

w241: Experiments and Causality

Blocking and Clustering

David Reiley, David Broockman, D. Alex Hughes

UC Berkeley, School of Information

Updated: 2021-09-07

Blocking

- Hard to know if it was due to chance when there are large differences between treatment and control.
- Need to reduce the size of the differences that can arise by chance.
- Increase statistical power given an experiment with same sample and effect size.
- If some variables are related to the outcome, restrict ourselves to randomizations that keep treatment and control similar.

Make Data

```
d ← make_data(effect_size=0)
head(d)
```

```
##      Control Treatment group
## 1:         1          1   Man
## 2:         2          2   Man
## 3:         3          3   Man
## 4:         4          4   Man
## 5:         5          5   Man
## 6:         6          6   Man
```

Randomization

```
d[, assignment := randomize(size=40)][ ,  
  table('Sex' = group, 'Assignment' = assignment)]
```

```
##           Assignment  
## Sex      Control Treatment  
##   Man         10         10  
##   Woman        10         10
```

```
d[, assignment := randomize(size=40)][ ,  
  table('Sex' = group, 'Assignment' = assignment)]
```

```
##           Assignment  
## Sex      Control Treatment  
##   Man         10         10  
##   Woman        10         10
```

Block Randomization

```
block_randomize ← function(size) {  
  ## this function will be executed /within/ the data.table that  
  ## holds the data. It could be run outside, but the assignment  
  ## in place that data.table provides make it clean inside.  
  conditions ← c('Control', 'Treatment')  
  
  if(size %% 2 == 0) {  
    ## if there are an even number of units in each block this is easy  
    urn ← rep(conditions, times = size/2)  
  } else if(size %% 2 == 1) {  
    ## if there are an odd number, then produce conditions to the  
    ## nearest even number that is less than the number of units  
    ## then add one more assignment condition, sampled at random  
    urn ← c(rep(conditions, times = (size/2) - 0.5), sample(conditions, size = 1))  
  }  
  
  ## now, shuffle it up return the shuffled sequence  
  assignment ← sample(urn)  
  return(assignment)  
}
```

Randomization

```
d[ , block_assignment := block_randomize(size=.N), by = group][ ,  
  table('Sex' = group, 'Assignment' = block_assignment)]
```

```
##           Assignment  
## Sex      Control Treatment  
##   Man         10         10  
##   Woman        10         10
```

```
d[ , block_assignment := block_randomize(size=.N), by = group][ ,  
  table('Sex' = group, 'Assignment' = block_assignment)]
```

```
##           Assignment  
## Sex      Control Treatment  
##   Man         10         10  
##   Woman        10         10
```

Conduct Experiment

```
conduct_experiment ← function(potential_control, potential_treatment, assignment) {  
  outcomes ← potential_treatment * I(assignment == "Treatment") +  
    potential_control * I(assignment == "Control")  
  
  return(outcomes)  
}
```

```
d[, Y := conduct_experiment(Control, Treatment, block_assignment)]
```

```
head(d)
```

```
##      Control Treatment group assignment block_assignment Y  
## 1:         1         1   Man  Treatment      Treatment 1  
## 2:         2         2   Man   Control      Control 2  
## 3:         3         3   Man  Treatment      Treatment 3  
## 4:         4         4   Man  Treatment      Control 4  
## 5:         5         5   Man   Control      Treatment 5  
## 6:         6         6   Man  Treatment      Treatment 6
```


Estimate ATE

```
estimate_ate <- function(y_values, treatment, verbose=FALSE) {  
  
  treatment_group_mean <- mean(y_values[treatment == 'Treatment'])  
  control_group_mean   <- mean(y_values[treatment == 'Control'])  
  
  ate <- treatment_group_mean - control_group_mean  
  
  if(verbose) {  
    return(  
      list(  
        "tg_mean" = treatment_group_mean,  
        "cg_mean" = control_group_mean,  
        "ate" = ate))  
    } else {  
      return("ate" = ate)  
    }  
}  
  
ate <- d[, estimate_ate(y_values = Y, treatment = block_assignment, verbose=TRUE)]  
ate  
  
##      tg_mean cg_mean  ate  
## 1:      34.2      36.8 -2.6
```

Simulate A Normal Study

```
simulate_normal_study <- function(effect_size) {  
  ## create world  
  d <- make_data(effect_size=effect_size)  
  
  ## randomly assign and count the number of women in treatment  
  d[, assignment := randomize()]  
  
  women_in_treatment <- d[group = 'Woman' & assignment = 'Treatment', .N]  
  
  ## measure outcomes  
  d[, Y := conduct_experiment(Control, Treatment, assignment)]  
  
  ## estimate ate  
  ate <- d[, estimate_ate(y_values = Y, treatment = assignment)]  
  
  ## return objects  
  ## - `ate` from the `estimate_ate` function.  
  ## - `women_in_treatment` as a count  
  return(list('ate' = ate, 'women_in_treatment' = women_in_treatment))  
}
```

