# Lecture 7

Alex Stephenson

9-10-2021

# Blocking

Blocking helps reduce sampling variability. In practice, block on what you can and randomize on what you cannot.

The total number of random allocations in a design with B blocks is always less than the total number of allocations under complete randomization.

Example N = 20, m = 10. Under complete randomization there are $\frac{20!}{10!10!} = 184,756$.

Under blocking with complete randomization with 2 equal sized blocks. $\frac{10!}{5!5!}\frac{10!}{5!5!} = 63,504$

Blocking also guarantees that a certain subgroup will be available for analysis. E.g. we guarantee that there are 10 Democrats and 10 Republicans in each block.

# Block ATE

$$ATE = \sum_{i}^{J} \frac{N_j}{N} ATE_j$$

$$ATE_j = E[Y_{ij}(1)] - E[Y_iij(0)]$$

$ATE_j$ is the within block j ATE.

J is total number of blocks.

$\frac{N_j}{N}$ is the weighted share of all units who are in block j

# Block ATE in R

```r
within_block_ate <- function(df, y0, y1, block, block_weights){
  # Block weights are the number of observations in each
  # block
  treat <- mean(df$y1[df$block==block], na.rm = T)
  control <- mean(df$y0[df$block==block], na.rm = T)

  return((treat-control)*block_weights)
}
```

# Block Conservative SE

For any number of blocks

$$\sigma_{\hat{ATE}} = \sqrt{\sum_{i}^{J}(\frac{N_j}{N})^2\sigma^2_{\hat{ATE_j}}}$$

For two blocks this is:

$$\sigma_{\hat{ATE}} = \sqrt{(\frac{N_1}{N})^2\sigma^2_{\hat{1}} + (\frac{N_2}{N})^2\sigma^2_{\hat{2}}}$$

# Block SE in R

```r
within_block_se <- function(df, y0, y1, block, N, m){
  # formula for se = sqrt(V[X]/N-m + V[Y]/m)
  # We need to get this within each block so subset appropriately
  control <- var(df$y0[df$block==block], na.rm = T)/(N-m)
  treat <-var(df$y1[df$block==block], na.rm = T)/(m)
  return(sqrt(treat+control))
}
```

# Block Confidence Intervals

Confidence Intervals for Blocking function like confidence intervals for non-blocked assignments.

Provided we have correctly estimated the ATE and SE, we proceed the same way as before.

# Block Confidence Intervals in R

```r
get_block_ci95 <- function(block_ate, block_se){
  # get 95% CI
  # 1.96 is the normal approximation value
  ci_l <- block_ate - block_se*1.96
  ci_u <- block_ate + block_se*1.96

  return(c(ci_l, ci_u))
}
```

# Block Example

Suppose we have the following data frame

```r
d <- tibble(
  block = c(rep("A", 8), rep("B",6)),
  y0 = c(0,1, NA, 4,4,6,6,NA, 14, NA, 16,16,17, NA),
  y1 = c(NA, NA, 1, NA, NA, NA, NA, 3, NA, 9, NA,NA,NA, 17)
)
```

# Block Example

We need to get the overall sum of each within block ATE

```
get_block_ate <- function(df, y0, y1, block,block_weights){
  val <- NULL
  for(i in 1:length(block)){
    val[i]<-within_block_ate(df, y0, y1, block[i],
                             block_weights[i])
  }
  return(sum(val))
}
```

# Block Example

We need to get the overall Block SE

```
get_block_se <- function(df, y0, y1, block, N, m, block_weights){
  # variance of sum of independent random variables
  # V[aX + bY] = a^2V[X]+b^2V[Y]

  # for loop to get sum of weighted variances
  val <- NULL
  for(i in 1:length(block)){
    val[i] <- within_block_se(df, y0, y1,
                              block[i], N[i],
                              m[i])^2*block_weights[i]^2
  }

  # sqrt of the sum to get se
  return(sqrt(sum(val)))
}
```

