

DS-UA 201: Problem Set 2

Ziwei Xu (Vanessa)

October 13, 2020

This problem set is due at **11:59 pm on Tuesday, October 13th**. The data are on the course website.

Please upload your solutions as a .pdf file saved as `Yourlastname_Yourfirstinitial_pset2.pdf`). In addition, an electronic copy of your .Rmd file (saved as `Yourlastname_Yourfirstinitial_pset2.Rmd`) must be submitted to the course website at the same time. We should be able to run your code without error messages. Please note on your problem set if you collaborated with another student and, if so, whom. In order to receive credit, homework submissions must be substantially started and all work must be shown. Late assignments will not be accepted.

Problem 1

Do international election monitors reduce the incidence of electoral fraud? Hyde (2007) studies the 2003 presidential election in Armenia, an election that took place during a period where the incumbent ruling party headed by President Robert Kocharian had consolidated power and often behaved in ways that were considered undemocratic.

The full citation for this paper is

Hyde, Susan D. “The observer effect in international politics: Evidence from a natural experiment.” *World Politics* 60.1 (2007): 37-63.

At the time of the election, OSCE/ODIHR election monitors reported widespread electoral irregularities that favored the incumbent party such as ballot-box stuffing (pp. 47). However, we do not necessarily know whether these irregularities would have been worse in the absence of monitors. Notably, not all polling stations were monitored – the OSCE/ODIHR mission could only send observers to some of the polling stations in the country. Since in the context of this election only the incumbent party would have the capacity to carry out significant election fraud, Hyde examines whether the presence of election observers from the OSCE/ODIHR mission at polling stations in Armenia reduced the incumbent party’s vote share at that polling station.

For the purposes of this problem, you will be using the `armenia2003.dta` dataset

The R code below will read in this data (which is stored in the STATA .dta format)

```
library(tidyverse)
library(haven)

### Hyde (2007) Armenia dataset
armenia <- read_dta("armenia2003.dta")
```

This dataset consists of 1764 observations polling-station-level election results from the 2003 Armenia election made available by the Armenian Central Election Commission. The election took place over two rounds with an initial round having a large number of candidates and a second, run-off election, between Kocharian and the second-place vote-getter, Karen Demirchyan. We will focus on monitoring and voting in the first round. The specific columns you will need are:

- **kocharian** - Round 1 vote share for the incumbent (Kocharian)
- **mon_voting** - Whether the polling station was monitored in round 1 of the election
- **turnout** - Proportion of registered voters who voted in Round 1
- **totalvoters** - Total number of registered voters recorded for the polling station
- **total** - Total number of votes cast in Round 1
- **urban** - Indicator for whether the polling place was in an urban area (0 = rural, 1 = urban)
- **nearNagorno** - Indicator for whether the polling place is near the Nagorno-Karabakh region (0 = no, 1 = yes)

Part A

Hyde describes the study as a “natural experiment,” stating:

“I learned from conversations with staff and participants in the OSCE observation mission to Armenia that the method used to assign observers to polling stations was functionally equivalent to random assignment. This permits the use of natural experimental design. Although the OSCE/ODIHR mission did not assign observers using a random numbers table or its equivalent, the method would have been highly unlikely to produce a list of assigned polling stations that were systematically different from the polling stations that observers were not assigned to visit. Each team’s assigned list was selected arbitrarily from a complete list of polling stations.” (p. 48)

What makes this study a “natural experiment” and not a true experiment? What assumption must the study defend in order to identify the causal effect of election monitoring that would be guaranteed to hold in a randomized experiment? What might be some possible reasons why that assumption might not hold in this design?

The study is not a true experiment, because the monitored polling stations were not truly chosen randomly. This would be important, because then the null hypothesis would be that the parameterized distributions of relative votes for the incumbent candidate are the same for the monitored as well as the unmonitored polling stations. The null hypothesis means that there is no evidence for difference in cheating in the vote ratios for monitored vs unmonitored polling stations. It would be a natural experiment if the distribution of monitoring points is near-representative for the vote ratios distribution over the polling stations.

The most critical part in the argument is that the monitoring stations may not have been chosen to avoid high opposition votes, because that would leave more room for cheating in these polling stations to be insignificant, because the monitored stations do have a higher voting ratio for the incumbent candidate anyway.