Run One Normal Study

```
normal_study ← simulate_normal_study(effect_size = 0)  
normal_study
```

```
## $ate  
## [1] -6.4  
##  
## $women_in_treatment  
## [1] 9
```

Simulate Many Normal Studies

```
many_normal_studies <- replicate(  
  n = 1000,  
  expr = simulate_normal_study(effect_size = 10))
```

```
many_normal_studies <- t(many_normal_studies)  
head(many_normal_studies)
```

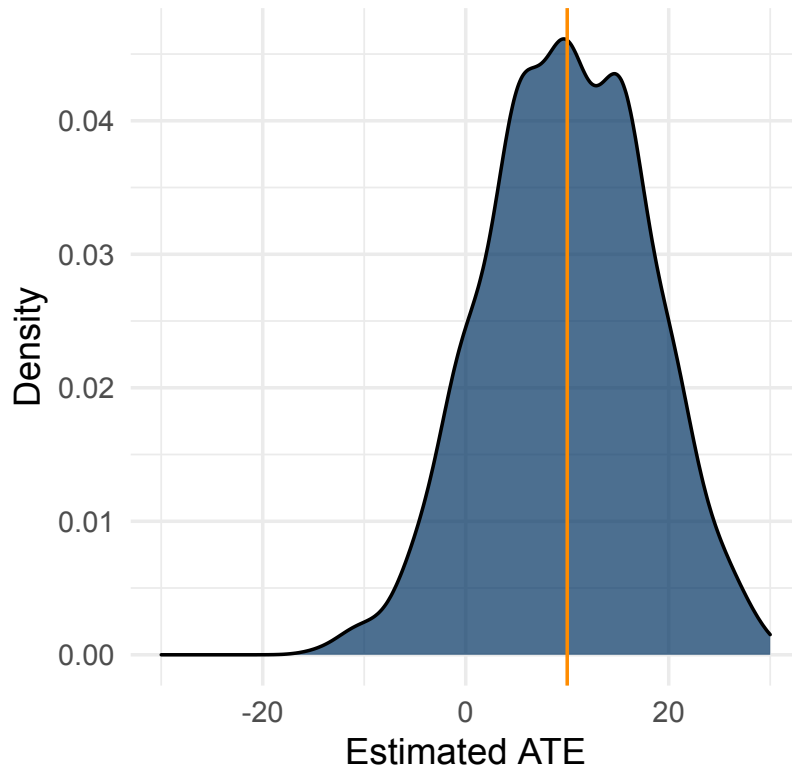
```
##      ate  women_in_treatment  
## [1,] 15.2 11  
## [2,]  4.8  9  
## [3,]  3.8  9  
## [4,] 15.5 11  
## [5,] 10.6 10  
## [6,]  7    9
```

Plot Normal ATE

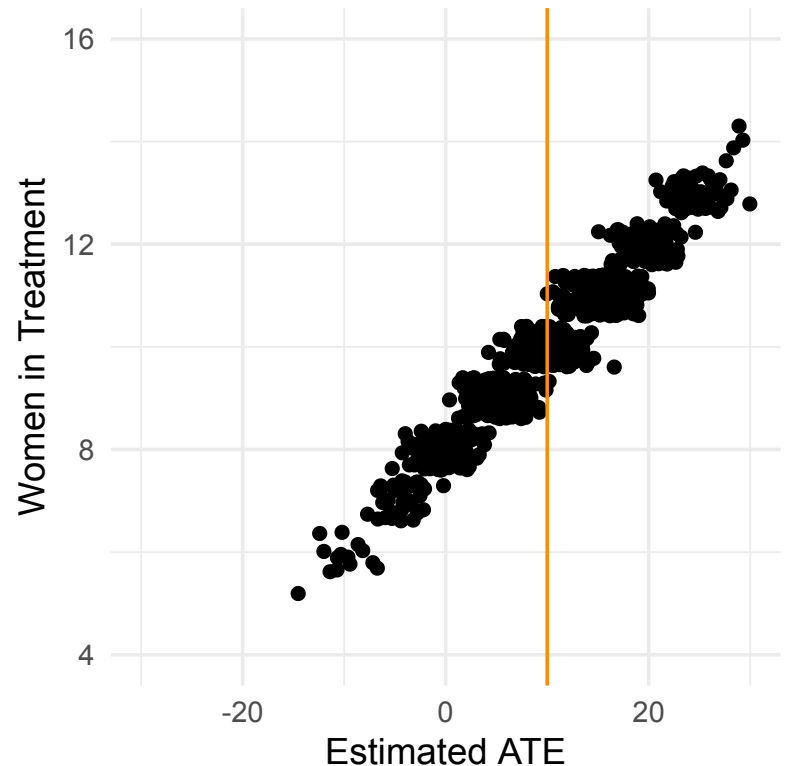
```
## Warning: Removed 8 rows containing non-finite values (stat_density).
```

```
## Warning: Removed 8 rows containing missing values (geom_point).
```

