Complier Average Causal Effect (CACE), Placebos (Ps), and Bad Controls (BCs)

Ethan Moody

November 2023

Contents

1 Consider Designs

1.1 Game night!

Suppose that you're advertising a board-game or online game to try and increase sales. You decide to individually randomly-assign into treatment and control. After you randomize, you learn that some treatment-group members are friends with control-group members IRL.

- What is the causal quantity that you would have **liked** to estimate?
- What is the causal quantity that you have **in fact** estimated?
- Is there any relationship between the two? Do you think that what you have estimated will be higher, lower, or about the same effect as the causal quantity that you would have liked to estimate?

Answer: The causal quantity that I would have **liked** to estimate is the *effect of the advertising treatment — delivered to the treatment group — on game sales* (in other words, I would have liked to determine whether or not the advertising alone caused additional game sales). Instead, the causal quantity that I have **in fact** estimated is *partly the effect of advertising and partly the effect of word-of-mouth promotion on game sales* since some participants in the treatment group have friends in the control group. Given these friendships, it's plausible to suspect that members in the treatment group would have told their friends in the control group about games they bought and enjoyed over the course of this experiment. It's even plausible to suspect that they shared the advertising/advertisement with them! This would cause us to potentially mis-measure (in this case underestimate) the effect of the advertising *alone* on game sales. I'd imagine that our estimate for the treatment effect of the advertising would be *lower* than the causal quantity that I would have liked to estimate. This is because we might expect average game sales in the control group to trend higher as a result of the social/friendship factor (or word-of-mouth promotion), which would reduce the difference in average sales between the treatment and control groups and lower the measured treatment effect.

1.2 Bonus time!

As we're writing this question, end-of-year bonuses are being given out in people's companies. (This is not a concept we your instructors have in the program – each day with your smiling faces is reward enough – and who needs money anyways?)

Suppose that you're interested in knowing whether this is a good idea from the point of view of worker productivity and so you agree to randomly assign bonuses to some people.

- What is the causal quantity that you would have **liked** to estimate?
- What is the causal quantity that you have **in fact** estimated?
- Is there any relationship between the two? Do you think that what you have estimated will be higher, lower, or about the same effect as the causal quantity that you would have liked to estimate?

Answer: The causal quantity that I would have **liked** to estimate is the *effect of bonuses — delivered/assigned to the treatment group — on employee productivity* (in other words, I would have liked to determine whether or not the bonuses alone caused a change in worker productivity). Instead, the causal quantity that I have **in face** estimated may be *partly the effect of the bonuses and partly (perhaps even more strongly!) the effect of news about the bonuses and an associated perception of unfairness in how they were doled out on employee productivity.* In this scenario — like in many workplaces — I think it's reasonable to assume that employees talk to one another about their bonuses. As a result, it's likely that those randomly assigned to receive a bonus in the treatment group would have friends/close coworkers that were not assigned to receive a bonus in the control group, and they'd share with these friends/coworkers the news that they received a bonus. I'd imagine the reaction of those in the control group to this news would be negative — for instance, they might express confusion, aggravation, or discouragement/devastation — and their productivity would change as a result. Based on the assumption that money/pay generally motivates employees to work harder (be more productive), those who received a bonus would likely have higher productivity (on average) and those who didn't receive a bonus would likely have lower productivity (on average). This would cause us to potentially mis-measure (in this case overestimate) the effect of bonuses *alone* on employee productivity. In fact, I'd imagine that our estimate for the treatment effect of the bonuses would be *higher* than the causal quantity

that I would have liked to estimate, because the news of no bonus for employees in the control group would reduce their productivity (rather than keep it flat/stable) and cause the difference in average productivity between treatment and control groups to look much more pronounced as a result.

2 Noncompliance in Recycling Experiment

2.1 Intent to treat effect

What is the ITT? Do the work to compute it, and store it into the object `recycling_itt`. Provide a short narrative using inline R code, such as r inline_reference.

```
# Generate data to simulate study - part 1: define key values from problem setup
treatment_n     <- 1500
treatment_dosed <- 700
treatment_rec   <- 500
control_n       <- 3000
control_rec     <- 600

# Generate data to simulate study - part 2: create table and treatment groups
recycling_d   <- data.table(
  treat = rep(c(1, 0), times = c(treatment_n, control_n)),
  rec   = 0
)

# Generate data to simulate study - part 3: add recycling outcomes
recycling_d[treat == 1, rec := rep(c(1, 0), times = c(
  treatment_rec, treatment_n - treatment_rec))]
recycling_d[treat == 0, rec := rep(c(1, 0), times = c(
  control_rec, control_n - control_rec))]

# Generate data to simulate study - part 4: shuffle rows and add ID's
shuffled_rows_recycling <- sample(nrow(recycling_d))
recycling_d             <- recycling_d[shuffled_rows_recycling,
                                       id := seq_len(treatment_n + control_n)]
recycling_d             <- recycling_d[order(id)]

# Generate data to simulate study - part 5: reorder columns
setcolorder(recycling_d, c("id", "treat", "rec"))

# Estimate ITT
recycling_itt <- recycling_d[treat == 1, mean(rec)] -
                 recycling_d[treat == 0, mean(rec)]

# Display estimated ITT
print(recycling_itt)
```

```
## [1] 0.1333333
```

Answer: The *ITT* ("intent-to-treat" effect) is the arithmetic difference between (1) the average recycling rate of the group assigned to receive treatment and (2) the average recycling rate of the group assigned to receive control. Based on the information given, we can estimate the *ITT* to be approximately 0.3333 - 0.2 = 0.1333. This means that the average recycling rate for the group of households that were provided information about the benefits of recycling (the treatment group) was 0.1333 higher than the average recycling rate for the group of households that were not provided information about the benefits of recycling (the control group). However, we know that this *ITT* is "diluted" compared to the *actual* treatment effect for the households who received treatment, since only a portion of the households intended to receive treatment (i.e., undergraduate students contacting them to explain the benefits of recycling) were actually contacted during the experiment. In order to produce an unbiased estimate of the *actual* treatment effect, we would need to reweight the *ITT*.

