w241: Experiments and Causality

Unit 4

David Reiley, David Broockman, D. Alex Hughes
UC Berkeley, School of Information
Updated: 2021-05-25

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        12         8
##   Woman       8        12
```

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

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

Block Randomization

```r
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  Treatment          Control 2
## 3:       3         3   Man    Control        Treatment 3
## 4:       4         4   Man  Treatment          Control 4
## 5:       5         5   Man    Control        Treatment 5
## 6:       6         6   Man    Control          Control 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:   35.95   35.05 0.9
```

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] -5.3
##
## $women_in_treatment
## [1] 9
```

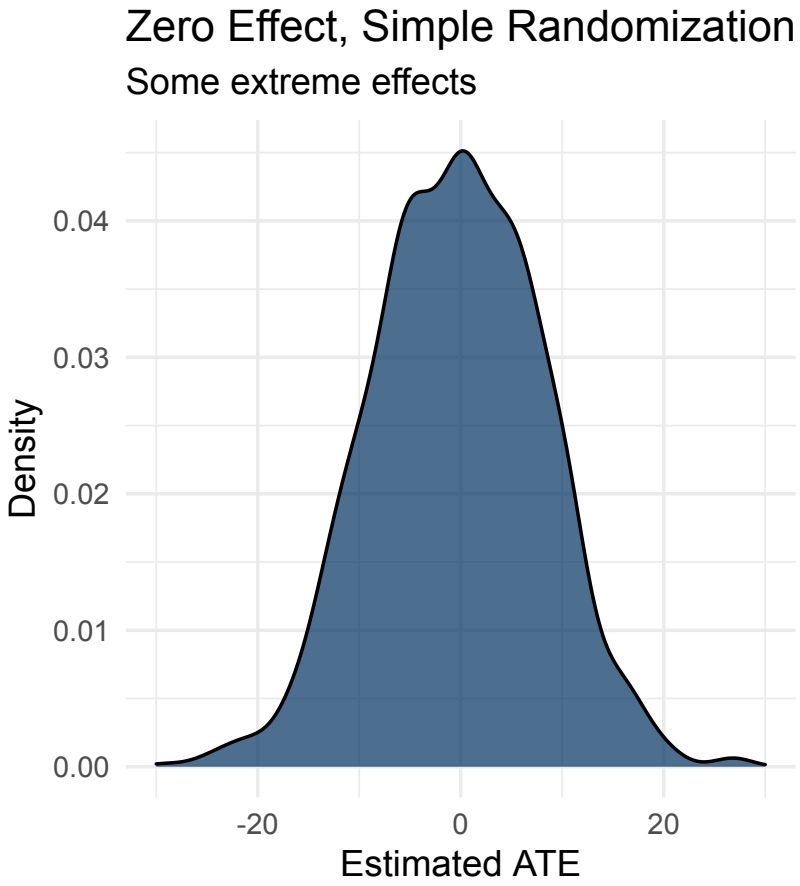
Simulate Many Normal Studies

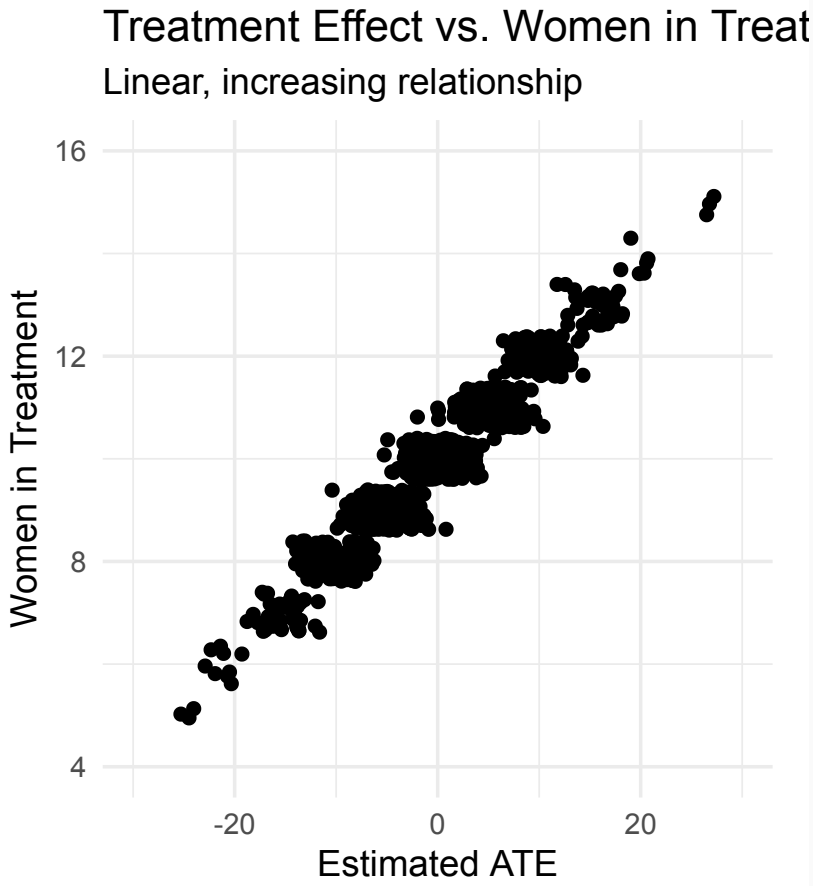
```
many_normal_studies ← replicate(
  n = 1000,
  expr = simulate_normal_study(effect_size = 0))

many_normal_studies ← t(many_normal_studies)
head(many_normal_studies)
```

```
##      ate  women_in_treatment
## [1,] 1.8  10
## [2,] 0.6  10
## [3,] 4.3  11
## [4,] -6   9
## [5,] -9.8 8
## [6,] 2.7  10
```