Zero Effect, Simple Randomization
Some extreme effects



Treatment Effect vs. Women in Treat
Linear, increasing relationship



Simulate a Block Randomized Study

```
simulate_blocked_study ← function(effect_size) {  
  ## create world  
  d ← make_data(effect_size=effect_size)  
  
  ## randomly assign and count the number of women in treatment  
  d[ , assignment := block_randomize(20), by = group]  
  
  women_in_treatment ← d[group = 'Woman' & assignment = 'Treatment', .N]  
  
  ## measure outcomes  
  d[ , Y := conduct_experiment(Control, Treatment, assignment)]  
  
  ## estimate ate  
  ate ← d[ , estimate_ate(y_values = Y, treatment = assignment)]  
  
  ## return objects  
  ## - `ate` from the `estimate_ate` function.  
  ## - `women_in_treatment` as a count  
  return(list('ate' = ate, 'women_in_treatment' = women_in_treatment))  
}
```

Simulate a Block Randomized Study

```
blocked_study ← simulate_blocked_study(effect_size = 10)
blocked_study

## $ate
## [1] 8.6
##
## $women_in_treatment
## [1] 10
```

Simulate Many Block Randomized

```
many_blocked_studies <- replicate(  
  n = 1000,  
  expr = simulate_blocked_study(effect_size = 10)  
)
```

```
many_blocked_studies <- t(many_blocked_studies)  
head(many_blocked_studies)
```

```
##      ate  women_in_treatment  
## [1,] 9.6   10  
## [2,] 12.8  10  
## [3,] 8.5   10  
## [4,] 9.5   10  
## [5,] 11.1  10  
## [6,] 9.1   10
```

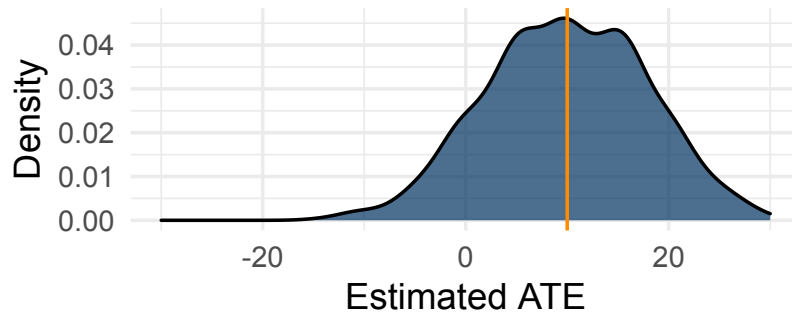

Plot Blocked and Unblocked ATE

```
## Warning: Removed 8 rows containing non-finite values (stat_density).
```

```
## Warning: Removed 8 rows containing missing values (geom_point).
```

