hw3

1

```r
dgp_p1 <- function() {
  ### Define number of schools, and number of students in the school
  G <- 50 # number of schools
  ng <- 10 # number of students in each school
  n <- G*ng # total n-size

  ### Assign students to schools
  school <-rep(seq(1,G), ng)
  school <- sort(school)

  ### Gamma (school-varying intercept for PO's)
  gamma <- rnorm(G, sd=sqrt(2))[school]

  ### Generate potential outcomes
  y0 <- gamma + rnorm(n)
  y1 <- 2 + gamma + rnorm(n)

  ### Put everything into a data-frame
  data <- data.frame("school" = school, "y0" = y0, "y1" = y1)

  return(data)
}
```

a

There are 10 students per school and 50 schools total, meaning taht there are 500 total students. The ATE here is 2, and the factors that influence the potential outcomes are some random noise from the $\mathcal{N}(0,1)$ instances added to each one, as well as the school specific random intercept called 'gamma'.

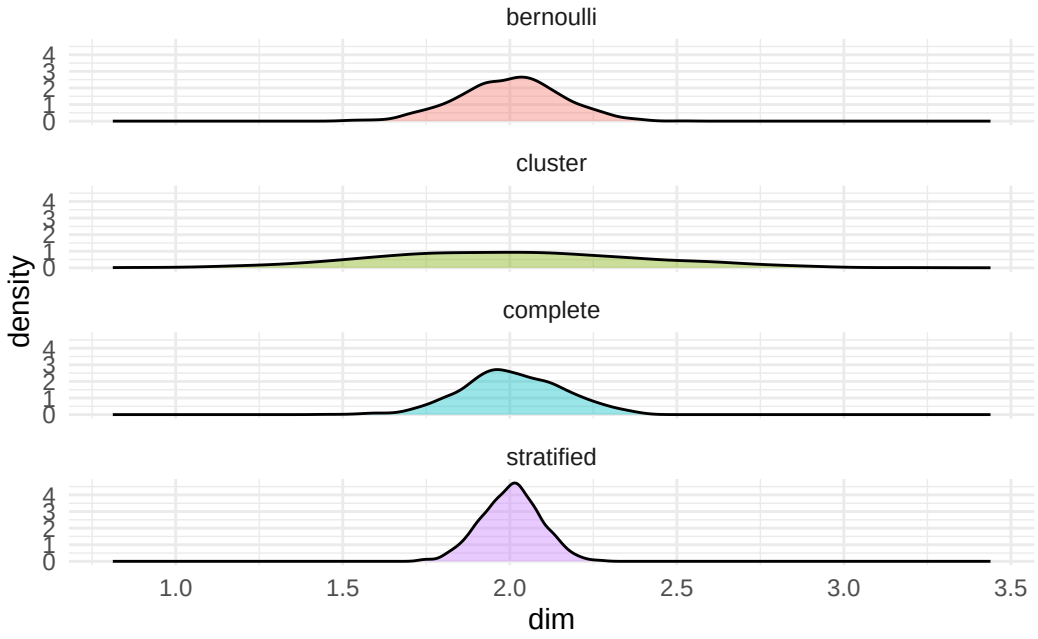
b

1) Complete: Randomly draw 250 students to be treated
2) Bernoulli: Probability of treatment for each student is 0.5
3) Cluster: Randomly draw 25 schools to be treated
4) Statified: Within each school, randomly draw 5 students to be treated

```r
# empty results df to fill
res <- tibble(
  complete = rep(0, 1000),
  bernoulli = rep(0, 1000),
  cluster = rep(0, 1000),
  stratified = rep(0, 1000)
)

for (i in 1:1000) {
  data <- dgp_p1()

  # complete
  row_ids <- sample(1:500, 250)
  complete <- data %>%
    mutate(
      id = row_number(),
      D = if_else(id %in% row_ids, 1, 0)
      ) %>%
    select(-id)

  # bernoulli
  bernoulli <- data %>%
    mutate(D = rbinom(500, 1, 0.5))

  # cluster
  school_ids <- sample(1:50, 25)
  cluster <- data %>%
    group_by(school) %>%
    mutate(
      group_id = cur_group_id(),
      D = if_else(group_id %in% school_ids, 1 , 0)
    ) %>%
    ungroup() %>%
    select(-group_id)
```

```r
  # stratified
  within_school_ids <- data %>%
    group_by(school) %>%
    mutate(chosen_id = row_number()) %>%
    slice_sample(n = 5) %>%
    ungroup() %>%
    select(school, chosen_id) %>%
    mutate(D = 1)

  stratified <- data %>%
    group_by(school) %>%
    mutate(
      id = row_number()
      ) %>%
    ungroup() %>%
    left_join(
      within_school_ids, by = c("id" = "chosen_id", "school")
      ) %>%
    mutate(D = replace_na(D, 0)) %>%
    select(-id)

  data_list <- list(complete, bernoulli, cluster, stratified)

  dim <- function(data){

    return(mean(data[data$D == 1, ]$y1) - mean(data[data$D == 0, ]$y0))

  }

  res[i, ] <- data_list %>%
    map(.f = dim)

}
```

```r
res %>%
  pivot_longer(cols = everything(), names_to = "sampling_mechanism", values_to = "dim") %>
  ggplot(aes(x = dim, fill = sampling_mechanism)) +
  geom_density(alpha = 0.4, show.legend = F) +
  facet_wrap(~ sampling_mechanism, ncol = 1) +
  theme_minimal()
```