2.2 Compliers average causal effect

What is the CACE? Do the work to compute it, and store it into the object `recycling_cace`. Provide a short narrative using inline R code.

```
# Estimate take-up rate (ITT_D or alpha)
recycling_ittd <- treatment_dosed / treatment_n

# Estimate CACE
recycling_cace <- recycling_itt / recycling_ittd

# Display estimated CACE
print(recycling_cace)
```

```
## [1] 0.2857143
```

Answer: The $CACE$ ("compliance average causal effect") is the average treatment effect for the households who actually comply with their assignment, known as the "compliers". In this case, we know that only 700 of 1500 households (or approximately 46.67%) complied with their assignment to treatment by being contacted by the undergraduate students. This factor — known as the ITT_D or take-up rate — is what we need to scale-up/reweight our estimate of the ITT in order to compute the $CACE$. The $CACE$ is equal to the ITT (approximately 0.1333) divided by the ITT_D (approximately 0.4667), which comes out to roughly 0.2857. Our $CACE$ value provides an estimate of the treatment effect under the assumption that all households in the treatment group had actually complied with their assignment.

2.3 Mike's CACE

What is the CACE if Mike is correct? Provide a short narrative using inline R code.

```
# Define key values from problem setup
treatment_dosed_mike <- 500

# Estimate take-up rate (ITT_D or alpha) based on Mike's info
recycling_ittd_mike  <- treatment_dosed_mike / treatment_n

# Estimate Mike's CACE
cace_mike            <- recycling_itt / recycling_ittd_mike

# Display estimated CACE
print(cace_mike)
```

```
## [1] 0.4
```

Answer: If Mike's information is correct, the new ITT_D or take-up rate becomes equal to 500 / 1500 = 33.33%. This causes Mike's estimated $CACE$ to become 0.1333 / 0.3333 = 0.4. This estimated $CACE$ is higher than before because Mike's info suggests that a lower proportion of households assigned to the treatment group actually complied with their assignment (which increases the extent to which the ITT is scaled-up/reweighted to arrive at the $CACE$).

2.4 Andy's CACE

What is the CACE if Andy is correct? Provide a short narrative using inline R code.

```
# Define key values from problem setup
treatment_dosed_andy <- 600

# Estimate take-up rate (ITT_D or alpha) based on Andy's info
recycling_ittd_andy  <- treatment_dosed_andy / treatment_n
```

```r
# Estimate Andy's CACE
cace_andy            <- recycling_itt / recycling_ittd_andy

# Display estimated CACE
print(cace_andy)
```

```
## [1] 0.3333333
```

Answer: If Andy's information is correct, the new ITT_D or take-up rate becomes equal to 600 / 1500 = 40%. This causes Andy's estimated $CACE$ to become 0.1333 / 0.4 = 0.3333. This estimated $CACE$ is higher than our original estimate — but lower than the estimate based on Mike's info — because Andy's claim suggests that a lower proportion of households assigned to the treatment group actually complied with their assignment, though not as low as Mike's claim. In both circumstances, we end up with the understanding that fewer households actually complied, leading to a smaller scaling-up/reweighting factor and a larger $CACE$.

2.5 Effect of false reporting

What was the impact of the undergraduates's false reporting on our estimates of the treatment's effectiveness?

Answer: In the calculations above, we saw that the undergraduates' false reporting directly impacted our estimates of the $CACE$, which represents the treatment effect for "compliers". However, it's also worth noting that in *Field Experiments*, Gerber & Green state that changing the ITT_D (or treatment take-up rate) doesn't always necessarily change the $CACE$; in fact, it could change both the numerator (the ITT) and the denominator (ITT_D) of the fraction to compute the $CACE$ — depending on how "compliers" respond to treatment — resulting in either a changed or unchanged $CACE$ (p. 147). While we didn't recompute the ITT above — since we could reasonably assume that any "complier" households responded to treatment under the scenario where Mike was correct or the scenario where Andy was correct, just as reasonably as we could have assumed that some of these "complier" households didn't respond to treatment in one or both of those scenarios — it's certainly possible that our earlier ITT estimate could have changed based on Mike or Andy's info, and that the $CACE$ (under Mike or Andy's assumptions) could have actually remain unchanged. If Mike's information was in fact accurate, and if all 500 of the households in the treatment group that were successfully contacted by undergraduates recycled (i.e., 100% of them), we might suspect that the effectiveness of the treatment is quite high. Meanwhile, if Mike's information was in fact accurate, yet none of the 500 households in the treatment group that were successfully contacted by undergraduates recycled (i.e., 0% of them) but only those in the remaining subset of the treatment group recycled, we might suspect that the effectiveness of the treatment is quite low. This shows that our understanding of the treatment's effectiveness would change depending on *which specific* recycling households from the treatment group complied with/received the treatment and which did not. Ultimately, either way, in experiments where non-compliance exists, estimates of the treatment's effectiveness are more challenging to nail down in a way that isn't biased, especially when non-compliance or uncertainty about the level of compliance (as in this particular recycling experiment) is high.

2.6 Effect of false reporting... on what quantity?

Does your answer change depending on whether you choose to focus on the ITT or the CACE?