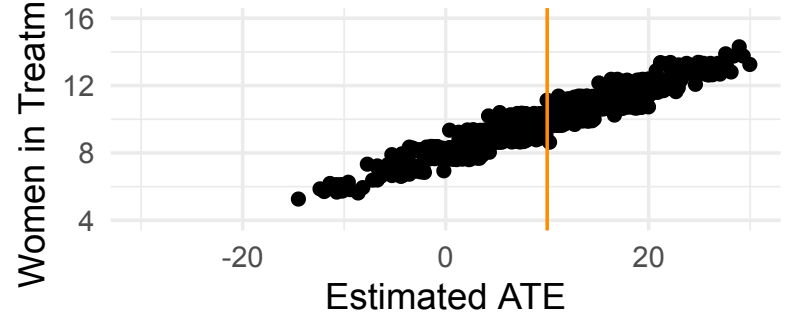
Zero Effect, Simple Randomization

Some extreme effects



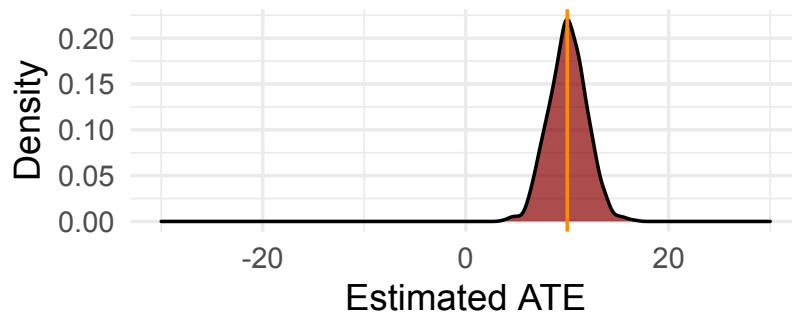
Treatment Effect vs. Women in Treatment

Linear, increasing relationship

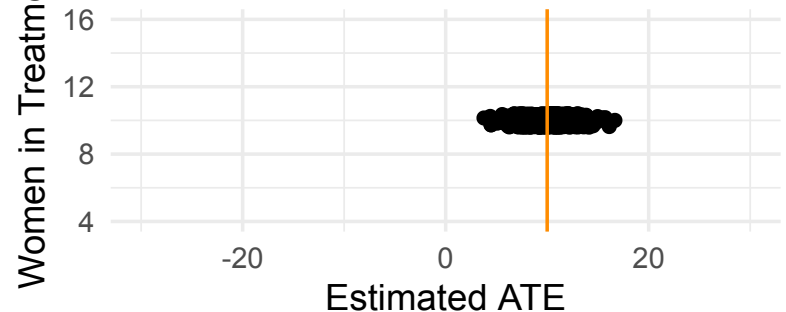


Zero Effect, Blocked Randomization

Results tightly centered at true treatment effect



Treatment Effect vs. Women in Treatment



P-values of Blocked vs. Normal Study

- Blocking allows for more precision (efficiency) by not conducting randomization where covaraites (e.g. sex) are very imbalanced
- This means that your estimator will produce an estimate that is closer to the *true* causal effect
- This is an estimate that is closer to the orange lines on the last slide

Preview: Regression

```
d ← make_data(effect_size = 10)
d[, assignment := randomize()]
d[, Y := conduct_experiment(Control, Treatment, assignment)]

model_simple ← d[, lm(Y ~ assignment)]
model_mf     ← d[, lm(Y ~ assignment + group)]
```

Preview: Regression, Cont'd

```
##
## =====
##                               Dependent variable:
##                               -----
##                               Y
##                               Unblocked          Blocked
##                               (1)              (2)
## -----
## Assigned Treatment          16.500           11.515
##                               (0.317, 32.683) (7.813, 15.218)
##                               p = 0.053*      p = 0.00000***
##
## Group: Woman
##                               49.848
##                               (46.146, 53.551)
##                               p = 0.000***
##
## Intercept                   32.250           9.818
##                               (20.807, 43.693) (6.726, 12.911)
##                               p = 0.00001***  p = 0.00000***
## -----
## Observations                40              40
## R2                          0.095           0.954
## Adjusted R2                 0.071           0.952
```

Summary of Blocking

Blocking

- Reduces the probability that a large Treatment vs. Control difference can occur by change by balancing the presence of similar units across Treatment and Control
- Can dramatically reduce the **standard error** of the estimator (i.e. the standard deviation of the sampling distribution)
- Is successful if the blocks predict the outcome
- Is unsuccessful if the blocks do **not** predict the outcome

Power

- Is affected by (a) sample size; and, (b) ratio of treatment effect to uncertainty about the estimate of treatment effect
- The standard deviation of the outcome is often much smaller within groups that are measurable before conducting the experiment

Clustering

Overview of Clustering

Clustering

- Often, units can be observed individually, but must be assigned to the same condition
- If there is covariance between group membership and outcomes (often there is) then this experiment does not do *as good a job* at breaking the relationship between treatment assignment and potential outcomes
- Alternatively, you might think of this experiment as producing less *information* about the treatment effect relative to the background noise of the world

Examples of Clustering

School Length

- Cannot randomly assign *at the student level* the length of the school day
- Instead, every student at the same school (or district, or state) has to receive the same length

Broadcast TV Advertisements

- In broadcast TV, it is not possible to assign advertisements (e.g. for **Babbitt's Sports** to specific individuals
- Instead, whole markets receive the same ads

Retail Stores and Prices

- **Retail stores** cannot individually assign price discounts to shoppers
- However, prices can be manipulated at the store-level and purchases observed at the individual-level

Reading Assignment

- Please read *Field Experiments* pages 80-85.