2

a

```
data_raw <- read.dta13("/Users/joshuayamamoto/Downloads/BD-SAN-FINAL.dta")
with_names <- read_dta("/Users/joshuayamamoto/Downloads/BD-SAN-FINAL.dta")

data <- data_raw %>%
  filter(eligible == "Eligible") %>%
  select(r4_any_od_adults, treat_cat_3) %>%
  mutate(r4_recode = case_when(
    r4_any_od_adults == "Yes" ~ 1L,
    r4_any_od_adults == "No" ~ 0L
  )) %>%
  drop_na()

coef(summary(lm(r4_recode ~ treat_cat_3 , data)))[ -1,1] %>%
  as.data.frame() %>%
  rename('Estimate' = '.')
```

```
                          Estimate
treat_cat_3LPP Only      -0.07935631
```

```
treat_cat_3Supply Only   -0.09581019
treat_cat_3Loser (Low)   -0.09035256
treat_cat_3Loser (Med)   -0.15611116
treat_cat_3Loser (High)  -0.13388554
treat_cat_3Winner (Low)  -0.10971356
treat_cat_3Winner (Med)  -0.19489560
treat_cat_3Winner (High) -0.16496828
```

b

First, most policies and interventions are conducted at the community level, so this design provides policy-relevant information. Second, behavioral spillovers would likely contaminate an individual-level randomization, and in fact measuring these behavioral spillovers was an important objective of our study