Part B

For the purposes of this part, assume election monitors were assigned as the author describes - in a manner “functionally equivalent to random assignment.” Using the difference-in-means estimator, estimate the average treatment effect of election monitoring on incumbent vote share in round 1. Provide a 95% asymptotic

confidence interval and interpret your results. Can we reject the null of no average treatment effect at the $\alpha = 0.05$ level?

So we decided to compute the differences depending on the 2 predictors rural/urban and nearNargorno/farofNargorno. The 95% confidence intervals are computed as below:

```
monitored <- armenia$mon_voting==1
unmon <- armenia$mon_voting==0
urban <- armenia$urban==1
nearNagorno <- armenia$nearNagorno==1

samples0un <- armenia[unmon & urban & nearNagorno, ]$kocharian
samples0uf <- armenia[unmon & urban & !nearNagorno, ]$kocharian
samples0rn <- armenia[unmon & !urban & nearNagorno, ]$kocharian
samples0rf <- armenia[unmon & !urban & !nearNagorno, ]$kocharian
n0 <- c(length(samples0un), length(samples0uf), length(samples0rn), length(samples0rf))
m0 <- c(mean(samples0un), mean(samples0uf), mean(samples0rn), mean(samples0rf))
s0 <- c(sd(samples0un), sd(samples0uf), sd(samples0rn), sd(samples0rf))

samples1un <- armenia[monitored & urban & nearNagorno, ]$kocharian
samples1uf <- armenia[monitored & urban & !nearNagorno, ]$kocharian
samples1rn <- armenia[monitored & !urban & nearNagorno, ]$kocharian
samples1rf <- armenia[monitored & !urban & !nearNagorno, ]$kocharian

n1 <- c(length(samples1un), length(samples1uf), length(samples1rn), length(samples1rf))
m1 <- c(mean(samples1un), mean(samples1uf), mean(samples1rn), mean(samples1rf))
s1 <- c(sd(samples1un), sd(samples1uf), sd(samples1rn), sd(samples1rf))
sprintf('unmonitored:\n  {(urban,nearN),(urban,farN),(rual,nearN),(rural,farN)}: count=(%d, %d, %d, %d)

## [1] "unmonitored:\n  {(urban,nearN),(urban,farN),(rual,nearN),(rural,farN)}: count=(25, 491, 34, 466)

sprintf('monitored: \n  {(urban,nearN),(urban,farN),(rual,nearN),(rural,farN)}: count=(%d, %d, %d, %d)

## [1] "monitored: \n  {(urban,nearN),(urban,farN),(rual,nearN),(rural,farN)}: count=(16, 447, 19, 466)

deltas <- m0 -m1
sError <- sqrt(s0*s0/n0 +s1*s1/n1)
DoFun <- (s0*s0/n0+s1*s1/n1)/((s0*s0/n0)^2/(n0-1)+(s1*s1/n1)^2/(n1-1))

mError <- c(qt(0.95, DoFun[1])*sError[1],qnorm(0.95)*sError[2], qt(0.95, DoFun[3])*sError[3],qnorm(0.95,
sprintf('95%% confidence intervals for {(urban,nearN),(urban,farN),(rural,nearN),(rural,farN)}: (%.3f±%

## [1] "95% confidence intervals for {(urban,nearN),(urban,farN),(rural,nearN),(rural,farN)}: (-0.062±0
```

So the probabilities are calculated as:

```
probabs <- c(1-pnorm(abs(deltas[1])/mError[1]),1-pnorm(abs(deltas[2])/mError[2]), 1-pnorm(abs(deltas[3])/mError[3]),
print(probabs)
```

```
## [1] 2.700204e-01 1.961419e-05 4.131635e-01 9.542763e-02
```

Therefore we are certain that there was an effect of monitoring in urban areas far off Nagorno. For the other parts we cannot prove anything.