Clustering Example

Setting up Data

Cluster: The level where treatment is assigned

- Notice that outcomes can be observed at more fine-grained levels
- There may be difference in *cluster-average* outcomes:
 - A teacher stubs their toe on the way to school
 - It is hot or cold in one classroom

```
classrooms_number ← 8
students_number   ← 16 # this is MIDS, not kindergarden

classroom_ids_vector ← rep(x = 1:8, each = 16)
classroom_ids_vector
```

```
##    [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 3 3 3 3 3
##   [38] 3 3 3 3 3 3 3 3 3 3 3 3 4 4 4 4 4 4 4 4 4 4 4 4 5 5 5 5 5 5 5 5 5
##   [75] 5 5 5 5 5 5 6 6 6 6 6 6 6 6 6 6 6 6 6 6 7 7 7 7 7 7 7 7 7 7 7 7 7
##  [112] 7 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8
```

Classroom Example

```
classroom_noise_vector ← 1:8 + rnorm(n=8, mean=0, sd=.5)
```

```
potential_outcomes_data ← data.table(  
  student_id      = 1:(8*16),  
  classroom_id    = classroom_ids_vector,  
  classroom_noise = rep(classroom_noise_vector, each = students_number)  
)
```

```
setkey(potential_outcomes_data, 'classroom_id')
```

```
potential_outcomes_data[,  
  student_outcomes_control := rnorm(n=.N, mean=10) + classroom_noise]  
potential_outcomes_data[,  
  student_outcomes_treat := student_outcomes_control + rnorm(n=.N, mean=1.5)]
```

Classroom Example, cont'd

```
urn ← rep(c('Control', 'Treatment'), each = classrooms_number/2)

cluster_assignment_table ← data.table(
  classroom_id      = potential_outcomes_data[, unique(classroom_id)],
  classroom_assignment = sample(urn),
  key = 'classroom_id'
)

experiment_data ← merge(
  x = potential_outcomes_data,
  y = cluster_assignment_table,
  on = 'classroom_id')
```

Classroom Example, cont'd

```
head(experiment_data)
```

```
##      classroom_id student_id classroom_noise student_outcomes_control
## 1:                1         1         0.9414733                12.304659
## 2:                1         2         0.9414733                11.671225
## 3:                1         3         0.9414733                10.932684
## 4:                1         4         0.9414733                10.888481
## 5:                1         5         0.9414733                10.146979
## 6:                1         6         0.9414733                 9.566612
##      student_outcomes_treat classroom_assignment
## 1:                13.62628                Treatment
## 2:                13.26672                Treatment
## 3:                12.26637                Treatment
## 4:                11.94485                Treatment
## 5:                11.49461                Treatment
## 6:                10.81955                Treatment
```

Classroom Example, cont'd

```
experiment_data[ , classroom_assignment = 'Treatment']
```

```
## [1] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [13] TRUE TRUE TRUE TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [25] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE TRUE TRUE TRUE TRUE
## [37] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [49] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [61] FALSE FALSE FALSE FALSE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [73] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE FALSE FALSE FALSE FALSE
## [85] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [97] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [109] FALSE FALSE FALSE FALSE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [121] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
```