Answer: Yes, as noted in the previous answer, the $CACE$ changes in response to the undergraduates' false reporting, while the ITT does not — though, to be clear, that's *not* because the ITT is by necessity always unimpacted by considerations around ITT_D or take-up rate (in the general sense). In this particular experiment, I'm assuming that the proportion of households assigned to the treatment group that recycled remains unchanged, along with the proportion of households assigned to the control group that recycled, *regardless of assignment compliance*. This means that our ITT doesn't change even if our understanding of the level of compliance across the treatment group changes (e.g., based on Mike's info or Andy's info), because the ITT is based on the arithmetic difference between the average recycling rate for households in the treatment group and the average recycling rate for households in the control group (with no compliance considerations

impacting either one). However, one could imagine a scenario where false reporting on complying participants also calls into question the accuracy of reporting on outcome behaviors — leading to a change in ITT — or, at the very least, adds more uncertainty (or potentially bias) to an understanding of the treatment's effectiveness (as noted in the previous answer).

3 Fun with the placebo

3.1 Make data

Construct a data set that would reproduce the table. (Too frequently we receive data that has been summarized up to a level that is not useful for our analysis. Here, we're asking you to "un-summarize" the data to conduct the rest of the analysis for this question.)

```
# Generate unsummarized data - part 1: define key values from problem setup
base_n                <- 2463
base_torate           <- 0.3008
base_voted            <- round(base_n * base_torate, 0)
treat_dosed_n         <- 512
treat_dosed_torate    <- 0.3890
treat_dosed_voted     <- round(treat_dosed_n * treat_dosed_torate, 0)
treat_nondosed_n      <- 1898
treat_nondosed_torate <- 0.3160
treat_nondosed_voted  <- round(treat_nondosed_n * treat_nondosed_torate, 0)
plac_dosed_n          <- 476
plac_dosed_torate     <- 0.3002
plac_dosed_voted      <- round(plac_dosed_n * plac_dosed_torate, 0)
plac_nondosed_n       <- 2108
plac_nondosed_torate  <- 0.3145
plac_nondosed_voted   <- round(plac_nondosed_n * plac_nondosed_torate, 0)
total_n               <- 2463 + 512 + 1898 + 476 + 2108

# Generate unsummarized data - part 2: create table and experimental groups
d <- data.table(
  Assignment = c(
    rep("Baseline", base_n),
    rep("Treatment", treat_dosed_n + treat_nondosed_n),
    rep("Placebo", plac_dosed_n + plac_nondosed_n)),
  Treated = c(
    rep("No", base_n),
    rep("Yes", treat_dosed_n),
    rep("No", treat_nondosed_n),
    rep("Yes", plac_dosed_n),
    rep("No", plac_nondosed_n)),
  Voted = c(
    rep(c(1, 0), times = c(base_voted, base_n - base_voted)),
    rep(c(1, 0), times = c(treat_dosed_voted, treat_dosed_n - treat_dosed_voted)),
    rep(c(1, 0), times = c(treat_nondosed_voted, treat_nondosed_n - treat_nondosed_voted)),
    rep(c(1, 0), times = c(plac_dosed_voted, plac_dosed_n - plac_dosed_voted)),
    rep(c(1, 0), times = c(plac_nondosed_voted, plac_nondosed_n - plac_nondosed_voted)))
)

# Generate unsummarized data - part 3: shuffle rows and add ID's
shuffled_rows_d <- sample(nrow(d))
d               <- d[shuffled_rows_d, ID := seq_len(total_n)]
d               <- d[order(ID)]

# Generate unsummarized data - part 4: reorder columns
setcolorder(d, c("ID", "Assignment", "Treated", "Voted"))

# Reproduce summarized data
```

```r
d_summary <- d[, .(N = .N , Turnout = round(sum(Voted) / .N, 4)),
               keyby = .(Assignment, Treated)][order(match(Assignment, c(
                 "Baseline", "Treatment", "Placebo")), -Treated)]

# Display summarized data as a QC check
kable(d_summary)
```

Assignment	Treated	N	Turnout
Baseline	No	2463	0.3009
Treatment	Yes	512	0.3887
Treatment	No	1898	0.3161
Placebo	Yes	476	0.3004
Placebo	No	2108	0.3145

3.2 Estimate the compliance rate using the treatment group

Estimate the proportion of compliers by using the data on the treatment group. Provide a short narrative using inline R code, such as r inline_reference.

```r
# Calculate compliance rate for treatment group
compliance_rate_t <- d[Assignment == "Treatment" & Treated == "Yes", sum(.N)] /
                     d[Assignment == "Treatment", sum(.N)]

# Display compliance rate for treatment group
print(compliance_rate_t)
```

```
## [1] 0.2124481
```

Answer: The compliance rate for the treatment group represents the proportion of all subjects in the treatment group who were actually treated (i.e., canvassers delivered an encouragement to vote). To compute this rate, we take 512 (the number of subjects in the treatment group who received treatment) and divide it by 2410 (the total number of subjects in the treatment group), which is equal to approximately 0.2124 (or 21.24%). This value indicates a fairly low compliance rate and actually suggests that a larger percentage of subjects in the treatment group were *not* compliers than were compliers.

3.3 Estimate the compliance rate using the control group

C. Estimate the proportion of compliers by using the data on the placebo group. Provide a short narrative using inline R code.

```r
# Calculate compliance rate for placebo group
compliance_rate_p <- d[Assignment == "Placebo" & Treated == "Yes", sum(.N)] /
                     d[Assignment == "Placebo", sum(.N)]

# Display compliance rate for placebo group
print(compliance_rate_p)
```

```
## [1] 0.1842105
```

Answer: The compliance rate for the placebo group represents the proportion of all subjects in the placebo group who were actually treated (i.e., canvassers delivered a message unrelated to voting or politics). To compute this rate, we take 476 (the number of subjects in the placebo group who received treatment) and divide it by 2584 (the total number of subjects in the placebo group), which is equal to approximately 0.1842 (or 18.42%). Just like for the treatment group, this value indicates a fairly low compliance rate and actually suggests that a larger percentage of subjects in the placebo group were *not* compliers than were compliers.

