Estimation of Causal Effects

Jake Bowers, Ben Hansen & Tom Leavitt

August 4, 2016

Contents

1 Estimation of Mean Unit-Level Causal Effects

```
## Load data
news_df <- read.csv("http://jakebowers.org/PS531Data/news.df.csv")

## Create potential outcomes
news_df %<>% rename(y = r) %>%
  mutate(yc = ifelse(test = z == 0,
                     yes = y,
                     no = NA),
         yt = ifelse(test = z == 1,
                     yes = y,
                     no = NA))

kable(news_df[, c(1, 3:4, 8:9)])
```

city	y	z	yc	yt
Saginaw	16	0	16	NA
Sioux City	22	1	NA	22
Battle Creek	14	0	14	NA
Midland	7	1	NA	7
Oxford	23	0	23	NA
Lowell	27	1	NA	27
Yakima	58	0	58	NA
Richland	61	1	NA	61

Now let's imagine the *true* unit-level treatment effect is some specific positive natural number that differs for almost every unit. What do we mean by "*true* unit-level treatment effect" as opposed to a unit-level treatment effect that we hypothesize and subsequently assume to be true?

```
news_df %<>% mutate(true_tau = c(6, 4, 19, 12, 9, 9, 13, 15),
                    yc = ifelse(test = is.na(yc),
                                yes = yt - true_tau,
                                no = yc),
                    yt = ifelse(test = is.na(yt),
                                yes = yc + true_tau,
                                no = yt))

true_ate <- news_df %$% mean(true_tau)

kable(news_df[, c(1, 3:4, 8:9)])
```

city	y	z	yc	yt
Saginaw	16	0	16	22
Sioux City	22	1	18	22
Battle Creek	14	0	14	33
Midland	7	1	-5	7
Oxford	23	0	23	32
Lowell	27	1	18	27
Yakima	58	0	58	71
Richland	61	1	46	61

We have an experiment that consists of complete random assignment within pairs. How many treatment assignment permutations are in the set Ω? How many elements would there be in Ω if the experiment consisted of complete random assignment *without* blocked pairs? What is the probability associated with each $\mathbf{Z} = \mathbf{z} \in \Omega$?

```r
treatment_permutations <- function(z, yc, yt, s){

    if(missing(s)){

      ## Permute treatment assignment
      ## Student Question: What is the difference between Z and z? Why is one uppercase and the other lowe
      Z = sample(z)

      Y = Z * yt + (1 - Z) * yc

      ## Calculate unstratified test statistic
      ate_hat_unstrat = coef(lm(Y ~ Z))[["Z"]]

      return(ate_hat_unstrat)

    }

  else {

    ## Permute treatment assignment WITHIN blocks
    Z = unsplit(lapply(split(x = z, f = s), sample), s)

    Y = Z * yt + (1 - Z) * yc

    ## Calculate stratified test-statistic
    ate_hat_strat = coef(lm(Y ~ Z + s))[["Z"]]

    return(ate_hat_strat)
  }
  }
```

Describe in words what the function above is doing.

```
## Set seed for simulations
set.seed(1:5)

obs_block_ate_hat <- coef(lm(y ~ z + s, data = news_df))[["z"]]

obs_ate_hat <- coef(lm(y ~ z, data = news_df))[["z"]]

all.equal(obs_block_ate_hat, obs_ate_hat)
```

```
## [1] TRUE
```

```
## If we use blocks
block_randomization_distribution <- data.frame(ate = replicate(10^3,
                                                               treatment_permutations(z = news_df$z,
                                                                                      yc = news_df$yc,
                                                                                      yt = news_df$yt,
                                                                                      s = news_df$s)))

colMeans(block_randomization_distribution)
```

```
##     ate
## 10.7115
```

```
## If we don't use blocks
randomization_distribution <- data.frame(ate = replicate(10^3,
                                                         treatment_permutations(z = news_df$z,
                                                                                yc = news_df$yc,
                                                                                yt = news_df$yt)))

colMeans(randomization_distribution)
```