Plot Normal ATE

```
## Warning: Removed 1 rows containing missing values (geom_point).
```





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 = 0)
blocked_study
```

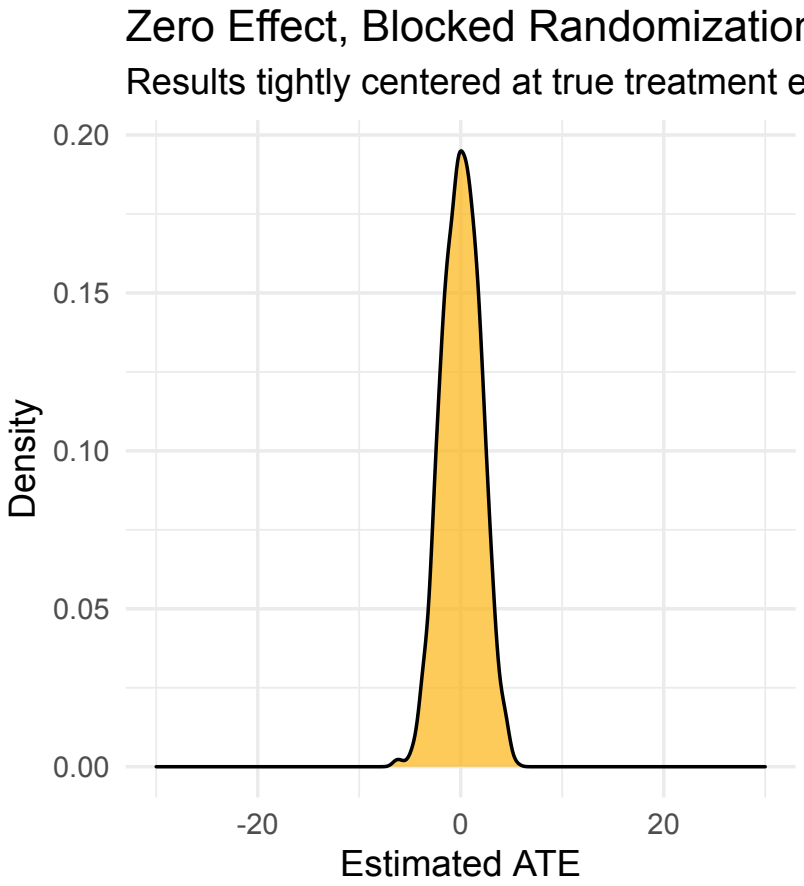
```
## $ate
## [1] -1.4
##
## $women_in_treatment
## [1] 10
```

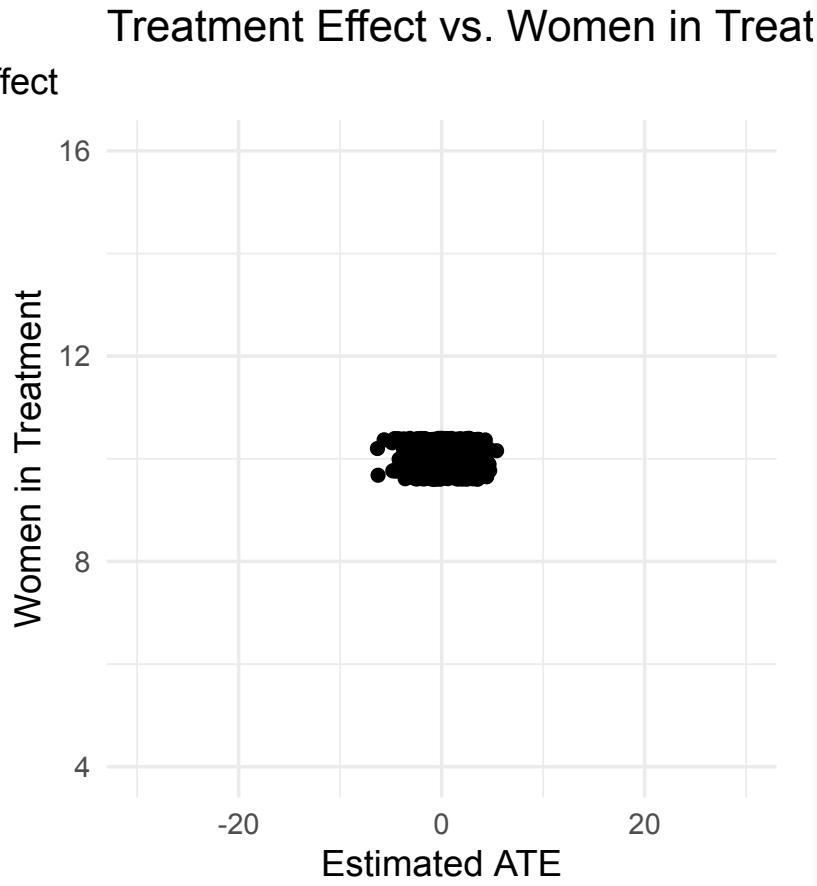
Simulate Many Block Randomized

```
many_blocked_studies ← replicate(
  n = 1000,
  expr = simulate_blocked_study(effect_size = 0)
)

many_blocked_studies ← t(many_blocked_studies)
head(many_blocked_studies)
```

```
##      ate  women_in_treatment
## [1,] 0.6  10
## [2,] -1.2 10
## [3,] -2.3 10
## [4,] 1.4  10
## [5,] 1.1  10
## [6,] 1.7  10
```

Plot Blocked ATE





Plot Unblocked and Blocked