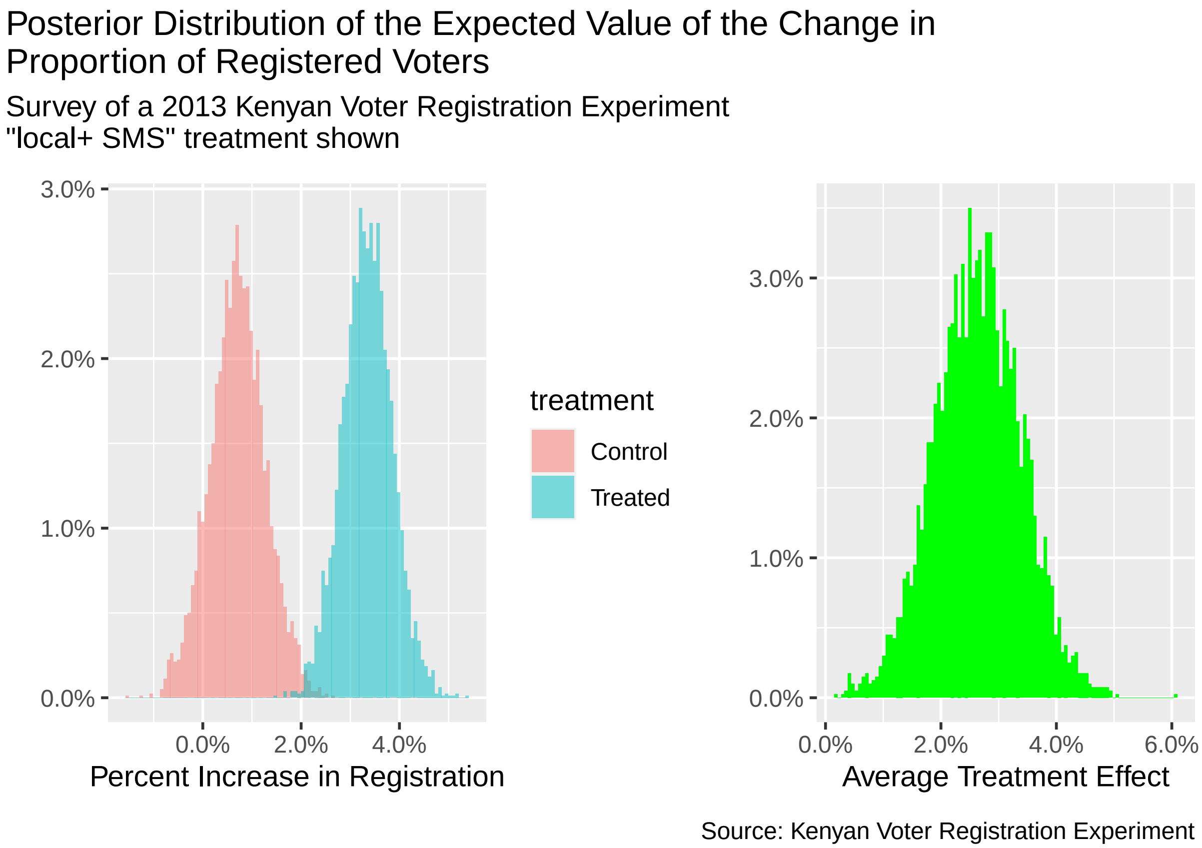
```
##       elpd_diff se_diff
## fit_2  0.0       0.0
## fit_1 -3.6       5.4
```