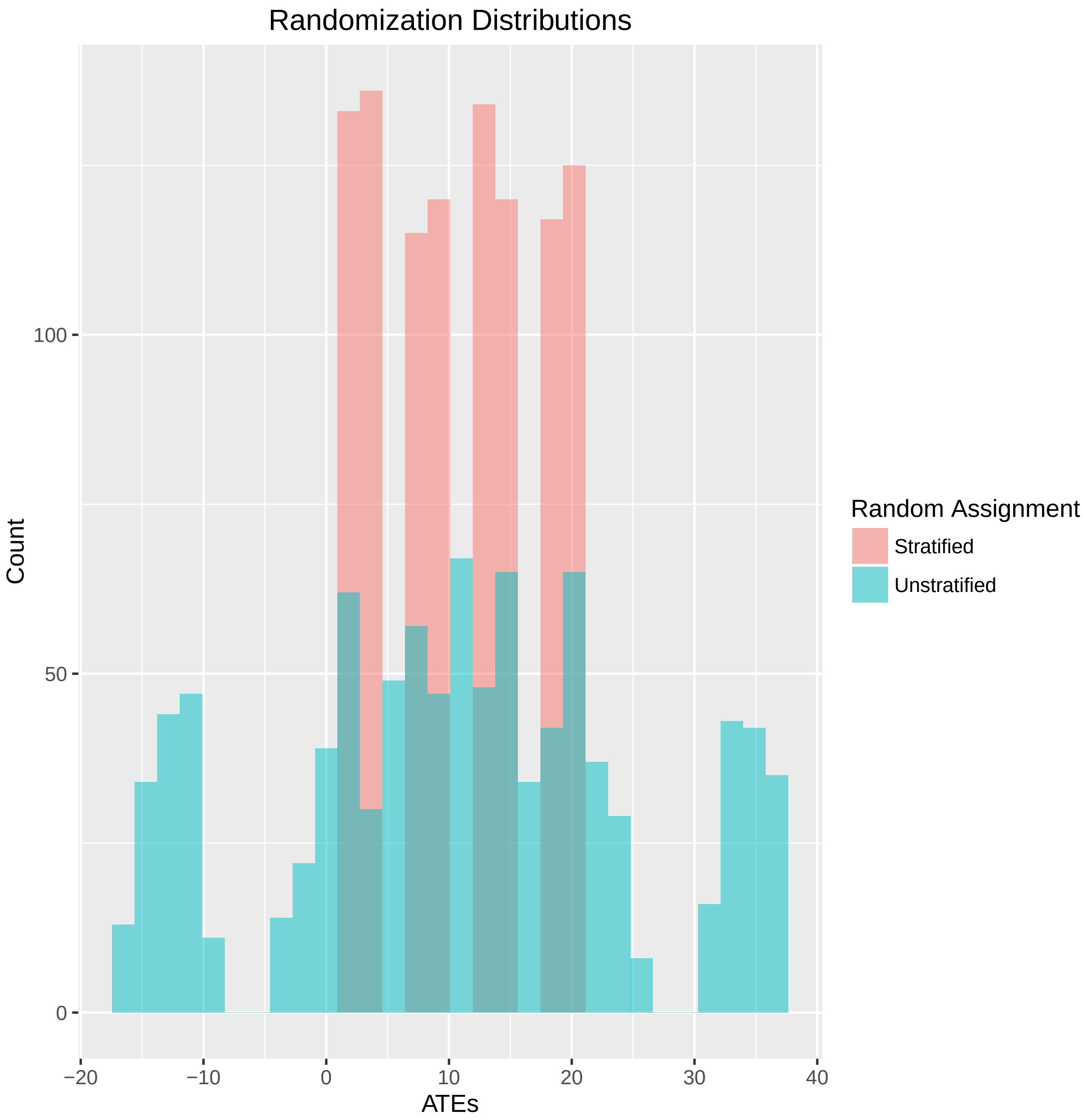
```
##      ate
## 10.64175
```

```
block_randomization_distribution$type <- 'Stratified'

randomization_distribution$type <- 'Unstratified'
```

```r
randomization_dists <- as.data.frame(rbind(block_randomization_distribution, randomization_distribution))

ggplot(randomization_dists, aes(x = ate, fill = type)) +
  geom_histogram(alpha = 0.5, aes(), position = 'identity') +
  xlab("ATEs") +
  ylab("Count") +
  ggtitle("Randomization Distributions") +
  scale_fill_discrete(name = "Random Assignment")
```



```
## If you want to add the observed ate hat
# +  geom_vline(xintercept = obs_ate_hat, colour = "black", linetype = "longdash")
```