c

```r
set.seed(46)

randomization_scheme <- function() {

  ## 1: Cluster randomization of villages to treatment conditions.
  ## note: stratify by number of neighborhoods per village (1-2 and 3+)

  ## Control, LPP Only, LPP + Subsidy, Supply only.

  strata_labels <- data_raw %>%
    distinct(vid, cid) %>%
    count(vid) %>%
    mutate(strata = case_when(
      n < 3 ~ "s1",
      n >= 3 ~ "s2"
    )) %>%
    select(-n)

  s1_villages <- strata_labels[strata_labels$strata == "s1", ]$vid
  s2_villages <- strata_labels[strata_labels$strata == "s2", ]$vid

  ## s1 has 50 villages
  ## s2 has 57 villages
```

```r
## 22 in control, 12 LPP only, 63 in LPP + Subsidy, 10 in Supply Only (following SM from

s1_clusters <- sample(
  c(rep("Control", 11), rep("LPP_Only", 6), rep("LPP+Subsidy", 29), rep("Supply_Only", 4
  size = 50, replace = F
  )

s2_clusters <- sample(
  c(rep("Control", 11), rep("LPP_Only", 6), rep("LPP+Subsidy", 34), rep("Supply_Only", 6
  size = 57, replace = F
  )

# assign clusters
cluster_labels <- tibble(
  vid = c(s1_villages, s2_villages),
  treat1 = c(s1_clusters, s2_clusters)
)

data_s1 <- data_raw %>%
  left_join(strata_labels, by = "vid") %>%
  left_join(cluster_labels, by = "vid")

## 2. Cluster randomization of neighborhoods in the "LPP + Subsidy" condition into a 2x3

## first cluster randomize into LPP + Subsidy and LPP + Subsidy + Supply

n_cid <- data_s1 %>%
  filter(treat1 == "LPP+Subsidy") %>%
  distinct(cid) %>%
  nrow()

# recover how many cid should be given LPP + Subsidy and LPP + Subsidy + Supply
if (n_cid %% 2 == 1) {
  n1 <- floor(n_cid/2)
  n2 <- floor(n_cid/2) + 1
} else {
  n1 <- n_cid/2
  n2 <- n_cid/2
}
```

```r
stage2_labels <- data_s1 %>%
  filter(treat1 == "LPP+Subsidy") %>%
  distinct(cid) %>%
  mutate(stage2grp = sample(c(rep("LPP+Subsidy", n1), rep("LPP+Subsidy+Supply", n2)), n1

p1 <- stage2_labels %>%
  filter(stage2grp == "LPP+Subsidy") %>%
  mutate(intensity = sample(c("High", "Medium", "Low"), n1, replace = T))


p2 <- stage2_labels %>%
  filter(stage2grp == "LPP+Subsidy+Supply") %>%
  mutate(intensity = sample(c("High", "Medium", "Low"), n2, replace = T))

stage2_labels_final <- rbind(p1, p2)

data_s2 <- data_s1 %>%
  left_join(stage2_labels_final, by = "cid") %>%
  mutate(treat1 = case_when(
    is.na(stage2grp) ~ treat1,
    T ~ stage2grp
  )) %>%
  select(-stage2grp) %>%
  mutate(treat2 = case_when(
    is.na(intensity) ~ treat1,
    T ~ intensity
  )) %>%
  select(-intensity)

## 3: Randomization of the vouchers for households in the "LPP + Subsidy" conditions, gi
## note: by household

# high- 60%
# medium- 50%
# low- 40%
# gotten from the table below
# data_raw %>%
#   group_by(treat_cat_3) %>%
#   summarise(n = n_distinct(hhid))

s2_1 <- data_s2 %>%
```

```
    filter(treat2 == "Low") %>%
    mutate(lottery = sample(c("Winner", "Loser"), n(), replace = T, prob = c(0.4, 0.6)))

  s2_2 <- data_s2 %>%
    filter(treat2 == "Medium") %>%
    mutate(lottery = sample(c("Winner", "Loser"), n(), replace = T, prob = c(0.5, 0.5)))

  s2_3 <- data_s2 %>%
    filter(treat2 == "High") %>%
    mutate(lottery = sample(c("Winner", "Loser"), n(), replace = T, prob = c(0.6, 0.4)))

  lottery_labels <- rbind(s2_1, s2_2, s2_3) %>%
    mutate(lottery = paste0(lottery, " (", treat2, ")")) %>%
    select(hhid, lottery)

  data_s3 <- data_s2 %>%
    left_join(lottery_labels, by = "hhid") %>%
    mutate(treat3 = case_when(
      is.na(lottery) ~ treat2,
      T ~ lottery
    )) %>%
    select(-lottery) %>%
    mutate(r4_recode = case_when(
      r4_any_od_adults == "Yes" ~ 1L,
      r4_any_od_adults == "No" ~ 0L
    )) %>%
    select(hhid, cid, vid, r4_recode, treat3)

  return(data_s3)

}
```

d

```
null_dist <- data.frame(dim = rep(0, 5000))

for(i in 1:5000) {
  iter <- randomization_scheme()
  iter <- iter %>%
    filter(treat3 == "Winner (Low)" | treat3 == "Control") %>%
```

```
    drop_na()

  null_dist[i, ] <- coef(lm(r4_recode ~ treat3, iter))[2]
}
```