From the results, our second model, fit_2, has a slightly lower elpd and is therefore more accurate. However, the median value for the difference between fit_1 and fit_2 is less than 4, a value so small it becomes hard to distinguish it from noise and so to determine which of the two values is truly better. This median estimate is really only our best guess of the difference between the two. We, however, can be 95% confident that the actual value of the difference is within two standard errors of the mean. With that taken into account, fit_1 could either be more appreciably worse of an estimate than fit_2 or appreciably greater than fit_2. This only gives more reason to not consider one of these models better than the other.

```
newobs <- tibble(mean_age = 42, treatment = c("local + SMS", "control"))
object_3 <- posterior_epred(object = fit_1,
                  newdata = newobs) %>%
  as.tibble() %>%
  mutate_all(as.numeric) %>%
  rowwise() %>%
  rename(Treated = `1`,
         Control = `2`) %>%
  mutate(ATE = Treated - Control)
```

```
## Warning: `as.tibble()` was deprecated in tibble 2.0.0.
## Please use `as_tibble()` instead.
## The signature and semantics have changed, see `?as_tibble`.
```

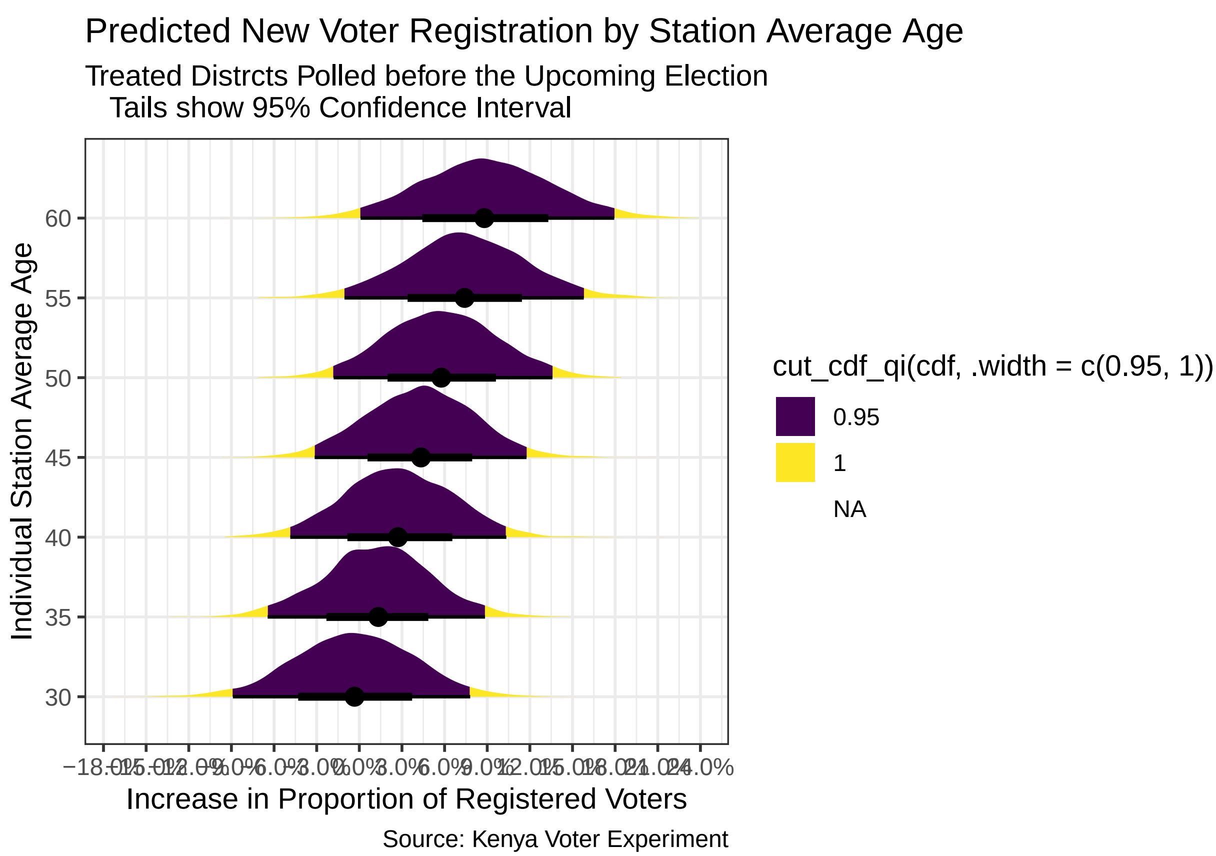
```
p1 <- object_3 %>%
   pivot_longer(cols = Treated:Control,
                names_to = "treatment",
               values_to = "results") %>%
  ggplot(aes(x = results, fill = treatment)) +
  geom_histogram(bins = 100,
                 position = "identity",
                 alpha = 0.5,
                 aes(y = after_stat(count) / sum(count))) +
  scale_x_continuous(labels = scales::percent_format()) +
  scale_y_continuous(labels = scales::percent_format()) +
  labs(x = "Percent Increase in Registration",
       y = "")

p2 <- object_3 %>%
  ggplot(aes(x = ATE)) +
  geom_histogram(bins = 100,
                 fill = "#00FF00",
                 aes(y = after_stat(count) / sum(count)))  +
  scale_x_continuous(labels = scales::percent_format()) +
  scale_y_continuous(labels = scales::percent_format()) +
  labs(x = "Average Treatment Effect",
       y = "")

p1 + p2 +
  plot_annotation(title = "Posterior Distribution of the Expected Value of the Change in
Proportion of Registered Voters",
         subtitle = 'Survey of a 2013 Kenyan Voter Registration Experiment
"local+ SMS" treatment shown',
caption = "Source: Kenyan Voter Registration Experiment")
```