Part C

Evaluate the author's identification assumptions by examining whether the treatment is balanced on three pre-treatment covariates: the total number of registered voters, whether a polling place was in an urban area, and whether the polling place was located near the Nagorno-Karabakh region (Kocharian's home region and a disputed territory between Armenia and Azerbaijan). Discuss your results. Are they consistent with the author's description of "as-if random" assignment?

compute 4 probabilities and compare that they are the same with and without monitoring. *****

Part D

Divide the sample into five strata based on the total number of registered voters at each polling station (`totalvoters`):

Stratum	Total Registered Voters
Tiny	<code>totalvoters < 430</code>
Small	<code>430 ≤ totalvoters < 1192</code>
Medium	<code>1192 ≤ totalvoters < 1628</code>
Large	<code>1628 ≤ totalvoters < 1879</code>
Huge	<code>1879 ≤ totalvoters</code>

Estimate the average treatment effect of election monitoring in round 1 on incumbent vote share using a stratified difference-in-means estimator, stratifying on the total number of registered voters. Provide a 95% asymptotic confidence interval and interpret your results. Can we reject the null of no average treatment effect at the $\alpha = 0.05$ level? Compare your answer to your estimate from Part B and discuss any differences you see.

```
tiny <- armenia$totalvoters<430
small <- 430<armenia$totalvoters & armenia$totalvoters<1192
medium <- 1192<armenia$totalvoters & armenia$totalvoters<1628
large <- 1628<armenia$totalvoters & armenia$totalvoters<1879
huge <- 1879<armenia$totalvoters
```

```
# Difference-in-means function
diff_in_means <- function(treated, control, alpha = .05){

  # Point estimate
```

```

est <- mean(treated) - mean(control)

# Standard error
se <- sqrt(var(treated)/length(treated) +
            var(control)/length(control))

# Confidence interval
ci <- c(est - qnorm(1-alpha/2)*se,
        est + qnorm(1-alpha/2)*se)

# P-value
pvalue <- 2*pnorm(-abs(est/se))

# Output
#out_data = data.frame(est = est, se = se, ci_lower = ci[1],
#                        ci_upper = ci[2],
#                        pval = pvalue,
#                        alpha = alpha)

return(c(est, se, pvalue))
}

```

```

un <- armenia[unmon & small, ]$kocharian
un

```

```

## [1] 0.2269000 0.5183000 0.5769000 0.3667000 0.3861000 0.4484000 0.2546000
## [8] 0.3508000 0.6483999 0.6777000 0.5567000 0.2792000 0.4196000 0.4316000
## [15] 0.3523000 0.2831000 0.3172000 0.2755000 0.5167000 0.2278000 0.4142000
## [22] 0.2699000 0.5102000 0.1538000 0.1703000 0.2930000 0.3689000 0.7290000
## [29] 0.2157000 0.5698000 0.4358000 0.1996000 0.3473000 0.9520000 0.3280000
## [36] 0.8584000 0.2707000 0.9507000 0.4045000 0.4557000 0.3176000 0.6014000
## [43] 0.8863000 0.7024000 0.5660000 0.4651000 0.3562000 0.4700000 0.9443000
## [50] 0.1436000 0.2903000 0.5032000 0.6120000 0.2762000 0.4978000 0.5193000
## [57] 0.6000000 0.1904000 0.4881000 0.4518000 0.3333000 0.4164000 0.2881000
## [64] 0.3428000 0.3756000 0.4126000 0.4629000 0.0764000 0.3775000 0.3687000
## [71] 0.8855000 0.6726000 0.3076000 0.6235000 0.5691000 0.5442000 0.4247000
## [78] 0.6376000 0.4735000 0.4464000 0.6539000 0.3108000 0.5314000 0.6235000
## [85] 0.3513000 0.3496000 0.8966000 0.5142000 0.3747000 0.6650000 0.7208999
## [92] 0.2402000 0.9156000 0.2198000 0.1690000 0.2258000 0.4486000 0.3531000
## [99] 0.9867000 0.3233000 0.7205999 0.3180000 0.2783000 0.4623000 0.6392000
## [106] 0.9345000 0.3980000 0.5692000 0.9157000 0.6005000 0.5411000 0.6526999
## [113] 0.4703000 0.4572000 0.7244000 0.4714000 0.8869000 0.5967000 0.7517000
## [120] 0.3984000 0.5618000 0.5992000 0.8634000 0.3475000 0.6580001 0.3281000
## [127] 0.7910000 0.4067000 0.6818000 0.4020000 0.5796000 0.7307000 0.7716000
## [134] 0.5062000 0.5913000 0.9395000 0.2424000 0.5074000 0.6435000 0.7357000
## [141] 0.9823000 0.7207000 0.6059000 0.7535000 0.7663000 0.3978000 0.5511000
## [148] 0.4549000 0.7837999 0.5208000 0.8774000 0.6302000 0.4479000 0.7730000
## [155] 0.6951000 0.6472000 0.4734000 0.6697000 0.6128000 0.5398000 0.5375000
## [162] 0.6536000 0.3881000 0.3983000 0.4959000 0.5261000 0.3802000 0.6694000
## [169] 0.5526000 0.5106000 0.6543000 0.4671000 0.5804000 0.7330000 0.5620000
## [176] 0.6828000 0.5332000 0.6516001 0.8159000 0.7169999 0.6761000 0.5224000