Notice that the randomization distributions looks a little strange; that is, both randomization distributions are very different from the normal distribution to which we are accustomed. This is important. Nothing so far has invoked any claim about the shape of the randomization distribution, such as its normality if the finite population central limit theorem obtains (Hájek, 1960). What matters (in terms of unbiasedness) is whether the mean of the randomization distribution—i.e., the mean of all the estimated difference in means—is equal to the true average treatment effect.

Is the variance of the randomization distribution tighter when treatment is assigned within blocks or when treatment is *not* assigned within blocks?

Our *single* estimated $\widehat{ATE}$ from the experiment is our "best guess" about the true ATE. But what do we mean by "best guess"? And can one's "best guess" be misleading?

How would you relate this [R] exercise to the analytic proof we did in class?

2 Binary Outcomes

Now let's pretend that we have an experiment with a binary outcome:

Freedman (2008) famously argues that "randomization does not justify logistic regression." Let's recall that the coefficient of a logistic regression model is the difference in potential log odds.

```
experiment <- data.frame(unit = seq(from = 1, to = 8, by = 1))

experiment %<>% mutate(Z = complete_ra(N = length(unit))) %>%
  arrange(desc(Z))

experiment %<>% mutate(y1 = c(1, 1, 0, 1, 1, 0, 0, 1),
                       y0 = c(1, 1, 0, 1, 0, 0, 1, 0),
                       Y = Z * y1 + (1 - Z) * y0)

N <- length(experiment$Z)

Z <- experiment$Z

m <- sum(experiment$Z == 1)
```

```r
y1 <- experiment$y1

y0 <- experiment$y0

Y <- experiment$Y

true_ate <- mean(y1) - mean(y0)

true_logit <- log(mean(y1)/(1 - mean(y1))) - log(mean(y0)/(1 - mean(y0)))

est_ate <- mean(Y[Z == 1]) - mean(Y[Z == 0])

est_logit <- log(mean(Y[Z == 1])/(1 - mean(Y[Z == 1]))) -
  log(mean(Y[Z == 0])/(1 - mean(Y[Z == 0])))

new_experiment <- function(z, y0, y1){

  Z = sample(z)

  Y = Z * y1 + (1 - Z) * y0

  est_ate = coef(lm(Y ~ Z))[["Z"]]

  est_logit = coef(glm(Y ~ Z,
                  family = binomial(link = "logit")))[["Z"]]

  return(c(est_ate, est_logit))
}

set.seed(1:5)

randomization_dists <- data.frame(t(replicate(10^3, new_experiment(z = Z,
                                                           y0 = y0,
                                                           y1 = y1)))) %>%
  rename(est_ates = X1,
         est_logits = X2)

colMeans(randomization_dists)


##   est_ates est_logits
##   0.133750   1.942527
```

Is the difference-in-means estimator still unbiased (give or take some simulation error) when potential outcomes are binary? Is the coefficient from a logistic regression model unbiased? To which quantities (defined as `[R]` objects above) are we comparing the means of the two respective randomization distributions in order to assess unbiasedness?

How does the randomization distribution above differ from the *null* randomization distribution?

In a code chunk below, plot both randomization distributions (of the estimated ATEs and estimated logit coefficients) in `[R]`:

3 Application of Unbiased Inference to Matched Designs

Lets's load a matched design that we created with the data from the Cerdá et al. (2012) paper.

```
rm(list=ls())

load(url("http://jakebowers.org/Matching/meddat.rda"))

meddat <- transform(meddat, HomRate03=(HomCount2003/Pop2003)*1000)
meddat <- transform(meddat, HomRate08=(HomCount2008/Pop2008)*1000)

## Scalar distance on baseline outcome
tmp <- meddat$HomRate03
names(tmp) <- rownames(meddat)
absDist <- match_on(tmp, z = meddat$nhTrt)

balfmla <- reformulate(c(names(meddat)[c(6:7,9:24)],"HomRate03"),response="nhTrt")

library(brglm)
brglm1 <- brglm(balfmla,data=meddat,family=binomial)
pScore2 <- predict(brglm1)
meddat$pScore2 <- pScore2
psDist2 <- match_on(nhTrt~pScore2,data=meddat)


mahalDist <- match_on(balfmla,data=meddat,method="rank_mahalanobis")

fm3 <- fullmatch(psDist2+caliper(psDist2,4)
                +caliper(absDist,2)
                +caliper(mahalDist,40),data=meddat,tol=.00001,min.controls=1)

summary(fm3)


## Structure of matched sets:
```

```
## 1:0 1:1 1:3 0:1
##   2  19   1   1
## Effective Sample Size:  20.5
## (equivalent number of matched pairs).
```

```
meddat[names(fm3),"fm3"] <- fm3

meddat %<>% filter(!is.na(fm3)) %>% arrange(desc(fm3)) %>% mutate(fm3 = as.numeric(fm3))
```

3.1 Harmonic Mean Weighting

```
obs_ate <- coef(lm(HomRate08 ~ nhTrt + as.factor(fm3), data = meddat))[["nhTrt"]]

blocks <- unique(meddat$fm3)

ATE_by_block <- sapply(blocks, function(x) {
    coef(lm(HomRate08 ~ nhTrt, data = meddat, subset = fm3 == x))[["nhTrt"]]
})

num_treated <- meddat %$% {sapply(split(nhTrt, fm3), sum)}
num_control <- meddat %$% {sapply(split(1 - nhTrt, fm3), sum)}

harm_mean <- 2/(1/num_treated + 1/num_control)

ATE <- sum(ATE_by_block * (harm_mean/sum(harm_mean)))
```

3.2 Block-Size Weighting

```
ATE_by_block <- sapply(blocks, function(x) {
    coef(lm(HomRate08 ~ nhTrt, data = meddat, subset = fm3 == x))[["nhTrt"]]
})

num_units_per_block <- sapply(blocks, function(x){
  length(meddat$nhTrt[meddat$fm3 == x])
})

overall_est_ATE <- sum(ATE_by_block * (num_units_per_block/nrow(meddat)))

block_size_ATE_weighting <- function(z, Y, s){
  # Permute treatment assignment
  Z <- unsplit(lapply(split(z, s), sample), s)

  # Average Treatment Effect mean(Y[Z_New == 1]) - mean(Y[Z_New == 0])
  # ATE <- coef(lm(Y ~ Z_New + B))[["Z_New"]]
  W_ATE <- 0

  for (i in unique(s)) {
    W_ATE <- W_ATE + (sum(s == i)/length(s)) *
      coef(lm(Y ~ Z, subset = (s == i)))[2]
    }

  return(W_ATE)
  }

null_dist_block_size_weighting <- data.frame(replicate(10^3,
                                            block_size_ATE_weighting(z = meddat$nhTrt,
                                                                     Y =  meddat$HomRate08,
                                                                     s = meddat$fm3)))

p_value <- mean(abs(overall_est_ATE) >= abs(null_dist_block_size_weighting))
```

3.3 ETT Weighting

```
ett_ATE_weighting <- function(z, Y, s){

  # Permute treatment assignment
  Z <- unsplit(lapply(split(z, s), sample), s)

  ett_ATE <- 0

  for (i in unique(s)) {
    ett_ATE <- ett_ATE + (sum(s == i[Z == 1])/length(s[Z == 1])) *
      coef(lm(Y ~ Z, subset = (s == i)))[2]
    }

  return(ett_ATE)
  }
```

References

Cerdá, M., J. D. Morenoff, B. B. Hansen, K. J. T. Hicks, L. F. Duque, A. Restrepo, and A. V. Diez-Roux (2012). Reducing violence by transforming neighborhoods: A natural experiment in medellín, colombia. *American Journal of Epidemiology 175*(10), 1045–1053. 8

Freedman, D. A. (2008). Randomization does not justify logistic regression. *Statistical Science 23*(2), 237–249. 6

Hájek, J. (1960). Limiting distributions in simple random sampling from a finite population. *Publications of the Mathematics Institute of the Hungarian Academy of Science 5*, 361–374. 6