Classroom Example, cont'd

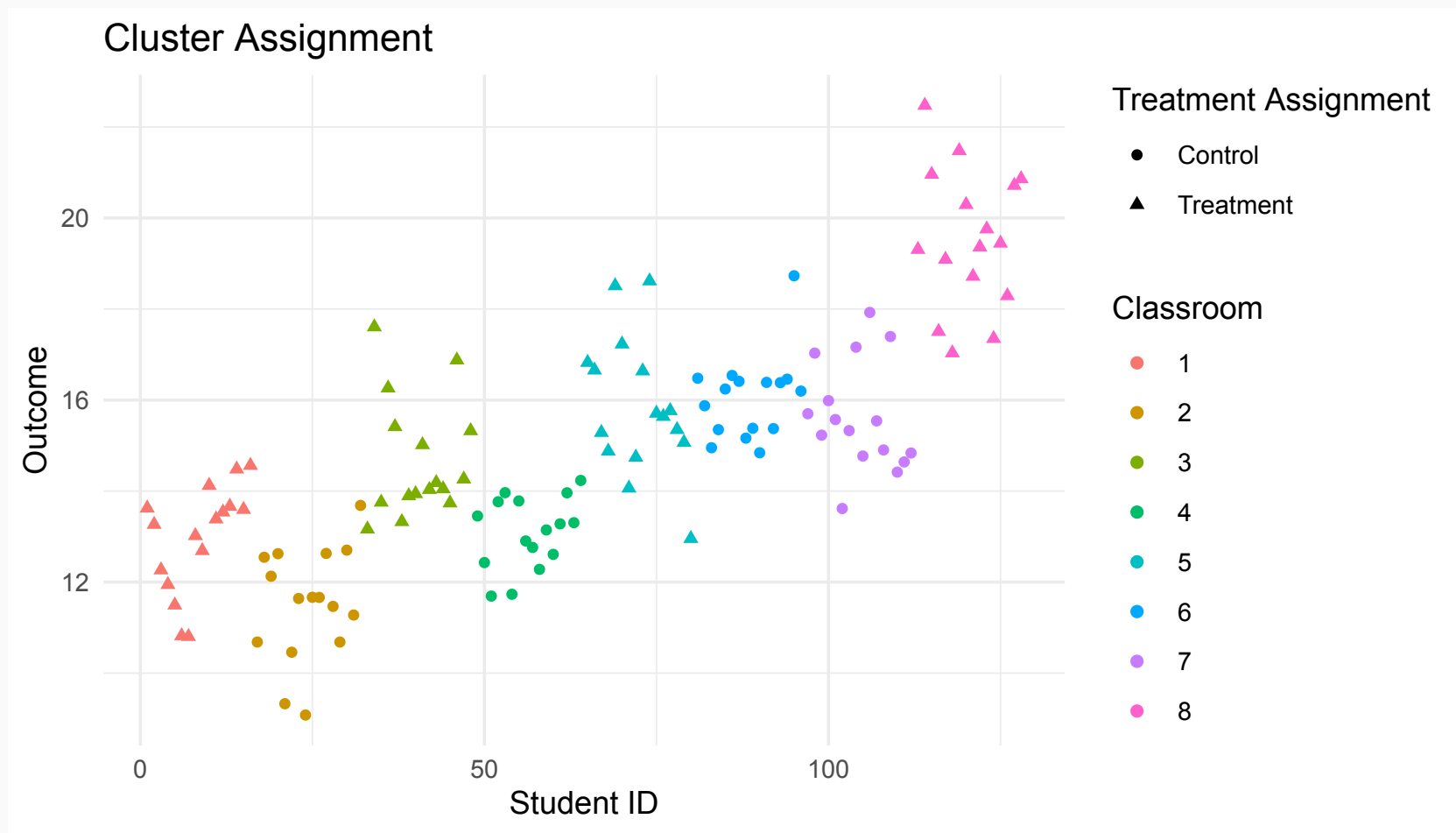
```
experiment_data[ , Y := conduct_experiment(  
  potential_control = student_outcomes_control,  
  potential_treatment = student_outcomes_treat,  
  assignment        = classroom_assignment )  
]
```

```
cluster_ate ← experiment_data[ , estimate_ate(  
  y_values = Y,  
  treatment = classroom_assignment,  
  verbose   = TRUE)  
]
```

```
cluster_ate
```

```
##      tg_mean  cg_mean      ate  
## 1: 15.76146 14.06886 1.692603
```


Notice ρ between classroom and Y



Classroom Example, Sharp Null

Repeat the reassignment process

- **Under the sharp null hypothesis** we assume that the potential outcomes to treatment that we *observe* for the units in the treatment group, $Y_i(1)|d_i = 1$, are equal to the potential outcomes to control that we *do not observe* for the units in the treatment group, $Y_i(0)|d_i = 1$.
- Similarly, **under the sharp null hypothesis** we assume that the potential outcomes to control that we *observe* for the people who are in the control group, $Y_i(0)|d_i = 0$ are equal to the potential outcomes to treatment that we *do not observe* for the units in the control group, $Y_i(1)|d_i = 0$

Take care

- Take care to notice how this statement of the sharp null hypothesis is different from the guarantees of apples-to-apples comparisons that are achieved through randomization

Classroom Example, Cluster RI Function

```
cluster_ri <- function(x_data, clustered) {  
  urn <- rep(c('Control', 'Treatment'), each = x_data[, length(unique(classroom_id))],  
  
  if(clustered == TRUE) {  
    cluster_assignment_table <- data.table(  
      classroom_id = x_data[, unique(classroom_id)],  
      ri_classroom_assignment = sample(urn),  
      key = 'classroom_id'  
    )  
  
    d <- merge(x_data, cluster_assignment_table, on = 'classroom_id')  
    d[, .(group_mean = mean(Y)),  
      keyby = .(ri_classroom_assignment)][, diff(group_mean)]  
  } else if(clustered == FALSE) {  
    d <- x_data  
  
    d[, .(group_mean = mean(Y)),  
      keyby = .(sample(classroom_assignment))[, diff(group_mean)]  
  }  
}
```

```
cluster_ri(x_data = experiment_data, clustered = TRUE)
```