Written 4 The first plot above shows the posterior for the expected value of the change in the number of voters in a district whose mean age is 42 under both treatment and control. While this same value was negative when age was 0, now that the age has been given a much more reasonable value, draws from the same fitted object show much more reasonable answers. The second graph is a posterior for the estimated average treatment effect. It was obtained by subtracting each of the 4000 draws which comprise the first plot's control posterior from one of the 4000 draws which comprise the first plot's treatment posterior. In doing so, we have calculated an estimate for the Average Treatment Effect, which we cannot know exactly due to the Fundamental Problem of Causal Inference.

```
newobs  <- tibble(mean_age = c(30, 35, 40, 45, 50, 55,
                               60), treatment = "local + SMS")
posterior_predict(fit_1,
                  newobs) %>%
  as.tibble() %>%
  janitor::clean_names() %>%
  mutate_all(as.numeric) %>%
  rename("30" = x1,
         "35" = x2,
         "40" = x3,
         "45" = x4,
         "50" = x5,
         "55" = x6,
         "60" = x7) %>%
  pivot_longer(cols = "30":"60",
               names_to = "age",
               values_to = "results") %>%
  ggplot(aes(x = results, y = age)) +
```

```
  stat_halfeye(aes(fill = stat(cut_cdf_qi(cdf, .width = c(0.95, 1))))) +
  labs(title = "Predicted New Voter Registration by Station Average Age",
       subtitle = "Treated Distrcts Polled before the Upcoming Election
   Tails show 95% Confidence Interval",
   y = "Individual Station Average Age",
   x = "Increase in Proportion of Registered Voters",
   caption = "Source: Kenya Voter Experiment") +
scale_x_continuous(breaks = (seq(-.24, .24, .03)),
                   labels = scales::percent_format()) +
  theme_bw()
```




```
guide_axis(check.overlap = TRUE)
```

```
## $title
## list()
## attr(,"class")
## [1] "waiver"
##
## $check.overlap
## [1] TRUE
##
## $angle
## NULL
##
## $n.dodge
```

```
## [1] 1
##
## $order
## [1] 0
##
## $position
## list()
## attr(,"class")
## [1] "waiver"
##
## $available_aes
## [1] "x" "y"
##
## $name
## [1] "axis"
##
## attr(,"class")
## [1] "guide" "axis"
```

Written 5

Each access plots