3.4 Compare these compliance rates

Are the two compliance rates statistically significantly different from each other? Provide *a test* – this means that you cannot simply "look at" or "eyeball" the coefficients and infer some conclusion – and a description about why you chose that particular test, and why you chose that particular set of data.

```
# Run test to check if compliance differs significantly between treatment/placebo
proportions_difference_test <- prop.test(
  c(d[Assignment == "Treatment" & Treated == "Yes", sum(.N)],
    d[Assignment == "Placebo" & Treated == "Yes", sum(.N)]),
  c(d[Assignment == "Treatment", sum(.N)],
    d[Assignment == "Placebo", sum(.N)]))

# Display results from proportions difference test
proportions_difference_test
```

```
##
##  2-sample test for equality of proportions with continuity correction
##
## data:  c(d[Assignment == "Treatment" & Treated == "Yes", sum(.N)], d[Assignment == "Placebo" & Treat
## X-squared = 6.0887, df = 1, p-value = 0.0136
## alternative hypothesis: two.sided
## 95 percent confidence interval:
##  0.005698449 0.050776764
## sample estimates:
##    prop 1    prop 2
## 0.2124481 0.1842105
```

Answer: Based on a two-groups proportions test, the difference between the treatment group compliance rate of 0.2124 and the placebo group compliance rate of 0.1842 *is* statistically significant ($p = 0.013605$). The R documentation on prop.test states that this test "can be used for testing the null that the proportions (probabilities of success) in several groups are the same", which forms an appropriate hypothesis and basis for our comparison of the two compliance rates (as we want to check if these rates are, in fact, statistically equivalent or different). The test we're running above takes four inputs: (1) the number of subjects in the treatment group who complied with (actually received) the treatment, (2) the number of subjects in the placebo group who complied with (actually received) the placebo treatment, (3) the total number of subjects in the treatment group, and (4) the total number of subjects in the placebo group. Inputs (1) and (3) are used to compute the proportion of treatment subjects who complied, while inputs (2) and (4) are used to compute the proportion of placebo subjects who complied. With this data, the proportions test can determine if the two proportions are/are not sufficiently different from each other and, like other between-groups tests, provide an estimate of the probability (p-value) of obtaining a difference between these proportions that's at least as extreme as the difference we observe.

3.5 Evaluate assumptions

What critical assumption does this comparison of the two groups' compliance rates test? Given what you learn from the test, how do you suggest moving forward with the analysis for this problem?

Answer: This comparison tests for potential asymmetry in the approaches (e.g., message delivery styles, neighborhood/demographic bias, etc.) taken by the canvassers for households within the treatment and placebo groups, or other indicators of "impure" randomization. A critical assumption for unbiased estimation of the $CACE$ in a placebo study is that the treatment and placebo messages produce an equivalent set of compliers. We would hope to see the *same* rate of successful message delivery across both our treatment and placebo groups (or at least a non-statistically significant difference between the rate in each group) if households were, in fact, randomly assigned to these experimental groups. This two-groups proportions test provides evidence to suggest that we actually do *not* see the same compliance rate across the groups (i.e.,

they're significantly different), and that there could be some meaningful differences in how the canvassers approached the treatment households vs. placebo households and delivered the message, or in the types of neighborhoods involved in the treatment vs. placebo groups. Based on this finding, I would recommend proceeding with caution for the remainder of the analysis; it may not provide us with an unbiased estimate of the $CACE$, or with a precise/clear indication of the treatment effect from the encouragement to vote on its own.

3.6 Compliers average treatement effect... of the placebo?

Estimate the CACE of receiving the placebo. Is the estimate consistent with the assumption that the placebo has no effect on turnout?

```
# Calculate estimated ITT of placebo group
itt_estimate  <- d[Assignment == "Placebo", mean(Voted)] -
                 d[Assignment == "Baseline", mean(Voted)]

# Calculate estimated CACE of placebo group
cace_estimate <- itt_estimate / compliance_rate_p

# Display estimated CACE of placebo group
print(cace_estimate)
```

```
## [1] 0.06007738
```

Answer: The $CACE$ of receiving the placebo is approximately 0.0601. This value does not *appear* to be the same as zero, suggesting (at least on the surface) that it's *inconsistent* with the assumption that the placebo has no effect on turnout; however, in order to tell for certain, we'd first need to calculate the robust standard error of the $CACE$ estimate and then run a hypothesis test, like a t-test, to check if the probability of obtaining this estimate is sufficiently small (e.g., <5% or <0.05) under the assumption that the null hypothesis of "no difference from zero" is actually true. If the result we obtained above *did* register as significantly different from zero, we could conclude that the placebo messaging itself is actually affecting voter turnout. For example, perhaps the placebo messaging is related in some way to a political candidate's platform and ultimately inspires some households to vote for that candidate. Or, perhaps there are randomization/design issues with the study, like the canvassers taking different approaches for treatment households vs. placebo households. Either way, if our hypothesis test yielded significance, it would be reasonable to suspect that one or more of these factors might be causing the placebo to have *some* meaningful effect on turnout; otherwise, without significance, we could conclude that the placebo $CACE$ is consistent with the assumption that the messaging has no effect on turnout (as expected).