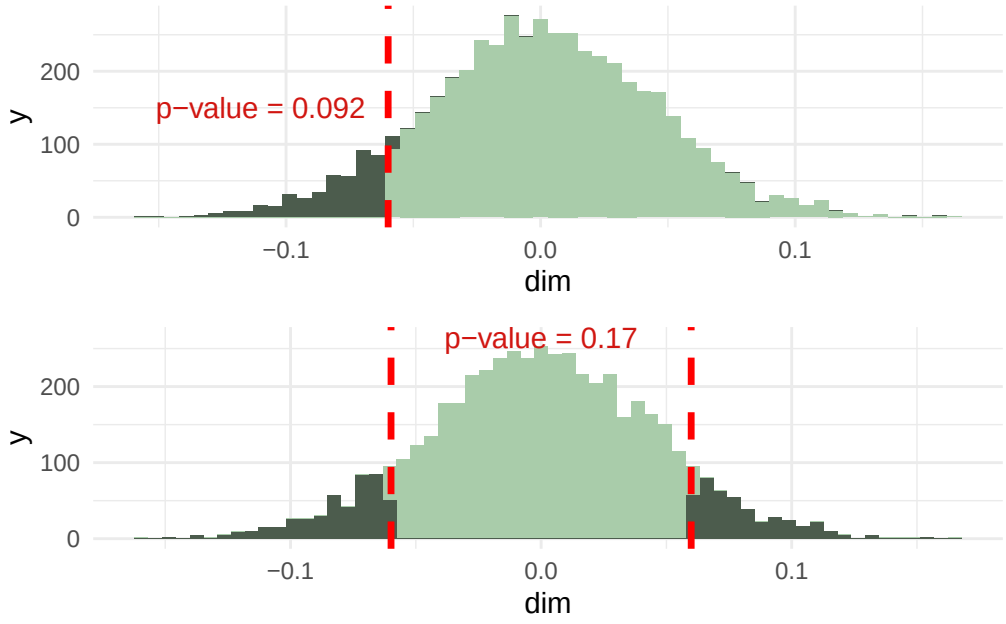
```
null_dist <- read_csv(here("data", "hw3_p2_null_dist.csv"))
```

```
Rows: 5000 Columns: 1
-- Column specification --------------------------------------------------------
Delimiter: ","
dbl (1): dim

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
data_sub <- data_raw %>%
  filter(treat_cat_3 == "Winner (Low)" | treat_cat_3 == "Control") %>%
   mutate(r4_recode = case_when(
      r4_any_od_adults == "Yes" ~ 1L,
      r4_any_od_adults == "No" ~ 0L
    )) %>%
  drop_na()

true_dim <- coef(lm(r4_recode ~ treat_cat_3, data_sub))[2]

p_val_one_sided <- mean(null_dist$dim < true_dim)
p_val_two_sided <- mean(null_dist$dim < -abs(true_dim)) + mean(null_dist$dim > abs(true_di

one_sided <- null_dist %>%
  mutate(filler = ifelse(dim > true_dim, "more", "less")) %>%
  ggplot(aes(x = dim, fill = filler)) +
  geom_histogram(binwidth = 0.0058, show.legend = F) +
  scale_fill_manual(values = c("#4c5c4d", "#a9ccaa")) +
  geom_vline(xintercept = true_dim, color = "red", linetype = "dashed", linewidth = 1.2) +
  theme_minimal() +
  annotate(geom = "text", x = -0.11, y = 150, label = "p-value = 0.092", color = "#d11611"



two_sided <- null_dist %>%
```

```
  mutate(filler = ifelse(dim < -abs(true_dim) | dim > abs(true_dim), "shade", "no-shade"))
  ggplot(aes(x = dim, fill = filler)) +
  geom_histogram(binwidth = 0.0055, show.legend = F) +
  geom_vline(xintercept = -abs(true_dim), color = "red", linetype = "dashed", linewidth =
  geom_vline(xintercept = abs(true_dim), color = "red", linetype = "dashed", linewidth = 1
  scale_fill_manual(values = c("#a9ccaa", "#4c5c4d")) +
  theme_minimal() +
  annotate(geom = "text", x = 0, y = 265, label = "p-value = 0.17", color = "#d11611" )
```

```
one_sided / two_sided
```



3

a

$$
\begin{aligned}
-B_n - E_n &= -\left(\frac{1}{n_1}\sum_{D_i=1} f_1(X_i) - \frac{1}{n}\sum_{i=1}^{n} f_1(X_i)\right) + \left(\frac{1}{n_0}\sum_{D_i=0} f_0(X_i) - \frac{1}{n}\sum_{i=1}^{n} f_0(X_i)\right) - \\
&\quad \left(\frac{1}{n_1}\sum_{D_i=1} \varepsilon_i(1) - \frac{1}{n}\sum_{i=1}^{n} \varepsilon_i(1)\right) + \left(\frac{1}{n_0}\sum_{D_i=0} \varepsilon_i(0) - \frac{1}{n}\sum_{i=1}^{n} \varepsilon_i(0)\right) \\
&= -\frac{1}{n_1}\sum_{D_i=1} f_1(X_i) + \frac{1}{n_0}\sum_{D_i=0} f_0(X_i) - \frac{1}{n_1}\sum_{D_i=1} \varepsilon_i(1) + \frac{1}{n_0}\sum_{D_i=0} \varepsilon_i(0) + \text{SATE} \\
&= -\frac{1}{n_1}\sum_{D_i=1} \left[f_1(X_i) + \varepsilon_i(1)\right] + \frac{1}{n_0}\sum_{D_i=0} \left[f_0(X_i) + \varepsilon_i(0)\right] + \text{SATE} \\
&= -\frac{1}{n_1}\sum_{D_i=1} Y_i(1) + \frac{1}{n_0}\sum_{D_i=0} Y_i(0) + \text{SATE}
\end{aligned}
$$