# Block Example

Use `get_block_ate()` with appropriate parameters to get: $\hat{\mu}$

```r
ate_hat <-get_block_ate(d, y0="y0",y1="y1",
                      block=c("A","B"),
                      block_weights = c(sum(d$block=="A")/nrow(d),
                                        sum(d$block=="B")/nrow(d)))
kable(ate_hat)
```

| x |
|---:|
| -2.035714 |

# Block Example

Use `get_block_se()` with appropriate parameters to get: $\hat{\sigma}_{ATE}$

```r
se_hat <- get_block_se(df = d,
                   y0 = "y0",
                   y1 = "y1",
                   block = c("A","B"),
                   N = c(8,6),
                   m = c(2,2),
                   block_weights = c(sum(d$block=="A")/nrow(d),
                                     sum(d$block=="B")/nrow(d)))
kable(se_hat)
```

| x |
|---:|
| 1.918559 |

# Block Example

Use `get_block_ci95()` with appropriate parameters to get our 95% CI

```
block_ci <- get_block_ci95(ate_hat, se_hat)

kable(tibble(
  interval = c("lower", "upper"),value = block_ci))
```

| interval | value |
|---|---:|
| lower | -5.796089 |
| upper | 1.724661 |

# Clustered Designs

Situations in which underlying potential outcomes are related. Examples include schools, villages, towns, states.

There may be large N within cluster, but because of relations we have to analyze at the cluster level.

Each unit in a cluster is placed into either treatment or control conditions

# Clustered Designs ATE

If clusters are the same size, then our estimate of the ATE via difference in means will be unbiased.

If clusters are not the same size, then our estimate of the ATE must take this into account. Naive estimates will be biased.

Clustered designs will have more variability (uncertainty) than non clustered designs.

We do clustered designs because we have to, not because we want to.

# Cluster Example

```r
cdf <- tibble(
  cluster = c(rep("Berkeley",3),
              rep("Stanford",3),
              rep("UCLA",3),
              rep("UCSD",3)),
  dorm = c(rep(c(1,2,3),4)),
  y0 = c(0:11),
  y1 = y0 + 4
)
```

| cluster | dorm | y0 | y1 |
|---|---|---|---|
| Berkeley | 1 | 0 | 4 |
| Berkeley | 2 | 1 | 5 |
| Berkeley | 3 | 2 | 6 |
| Stanford | 1 | 2 | 6 |
| Stanford | 2 | 3 | 7 |
| Stanford | 3 | 4 | 8 |
| UCLA | 1 | 3 | 7 |
| UCLA | 2 | 4 | 8 |
| UCLA | 3 | 5 | 9 |

# Cluster ATE and Conservative SE

The Cluster ATE formula with equal sized clusters

$$\hat{\mu}_{DM} = \left[ \frac{\sum_1^T \sum_1^n Y_{it}}{\sum_i^J n_t} - \frac{\sum_1^C \sum_1^N Y_{ic}}{\sum_1^C n_c} \right]$$

The Cluster Difference in Totals formula

$$\hat{\mu} = \frac{k_c + k_t}{N} \left( \frac{\sum Y_i(1) | d_i = 1}{k_t} - \frac{\sum Y_i(0) | d_i = 0}{k_c} \right)$$

The Cluster SE Formula

$$\hat{\sigma}_{A\hat{T}E} = \sqrt{\frac{N\hat{V}[\bar{Y}_j(0)]}{k(N-m)} + \frac{N\hat{V}[\bar{Y}_j(1)]}{km}}$$

# Cluster ATE with Equal Clusters

```
assignment_vectors <- list(
  c(rep(0,3), rep(0,3),
    rep(1,3), rep(1,3)),
  c(rep(0,3), rep(1,3),
    rep(0,3), rep(1,3)),
  c(rep(1,3), rep(0,3),
    rep(0,3), rep(1,3)),
  c(rep(1,3), rep(1,3),
    rep(0,3), rep(0,3)),
  c(rep(0,3), rep(1,3),
    rep(1,3), rep(0,3)),
  c(rep(1,3), rep(0,3),
    rep(1,3), rep(0,3))

)
```

Running over all possible assignments, the ATE will be equal to 4

| x |
|---|
| 4 |

# Next Week

Regression