3.7 Diference in means estimator

Using a difference in means (i.e. not a linear model), compute the ITT using the appropriate groups' data. Then, divide this ITT by the appropriate compliance rate to produce an estiamte the CACE. Provide a short narrative using inline R code.

```
# Calculate estimated ITT
itt        <- d[Assignment == "Treatment", mean(Voted)] -
              d[Assignment == "Baseline", mean(Voted)]

# Calculate estimated CACE
cace_means <- itt / compliance_rate_t

# Display estimated CACE
print(cace_means)
```

```
## [1] 0.1444242
```

Answer: Using a difference in means approach, the $CACE$ of receiving the treatment is approximately 0.1444. This value suggests that the treatment *does* have a positive, non-zero effect on turnout.

3.8 Linear model estimator

Use two separate linear models to estimate the CACE of receiving the treatment by first estimating the ITT and then dividing by ITT_D. Use the `coef()` extractor and in line code evaluation to write a descriptive statement about what you learn after your code.

```
# Create model to generate ITT
itt_model   <- d[, lm(Voted ~ Assignment)]

# Create model to generate ITT_D
# NOTE: `Treated` variable first needs to be transformed to have integer/numeric data
itt_d_model <- d[, lm(ifelse(Treated == "Yes", 1, 0) ~ Assignment)]

# Display ITT and ITT_D
paste("ITT:", round(coef(itt_model)[3], 4))
```

```
## [1] "ITT: 0.0307"
```

```
paste("ITT_D:", round(coef(itt_d_model)[3], 4))
```

```
## [1] "ITT_D: 0.2124"
```

```
# Calculate estimated CACE using ITT and ITT_D
cace_models <- coef(itt_model)[3] / coef(itt_d_model)[3]

# Display estimated CACE
paste("CACE:", round(cace_models, 4))
```

```
## [1] "CACE: 0.1444"
```

Answer: Based on this linear model approach, the $CACE$ of receiving the treatment is approximately 0.1444, which is identical to what we observed when calculating the $CACE$ using a difference in means estimator. As stated previously, this value suggests that the canvassers' message (one that encouraged subjects to vote) *does* have a positive effect on voting behavior.

3.9 Data subset estimator

When a design uses a placebo group, one additional way to estiamte the CACE is possible – subset to include only compliers in the treatment and placebo groups, and then estimate a linear model. Produce that estimate here. Provide a short narrative using inline R code.

```
# Create model to generate CACE estimate through subsetting data to compliers
cace_subset_model <- d[(Assignment == "Treatment" & Treated == "Yes") |
                       (Assignment == "Placebo" & Treated == "Yes"),
                       lm(Voted ~ Assignment)]

# Display estimated CACE
print(round(coef(cace_subset_model)[2], 4))
```

```
## AssignmentTreatment
##              0.0883
```

Answer: Based on this data subset approach, the $CACE$ of receiving the treatment is approximately 0.0883. Just like for the previous approach, this result suggests that the canvassers' message (one that encouraged subjects to vote) *does* have a positive effect on voting behavior when this message is actually delivered to

those subjects intended to receive it. However, the $CACE$ estimate using this method differs by 0.0883 - 0.1444 = -0.0561 from the previous method. This finding is consistent with the fact that we observed a significant difference in compliance rates between the treatment and placebo groups — and given this significant difference, we'd expect to see some bias in an approach that looks at the effect of the messaging for both of these groups. In other words, this data subset estimator method may not always yield the same $CACE$ estimate as the difference in means estimator method or linear model estimator method, especially in situations (like this one) when there could be unintentional design/experimental execution issues at play (e.g., randomization issues, approach differences, etc.).

3.10 Evaluate estimators

In large samples (i.e. "in expectation") when the design is carried out correctly, we have the expectation that the results from 7, 8, and 9 should be the same. Are they? If so, does this give you confidence that these methods are working well. If not, what explains why these estimators are producing different estimates?

Answer: The results from questions 7 and 8 are identical, but are *not* the same for 9. The dissimilarity between the first/second estimate and the third estimate actually doesn't surprise me, since we observed a statistically significant difference between the treatment and placebo group compliance rates when we ran a two-groups proportions test earlier. Given this significant difference in compliance rates, we concluded that there *could* be some meaningful differences in how the canvassers approached the treatment households vs. placebo households and delivered their respective messages in this experiment, or in the types of neighborhoods canvassed for the treatment group and placebo group, or in how randomization of households to experimental groups was conducted in general, etc. Without an equivalent set of compliers across these groups, we actually don't meet a critical assumption for *unbiased estimation of the* $CACE$; instead, we're ultimately left with the possibility that the estimator for the $CACE$ could be biased due to potential design execution issues. This would explain why we see a different value for our third $CACE$ estimate (using a compliers subset approach that accounts for both the treatment and placebo groups), but equivalent values for our first two $CACE$ estimates (using a difference in means approach and a linear model approach that accounts for just the treatment group). If we had seen no significant difference between the treatment/placebo compliance rates when we compared them earlier, we would have expected to see three equivalent $CACE$ estimates for 7, 8, and 9.