so now we can say that

$$
\begin{aligned}
\hat{\tau}_{dim} - B_n - E_n &= \frac{1}{n_1}\sum_{D_i=1} Y_i(1) - \frac{1}{n_0}\sum_{D_i=0} Y_i(0) - \frac{1}{n_1}\sum_{D_i=1} Y_i(1) + \frac{1}{n_0}\sum_{D_i=0} Y_i(0) + \text{SATE} \\
&= \text{SATE}
\end{aligned}
$$

b

First recall that we can express $\hat{\tau}_{ols}$ as follows by regressing $(Y - X^T\hat{\beta}_{ols})$ on D. If we simplify the expression down we get the following:

$$
\begin{aligned}
\hat{\tau}_{ols} &= \left(\frac{1}{n_1}\sum_{D_i=1} Y_i - \frac{1}{n_0}\sum_{D_i=1} Y_i\right) - \left(\frac{1}{n_1}\sum_{D_i=1} X_i^T\hat{\beta}_{ols} - \frac{1}{n_0}\sum_{D_i=0} X_i^T\hat{\beta}_{ols}\right) \\
&= \hat{\tau}_{dim} - \left(\frac{1}{n_1}\sum_{D_i=1} X_i^T\hat{\beta}_{ols} - \frac{1}{n_0}\sum_{D_i=0} X_i^T\hat{\beta}_{ols}\right)
\end{aligned}
$$

Next we can simplify the term $-\hat{\beta}_{n,ols}$ as follows:

$$
\begin{aligned}
-\hat{\beta}_{n,ols} &= -\left(\frac{n_1\hat{\tau}_{ols}}{n_1} + \frac{1}{n_1}\sum_{D_i=1} X_i^T\hat{\beta}_{ols} - \frac{n\hat{\tau}_{ols}}{n} - \frac{1}{n}\sum_{i=1}^{n} X_i^T\hat{\beta}_{ols}\right) + \left(\frac{1}{n_0}\sum_{D_i=0} X_i^T\hat{\beta}_{ols} - \frac{1}{n}\sum_{i=1}^{n} X_i^T\hat{\beta}_{ols}\right) \\
&= -\left(\frac{1}{n_1}\sum_{D_i=1} X_i^T\hat{\beta}_{ols} - \frac{1}{n}\sum_{i=1}^{n} X_i^T\hat{\beta}_{ols}\right) + \left(\frac{1}{n_0}\sum_{D_i=0} X_i^T\hat{\beta}_{ols} - \frac{1}{n}\sum_{i=1}^{n} X_i^T\hat{\beta}_{ols}\right) \\
&= -\frac{1}{n_1}\sum_{D_i=1} X_i^T\hat{\beta}_{ols} + \frac{1}{n_0}\sum_{D_i=0} X_i^T\hat{\beta}_{ols} \\
&= -\left(\frac{1}{n_1}\sum_{D_i=1} X_i^T\hat{\beta}_{ols} - \frac{1}{n_0}\sum_{D_i=0} X_i^T\hat{\beta}_{ols}\right)
\end{aligned}
$$

Finally if we take $\hat{\tau}_{dim} - \hat{\beta}_{n,ols}$ along with the first equality that we recovered, we get

$$
\begin{aligned}
\hat{\tau}_{dim} - \hat{\beta}_{n,ols} &= \hat{\tau}_{dim} - \left(\frac{1}{n_1}\sum_{D_i=1} X_i^T\hat{\beta}_{ols} - \frac{1}{n_0}\sum_{D_i=0} X_i^T\hat{\beta}_{ols}\right) \\
&= \hat{\tau}_{ols}
\end{aligned}
$$

as desired

c