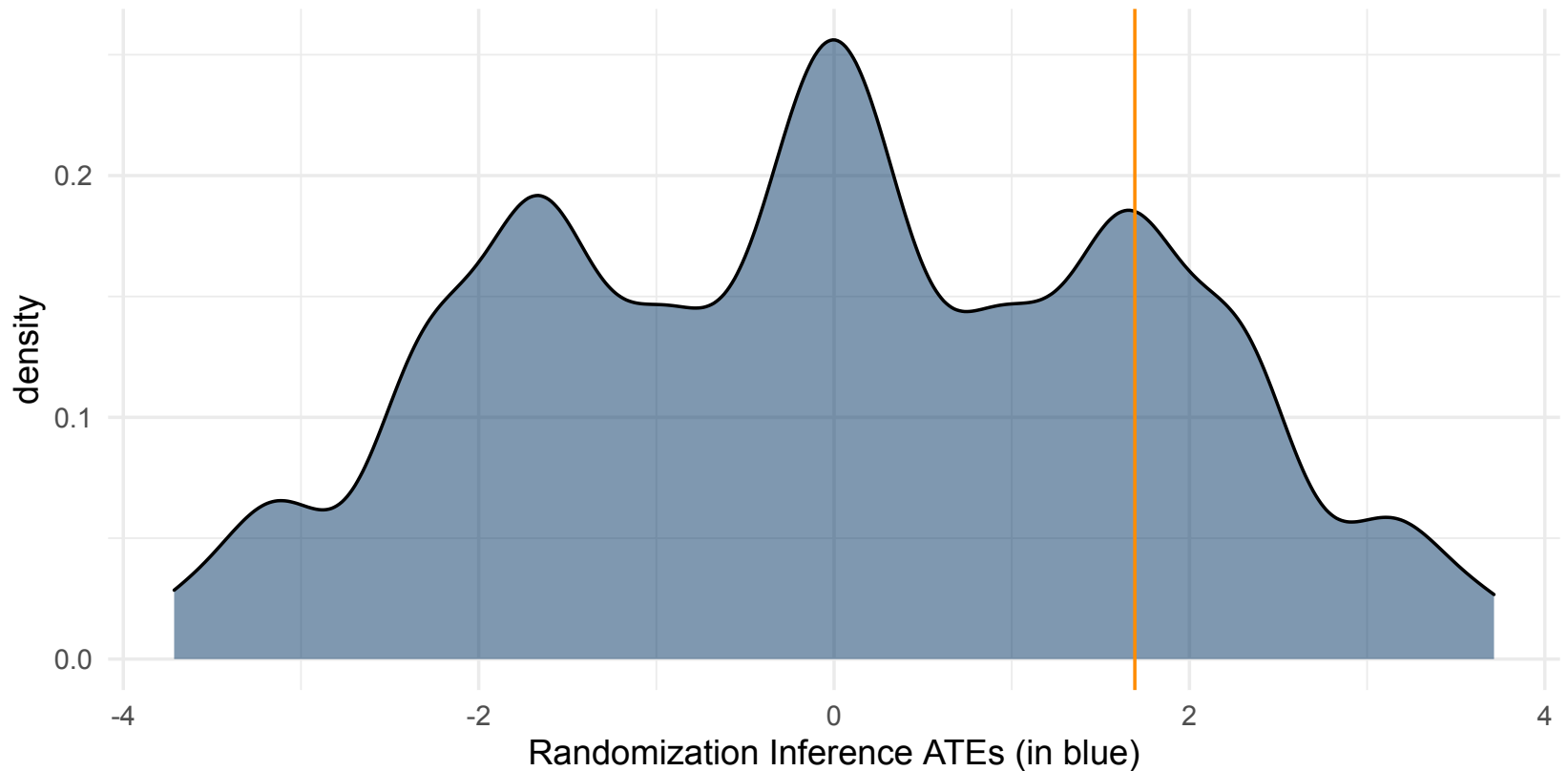
Classroom Example, Conduct Cluster RI

```
cluster_ri_ates <- replicate(  
  n = 5000,  
  cluster_ri(x_data = experiment_data, clustered = TRUE)  
)
```

Classroom Example, Plot Cluster RI

Cluster RI and estimated Treatment Effect

Two-sided p-value: 0.38



Perils of Ignoring Clustering

What if we ignore clustering?

What if we analyze results as though there wasn't clustering?

- Randomization happened at the classroom level, but our analysis ignores it?
- **Result:** p-values will be *too small* and false rejection rate will be larger than α that you're controlling for
- Potentially *much* larger depending on the correlation between classrooms and outcomes.