4 Another Turnout Question

4.1 Simple treatment effect

Load the data and estimate a `lm` model that compares the rates of turnout in the control group to the rate of turnout among anybody who received *any* letter. This model combines all the letters into a single condition – "treatment" compared to a single condition "control". Report robust standard errors, and include a narrative sentence or two after your code using inline R code, such as r inline_reference.

```
# Create linear model to estimate treatment effect of any letter on turnout
mod_ste      <- d[, lm(vote ~ any_letter)]

# Calculate robust standard errors (RSEs)
mod_ste_rses <- sqrt(diag(vcovHC(mod_ste)))

# Display stargazer table with model results and RSEs
stargazer(
  mod_ste,
  se = list(mod_ste_rses),
  add.lines = list(
    c("Using Robust Standard Errors", "Yes"),
    c("Treatment = Any Letter", "Yes")),
  type = "latex",
  #type = "text")
  header = FALSE)
```

Table 2:

	Dependent variable:
	vote
any_letter	0.001
	(0.005)
Constant	0.085***
	(0.001)
Using Robust Standard Errors	Yes
Treatment = Any Letter	Yes
Observations	100,000
R^2	0.00000
Adjusted R^2	−0.00001
Residual Std. Error	0.279 (df = 99998)
F Statistic	0.077 (df = 1; 99998)
Note:	*p<0.1; **p<0.05; ***p<0.01

Answer: The estimated treatment effect of receiving *any* letter is both non-significant and negligible/near-null. Based on our linear model, this treatment effect is approximately 0.0013 — that is, approximately 0.13% points different from the control group's voter turnout. This value suggests that voter turnout is only marginally different for the voters who receive *any* letter in the mail than for the control group of voters.

4.2 Letter-specific treatment effects

Suppose that you want to know whether different letters have different effects. To begin, what are the effects of each of the letters, as compared to control? Estimate an appropriate linear model and use robust standard errors. Provide a short narrative using inline R code.

```
# Create linear model to estimate treatment effects of different letters on turnout
mod_lte      <- d[, lm(vote ~ treatment)]

# Calculate robust standard errors (RSEs)
mod_lte_rses <- sqrt(diag(vcovHC(mod_lte)))

# Display stargazer table with model results and RSEs
stargazer(
  mod_lte,
  se = list(mod_lte_rses),
  add.lines = list(
    c("Using Robust Standard Errors", "Yes"),
    c("Treatment = Letter-Specific", "Yes")),
  type = "latex",
  #type = "text")
  header = FALSE)
```

Table 3:

	Dependent variable:
	vote
treatmentElection info	−0.004
	(0.010)
treatmentPartisan	0.001
	(0.007)
treatmentTop-two info	0.004
	(0.007)
Constant	0.085***
	(0.001)
Using Robust Standard Errors	Yes
Treatment = Letter-Specific	Yes
Observations	100,000
R^2	0.00000
Adjusted R^2	−0.00003
Residual Std. Error	0.279 (df = 99996)
F Statistic	0.154 (df = 3; 99996)
Note:	*p<0.1; **p<0.05; ***p<0.01

Answer: There do *appear* to be different treatment effects associated with each type of letter based on the above linear model. However, like before, we don't see statistical significance for *any* of these effects. From the coefficients in our model, we can say the following about the treatment effect of each letter: (1) the effect of the "election info" letter is roughly -0.0037 — that is, approximately -0.37% points different from the control group's voter turnout; (2) the effect of the "partisan" letter is roughly 0.0014 — that is, approximately

0.14% points different from the control group's voter turnout; and (3) the effect of the "top-two info" letter is roughly 0.0038 — that is, approximately 0.38% points different from the control group's voter turnout. These findings suggest that while the "election info" letter *may* actually reduce voter turnout, the "partisan" letter and "top-two info" letter *may* increase voter turnout, with the "top-two info" letter having an even larger effect than the "partisan" letter. That being said, these conclusions should be tempered by the fact that the coefficients from our model are *not* statistically significant and the RSEs could truthfully swing them into either negative, positive, or null (zero) territory.

4.3 Test for letter-specific effects

Does the increased flexibilitiy of a different treatment effect for each of the letters improve the performance of the model? Test, using an F-test. What does the evidence suggest, and what does this mean about whether there **are** or **are not** different treatment effects for the different letters?

```
# Run F-test (ANOVA) to see if letter-specific model shows improved performance
ftest_ste_lte        <- anova(mod_ste, mod_lte, test = "F")

# Display result of F-test (ANOVA)
ftest_ste_lte
```

```
## Analysis of Variance Table
##
## Model 1: vote ~ any_letter
## Model 2: vote ~ treatment
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1  99998 7772.5
## 2  99996 7772.5  2  0.029975 0.1928 0.8246
```

```
# Extract and print estimated F-statistic from F-test
ftest_ste_lte_fstat  <- round(ftest_ste_lte$F[2], 4)

# Extract and print estimated p-value from F-test
ftest_ste_lte_pvalue <- round(ftest_ste_lte$`Pr(>F)`[2], 4)

# Print F-statistic and p-value from F-test
paste("Estimated F-statistic:", ftest_ste_lte_fstat)
```

```
## [1] "Estimated F-statistic: 0.1928"
```

```
paste("Estimated p-value:", ftest_ste_lte_pvalue)
```

```
## [1] "Estimated p-value: 0.8246"
```

Answer: Our F-test suggests that accounting for the specific letter types doesn't yield any meaningful improvement in model fit (amount of variability explained). The two main pieces of evidence for this conclusion are (1) the F-statistic, which is 0.1928 (very low), and (2) the p-value, which is 0.8246 (very high — ultimately greater than the typical significance threshold of 0.05). This means that even though the coefficients appear to differ for each letter type in our letter-specific model, we must ultimately conclude that there *are not* (statistically) different treatment effects for the different letters.