```

```
## [183] 0.4816000 0.5237000 0.7002000 0.8250000 0.6003000 0.2058000 0.4329000
## [190] 0.5470000 0.6598001 0.6416000 0.3652000 0.6703000 0.4589000 0.4632000
## [197] 0.6515000 0.5871000 0.4149000 0.4474000 0.5579000 0.6603000 0.6321000
## [204] 0.4184000 0.6611000 0.6322000 0.4375000 0.3132000 0.5502000 0.5798000
## [211] 0.9463000 0.7692000 0.3364000 0.3492000 0.7307000 0.1900000 0.5053000
## [218] 0.1174000 0.8068000 0.6062000 0.5856000 0.7205001 0.4874000 0.4619000
## [225] 0.6008000 0.3757000 0.5176000 0.2039000 0.4701000 0.7960000 0.8128000
## [232] 0.7009000 0.8196000 0.8947000 0.7778000 0.9719000 0.4783000
## attr("label")
## [1] "R1 vote share of Kocharian"
## attr("format.stata")
## [1] "%9.0g"
```

```
# for (stratum in c(tiny, small, medium, large, huge)) {
  un <- armenia[unmon & tiny, ]$kocharian
  mon <- armenia[monitored & tiny, ]$kocharian
  result <- diff_in_means(mon, un)
  print(sprintf('for tiny towns: ratio = %.3f±%.3f, p=%.3f', result[1], result[2], result[3]))
}
```

```
## [1] "for tiny towns: ratio = 0.007±0.029, p=0.794"
```

```
un <- armenia[unmon & small, ]$kocharian
mon <- armenia[monitored & small, ]$kocharian
result <- diff_in_means(mon, un)
print(sprintf('for small towns: ratio = %.3f±%.3f, p=%.3f', result[1], result[2], result[3]))
```

```
## [1] "for small towns: ratio = -0.018±0.025, p=0.452"
```

```
un <- armenia[unmon & medium, ]$kocharian
mon <- armenia[monitored & medium, ]$kocharian
result <- diff_in_means(mon, un)
print(sprintf('for medium towns: ratio = %.3f±%.3f, p=%.3f', result[1], result[2], result[3]))
```

```
## [1] "for medium towns: ratio = -0.010±0.021, p=0.615"
```

```
un <- armenia[unmon & large, ]$kocharian
mon <- armenia[monitored & large, ]$kocharian
result <- diff_in_means(mon, un)
print(sprintf('for large towns: ratio = %.3f±%.3f, p=%.3f', result[1], result[2], result[3]))
```

```
## [1] "for large towns: ratio = -0.038±0.019, p=0.042"
```

```
un <- armenia[unmon & huge, ]$kocharian
mon <- armenia[monitored & huge, ]$kocharian
result <- diff_in_means(mon, un)
print(sprintf('for huge towns: ratio = %.3f±%.3f, p=%.3f', result[1], result[2], result[3]))
```

```
## [1] "for huge towns: ratio = -0.025±0.018, p=0.169"
```

With only this stratification it seems that he cheated only in the large towns.