What if we ignore clustering?

```
non_clustered_ates ← replicate(  
  n = 5000,  
  cluster_ri(x_data = experiment_data, clustered = FALSE)  
)
```


Recall: Two-sample t-test

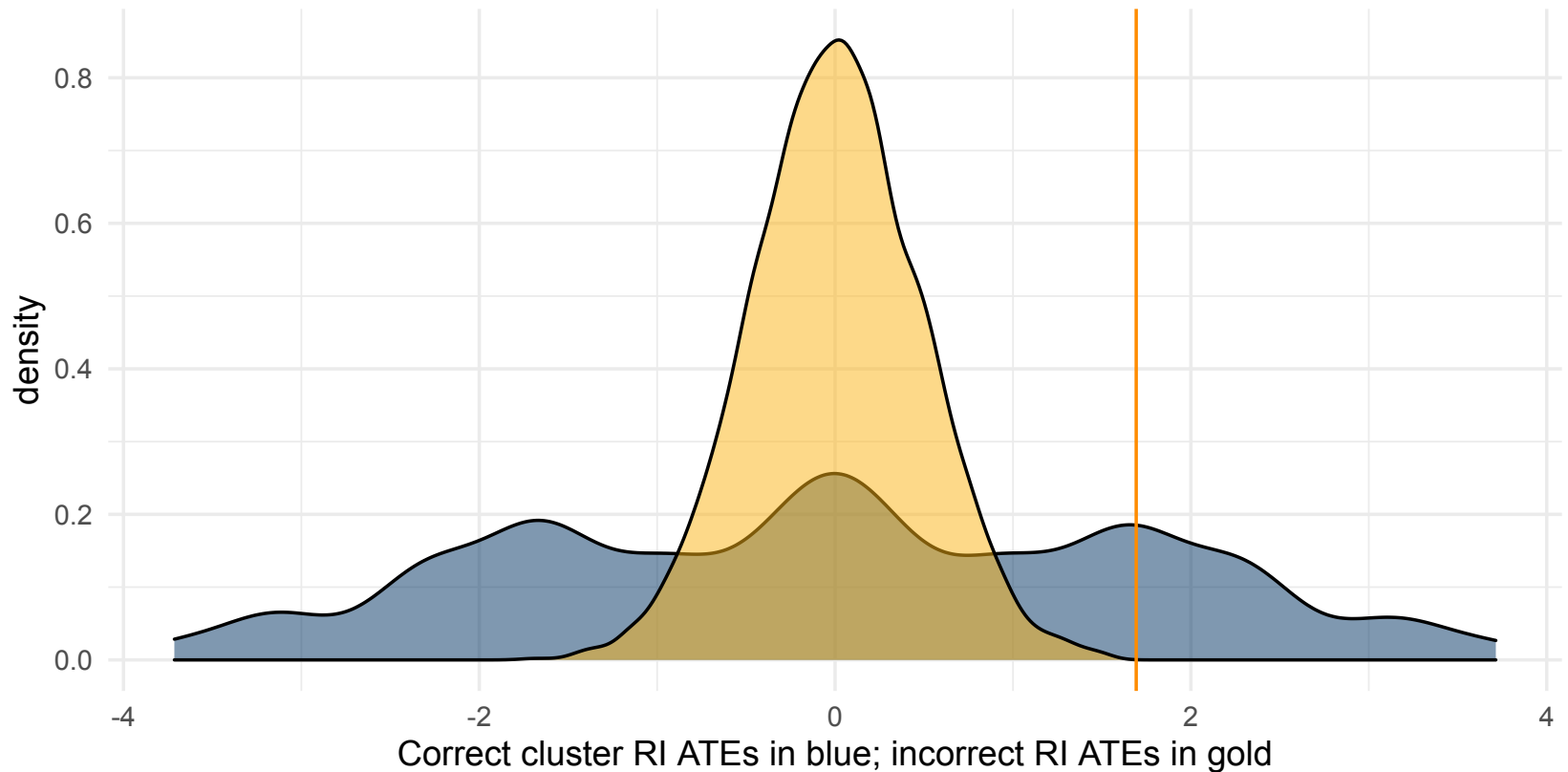
(Note: It isn't necessary to remember this formula from previous classes.)

$$t = \frac{\overline{X}_1 - \overline{X}_2}{\sqrt{\frac{s_1^2}{N_1} + \frac{s_2^2}{N_2}}}$$

What if we ignore clustering? (cont'd)

Cluster RI and estimated Treatment Effect

Correct Two-sided p-value: 0.38; incorrect two-sided p-value: 0.00



Summary

- When units are assigned to treatment or control in clusters, larger differences between `Treatment` and `Control` outcomes will happen by chance
- To account for this uncertainty, when conducting randomization inference, *match the experiment's assignment process*
 - If experimental assignment happened at random, randomization inference needs to as well
 - If experimental assignment happened with clusters, randomization inference needs to as well;
 - If experimental assignment happened with blocks, randomization inference needs to as well.
- Stats software (e.g. `vcovCL` in R) can estimate correct clustered *SEs* in a regression framework

Summary

Power is relatively worse when

- Average between cluster differences are larger
- Within cluster differences are smaller
- The number of clusters is small

Power is relatively better when

- Average between cluster differences are smaller
- Within cluster differences are larger
- The number of clusters is large