4.4 Compare letter-specific effects

Is one message more effective than the others? The authors have drawn up this design as a full-factorial design. Write a *specific* test for the difference between the *Partisan* message and the *Election Info* message. Write a *specific* test for the difference between *Top-Two Info* and the *Election Info* message. Report robust standard errors on both tests and include a short narrative statement after your estimates.

```r
# Create copy of original data.table for editing
d_cpy               <- copy(d)

# Subset copied data.table into individual groups for pairwise comparisons
d_cpy_pm_em         <- d_cpy[(treatment == "Partisan" | treatment == "Election info"),
                             c("treatment", "vote"), with = FALSE]
d_cpy_tm_em         <- d_cpy[(treatment == "Top-two info" | treatment == "Election info"),
                             c("treatment", "vote"), with = FALSE]

# Create linear model to check for difference between `Partisan` and `Election info`
mod_diff_pm_em      <- d_cpy_pm_em[, lm(vote ~ treatment)]

# Calculate robust standard errors (RSEs)
mod_diff_pm_em_rses <- sqrt(diag(vcovHC(mod_diff_pm_em)))

# Display stargazer table with model results and RSEs
stargazer(
  mod_diff_pm_em,
  se = list(mod_diff_pm_em_rses),
  add.lines = list(
    c("Using Robust Standard Errors", "Yes"),
    c("Treatments Compared", "Partisan, Election info")),
  type = "latex",
  #type = "text")
  header = FALSE)
```

Table 4:

	Dependent variable:
	vote
treatmentPartisan	0.005
	(0.012)
Constant	0.081***
	(0.010)
Using Robust Standard Errors	Yes
Treatments Compared	Partisan, Election info
Observations	2,355
R^2	0.0001
Adjusted R^2	−0.0003
Residual Std. Error	0.278 (df = 2353)
F Statistic	0.177 (df = 1; 2353)
Note:	*p<0.1; **p<0.05; ***p<0.01

```r
# Create linear model to check for difference between `Top-two info` and `Election info`
mod_diff_tm_em      <- d_cpy_tm_em[, lm(vote ~ treatment)]

# Calculate robust standard errors (RSEs)
mod_diff_tm_em_rses <- sqrt(diag(vcovHC(mod_diff_tm_em)))
```

```r
# Display stargazer table with model results and RSEs
stargazer(
  mod_diff_tm_em,
  se = list(mod_diff_tm_em_rses),
  add.lines = list(
    c("Using Robust Standard Errors", "Yes"),
    c("Treatments Compared", "Top-two info, Election info")),
  type = "latex",
  #type = "text")
  header = FALSE)
```

Table 5:

	Dependent variable:
	vote
treatmentTop-two info	0.007
	(0.012)
Constant	0.081***
	(0.010)
Using Robust Standard Errors	Yes
Treatments Compared	Top-two info, Election info
Observations	2,370
R^2	0.0002
Adjusted R^2	−0.0003
Residual Std. Error	0.281 (df = 2368)
F Statistic	0.380 (df = 1; 2368)
Note:	*p<0.1; **p<0.05; ***p<0.01

Answer: To test if one message is more effective than the others, we can subset the original data into paired groups of treatments (e.g., "partisan" and "election info", "top-two info" and "election info") and then run a linear model that regresses voter turnout on treatment message. Doing so (above) results in a series of tables showing the coefficients/effects of a particular treatment message and the constant term, representing the other treatment message included in the pairwise test. For example, the first table shows the outcome of the comparison between "partisan" and "election info", with "election info" representing the constant term; meanwhile, the second table shows the outcome of the comparison between "top-two info" and "election info", with "election info" again representing the constant term. *The conclusion I draw from both tests is that there is no statistically significant difference between the "partisan" and "election info" effects and between the "top-two info" and "election info" effects.* While it's tempting to say that "partisan" messaging is slightly more effective than "election info" messaging, and "top-two info" messaging also is slightly more effective than "election info" messaging (based on the positive coefficients for these treatments in the tables above), we cannot — in good faith — conclude that this is a scientifically compelling and statistically significant finding. Rather, given the lack of significance on the coefficients and the reported RSEs, we must say that the apparent differences in effectiveness could be due to chance alone.

4.5 Count the number of blocks

Blocks? We don't need no stinking blocks? The blocks in this data are defined in the `block.num` variable (which you may have renamed). There are a *many* of blocks in this data, none of them are numerical – they're all category indicators. How many blocks are there?

```r
# Count number of unique `block` values
d_block_count <- d[, .(unique_blocks = uniqueN(block))]

# Display unique `block` count
d_block_count
```

```
##    unique_blocks
##            <int>
## 1:           283
```

Answer: There are exactly 283 unique blocks in this 100,000-row subset of data.

4.6 Add block fixed effects

SAVE YOUR CODE FIRST but then try to estimate a `lm` that evaluates the effect of receiving *any letter*, and includes this block-level information. What happens? Why do you think this happens? If this estimate *would have worked* (that's a hint that we don't think it will), what would the block fixed effects have accomplished?

```r
##
## SAVE YOUR CODE: before you run the next lines, because it's going
##                 to crash you if you're on the ischool.datahub.
##                 ... but why does it crash you?
##
##  Notice that in the chunk declaration, we have set `eval = FALSE`; this is so the code doesn't run u
##
## We'll even write some code that will help.
##   - In the first chain of this data.table, we are scoping only to the columns that we will use in th
##   - In the second, we are estimating the model
model_block_fx      <- d[ , .(vote, any_letter, block)][ , lm(vote ~ any_letter + factor(block))]

################################################################################
# Calculate robust standard errors (RSEs)
# NOTE: RAN INTO ISSUES WITH THIS GIVEN NUMBER OF BLOCKS AND SIZE OF DATA
# model_block_fx_rses <- sqrt(diag(vcovHC(model_block_fx)))
################################################################################

# Display stargazer table with model results and RSEs
stargazer(
  model_block_fx,
  #se = list(model_block_fx_rses),
  omit = "block",
  add.lines = list(
    c("Using Robust Standard Errors", "No"),
    c("Treatment = Letter-Specific", "Yes"),
    c("Including Block Fixed Effects", "Yes")),
  type = "latex",
  #type = "text")
  header = FALSE)
```

Answer: The blocked fixed effects model should give us more insight into the degree to which differences in voting turnout might be related to whatever variables were "blocked" on. The documentation above gives us very limited detail on the exact blocking variables, but the published paper from Hill and Kousser (2015) provides more info; it indicates that the researchers "blocked on age, party, individual 2010 and 2008 general election vote history, district competitiveness, and whether or not a district had a plurality or voters from an

ethnic or racial minority" (p. 423). Because voting behavior/turnout likely co-varies with or is impacted by these factors (specifically in the sense that, as the paper states, "different parts of the population might be more hindered by different factors of non-participation"), incorporating blocked fixed effects seems like a good way to more precisely estimate the *effect of the treatment messaging* across a wide variety of prospective voters, all while controlling for variables other than that messaging. Generally speaking, including blocked fixed effects should also reduce the error in the model predictions (i.e., lower RSEs/standard error estimates) and improve model precision.