Part E

In Table 4 of the paper, Hyde uses an estimator for the average treatment effect of a polling place receiving election monitors in round 1 on the incumbent's vote share in round 1 *conditional* on the total number of votes cast in the election. Will this approach be unbiased for the average treatment effect of election monitors on the incumbent's vote share if we believe that one of the mechanisms through which election monitoring operates is by reducing the incidence of ballot-stuffing (inflating the number of "cast" votes in the election)? Why or why not?

No, this approach is unbiased.

Problem 2

Consider an experiment with N units. Each unit i in the sample belongs to one of G mutually exclusive strata. $G_i = g$ denotes that the i th unit belongs to stratum g . N_g denotes the size of stratum g and $N_{t,g}$ denotes the number of treated units in that stratum. Suppose that treatment is assigned via block-randomization. Within each stratum, $N_{t,g}$ units are randomly selected to receive treatment and the remainder receive control (complete randomization). Suppose also that the proportion of treated units in each stratum, $\frac{N_{t,g}}{N_g}$, varies depending on the stratum. After treatment is assigned, you record an outcome Y_i for each unit in the sample. Assume consistency holds with respect to the potential outcomes:

$$Y_i = D_i Y_i(1) + (1 - D_i) Y_i(0)$$

Let $w(g) = P(D_i = 1 | G_i = g)$ denote the known (constant) probability that unit i would receive treatment if it's stratum membership is g .

Instead of using the stratified difference-in-means estimator, your colleague suggests an alternative that assigns a weight to each unit and takes two weighted averages.

$$\hat{\tau}_w = \frac{1}{N} \sum_{i=1}^N \frac{D_i Y_i}{w(G_i)} - \frac{(1 - D_i) Y_i}{1 - w(G_i)}$$

Show that $\hat{\tau}_w$ is unbiased for the average treatment effect τ :

$$\tau = E[Y_i(1) - Y_i(0)]$$

Hints:

- For a unit with $G_i = g$, the probability of receiving treatment is $\frac{N_{t,g}}{N_g}$

- Consider splitting up the sum from $i = 1$ to N to a double-sum over each of the G groups and over the units i in group $G_i = g$ (since the groups are exhaustive and mutually exclusive, you can think of this as partitioning of the set of i observations by their group membership).

$$\begin{aligned}
 \hat{T}_W &:= \frac{1}{N} \sum_{i=1}^N \left[\frac{D_i Y_i}{w(G_i)} - \frac{(1-D_i) Y_i}{1-w(G_i)} \right] \\
 &= \frac{1}{N} \sum_{g \in G} \sum_{i: G_i = g} \left[\frac{D_i Y_i}{w(g)} - \frac{(1-D_i) Y_i}{1-w(g)} \right] \\
 &= \frac{1}{N} \sum_{g \in G} \left(\sum_{\substack{i: G_i = g \\ D_i = 1}} \frac{Y_i}{w(g)} - \sum_{\substack{i: G_i = g \\ D_i = 0}} \frac{Y_i}{1-w(g)} \right) \\
 &= \frac{1}{N} \sum_{g \in G} \left[\frac{N_{1,g}}{w(g)} \frac{1}{N_{1,g}} \sum_{\substack{i: G_i = g \\ D_i = 1}} Y_i - \frac{N_{0,g}}{1-w(g)} \frac{1}{N_{0,g}} \sum_{\substack{i: G_i = g \\ D_i = 0}} Y_i \right] \\
 &\stackrel{(***)}{=} \frac{1}{N} \sum_{g \in G} N_g \left[\mathbb{E}(Y_i | 1; G_i = g) - \mathbb{E}(Y_i | 0; G_i = g) \right] \\
 &= \sum_{g \in G} \frac{N_g}{N} \mathbb{E}(Y_i | 1) - \mathbb{E}(Y_i | 0; G_i = g) \\
 &= \mathbb{E}(Y | 1) - \mathbb{E}(Y | 0) \quad \square
 \end{aligned}$$

The proof to this question can be found in the image below:

If the image file can not be found, please check the submitted pdf file.

Problem 3

For this question you will need the `SierraLeone_data.dta` dataset (available on the course website) based on a field experiment conducted by Casey et al 2012. You will re-analyze this experiment using Fisher's randomization inference.

Aid organizations in developing countries spend billions of dollars every year promoting accountability, competence and inclusion of under-represented groups. Arguably the most popular strategy for these efforts has been community-driven development (CDD) programs. CDD projects are initiatives that attempt to bolster local coordination and enhance political participation by providing financial grants for local public goods and small enterprise development.

Casey et al 2012 explore the effectiveness of a CDD program in post-conflict Sierra Leone. The researchers block-randomized treatment (access to the CDD program) at the village level. That is, within each block (here chiefdoms) consisting of N_g villages, the researchers randomly assigned $N_{t,g}$ villages to receive treatment. Overall, $N_t = 116$ villages received the treatment out of a total of $N = 233$.

```
# Read in the data
raw <- read_dta("SierraLeone_data.dta")
sierraLeone <- raw[complete.cases(raw), ]
```

The variables you will need are:

- `community_bank` - Whether the village has a community bank by the end of the experiment
- `treat_control` - The village's treatment status
- `chief_2004census` - The census area (chiefdom) at which block randomization was conducted
- `id_vill` - Unique village identifier

Part A

Estimate the average treatment effect of the CDD program on the probability that a village has a community bank using a stratified difference-in-means estimator (the strata here being chiefdoms).

```
# Difference-in-means function
diff_in_means <- function(treated, control, alpha = .05){

  # Point estimate
  est <- mean(treated) - mean(control)

  # Standard error
  se <- sqrt(var(treated)/length(treated) +
             var(control)/length(control))

  # Confidence interval

  ci <- c(est - qnorm(1-alpha/2)*se,
         est + qnorm(1-alpha/2)*se)

  # P-value
  pvalue <- 2*pnorm(-abs(est/se))

  # Output
  out_data = data.frame(est = est, se = se, ci_lower = ci[1],
                        #
                        ci_upper = ci[2],
                        #
                        pval = pvalue,
```

```

#                               alpha = alpha)

return(c(est, se, pvalue))
}

chiefs <- unique(sierraLeone$chief_2004census)
treated <- sierraLeone$treat_control==1
for (chief in chiefs) {
  treated_values = sierraLeone[treated & sierraLeone$chief_2004census==chief,]$community_bank
  control_values = sierraLeone[!treated & sierraLeone$chief_2004census==chief,]$community_bank
  if (length(treated_values)>0 && length(control_values)>0) {
    result <- diff_in_means(treated_values, control_values)
    print(sprintf('%s: %.3f±%.3f',chief, result[1], result[2]))
  }else
    print(sprintf('%s: insufficient data, only %d treatments and %d control values.', chief, length(tr
  }

## [1] "biriwa: 1.000±0.000"
## [1] "safroko limba: 0.667±0.142"
## [1] "sanda loko: 0.633±0.260"
## [1] "bombali shebora: 0.909±0.091"
## [1] "sanda tendaren: 0.200±0.200"
## [1] "sella limba: 0.333±0.279"
## [1] "gbendenbu ngowahun: 0.833±0.090"
## [1] "gbanti kamaranka: 0.833±0.167"
## [1] "makari gbanti: 0.667±0.211"
## [1] "bum: 0.636±0.168"
## [1] "imper: 0.600±0.245"
## [1] "jong: 0.735±0.148"
## [1] "nongoba bullom: 0.800±0.200"
## [1] "bothe town: 0.667±0.333"

```

Part B

Let's obtain a p-value under the sharp null of no treatment effect using randomization inference. We'll use the absolute value of the difference-in-means as our test statistic.

Assume $N_{t,g}$ is fixed for each block and that within each block, the researchers assigned treatment using complete randomization.

Approximate the randomization distribution of the absolute difference-in-means given the stratified randomization procedure (fixed $N_{t,g}$ for each stratum and complete randomization within each stratum). Use a simulation with 5000 total draws. Set your random seed to 10003 prior to the start of the simulation.

Make a histogram of your draws.

```
library(estimatr)
```

use function `difference_in_means` to sample the described distribution (chiefs as blocks/strata), $N = 5000$.

Part C

Calculate a p-value for the observed test statistic under the sharp null of no treatment effects using your randomization distribution from Part B. Interpret your results. Would you reject the null at the $\alpha = .05$ level?

use `diff_in_mean` analogous to Part A.