4.7 A clever work-around?

Even though we can't estimate this fixed effects model directly, we can get the same information and model improvement if we're *just a little bit clever*. Create a new variable that is the *average turnout within a block* and attach this back to the data.table. Use this new variable in a regression that regresses voting on `any_letter` and this new `block_average`. Then, using an F-test, does the increased information from all these blocks improve the performance of the *causal* model? Use an F-test to check.

```
# Calculate average voter turnout by `block` and assign to new column `block_average`
d                           <- d[, block_average := mean(vote), by = block]

# Create linear model to estimate effects of letters (treatment) and blocks (fixed) on turnout
mod_lte_bfe_wa              <- d[, lm(vote ~ any_letter + block_average)]

# Calculate robust standard errors (RSEs)
mod_lte_bfe_wa_rses         <- sqrt(diag(vcovHC(mod_lte_bfe_wa)))

# Display stargazer table with model results and RSEs
stargazer(
  mod_lte_bfe_wa,
  se = list(mod_lte_bfe_wa_rses),
  add.lines = list(
    c("Using Robust Standard Errors", "Yes"),
    c("Treatment = Letter-Specific", "Yes"),
    c("Block Fixed Effects Proxy (Avg. Turnout)", "Yes")),
  type = "latex",
  #type = "text")
  header = FALSE)
```

```
# Run F-test (ANOVA) to see if new model shows improved performance
ftest_ste_lte_bfe_wa        <- anova(mod_ste, mod_lte_bfe_wa, test = "F")

# Display result of F-test (ANOVA)
ftest_ste_lte_bfe_wa
```

Analysis of Variance Table

Model 1: vote ~ any_letter Model 2: vote ~ any_letter + block_average Res.Df RSS Df Sum of Sq F Pr(>F)
1 99998 7772.5
2 99997 7520.8 1 251.67 3346.3 < 2.2e-16 *** — Signif. codes: 0 '' ***0.001*** '' *0.01* '' 0.05 '.' 0.1 ' ' 1

```
# Extract and print estimated F-statistic from F-test
ftest_ste_lte_bfe_wa_fstat  <- round(ftest_ste_lte_bfe_wa$F[2], 4)

# Extract and print estimated p-value from F-test
ftest_ste_lte_bfe_wa_pvalue <- round(ftest_ste_lte_bfe_wa$`Pr(>F)`[2], 4)

# Print F-statistic and p-value from F-test
```

Table 6:

	Dependent variable:
	vote
any_letter	0.001
	(0.005)
block_average	1.000***
	(0.023)
Constant	−0.0001
	(0.002)
Using Robust Standard Errors	Yes
Treatment = Letter-Specific	Yes
Block Fixed Effects Proxy (Avg. Turnout)	Yes
Observations	100,000
R^2	0.032
Adjusted R^2	0.032
Residual Std. Error	0.274 (df = 99997)
F Statistic	1,673.167*** (df = 2; 99997)
Note:	*p<0.1; **p<0.05; ***p<0.01

```
paste("Estimated F-statistic:", ftest_ste_lte_bfe_wa_fstat)
```

[1] "Estimated F-statistic: 3346.2543"

```
paste("Estimated p-value:", ftest_ste_lte_bfe_wa_pvalue)
```

[1] "Estimated p-value: 0"

Answer: This is interesting... Our F-test suggests that incorporating our newly-created `block_average` variable into the regression *does* improve the model fit. The two main pieces of evidence for this conclusion are (1) the F-statistic, which is now practically off-the-charts (!) at a whopping 3346.2543, and (2) the p-value, which is so miniscule it registers as 0 (far below the typical significance threshold of 0.05).

4.8 Does cleverness create a bad-control?

Doesn't this feel like using a bad-control in your regression? Has the treatment coefficient changed from when you didn't include the `block_average` measure to when you did? Have the standard errors on the treatment coefficient changed from when you didn't include the `block_average` measure to when you did? Why is this OK to do?

```
# Compare simple any-letter model to our "clever workaround" any-letter model
stargazer(
  mod_ste,
  mod_lte_bfe_wa,
  se = list(mod_ste_rses, mod_lte_bfe_wa_rses),
  add.lines = list(
    c("Using Robust Standard Errors", "Yes", "Yes"),
    c("Treatment = Letter-Specific", "Yes", "Yes"),
    c("Block Fixed Effects Proxy (Avg. Turnout)", "No", "Yes")),
  type = "latex",
```

```
  #type = "text")
  header = FALSE)
```

Table 7:

	Dependent variable:	
	vote	
	(1)	(2)
any_letter	0.001	0.001
	(0.005)	(0.005)
block_average		1.000***
		(0.023)
Constant	0.085***	−0.0001
	(0.001)	(0.002)
Using Robust Standard Errors	Yes	Yes
Treatment = Letter-Specific	Yes	Yes
Block Fixed Effects Proxy (Avg. Turnout)	No	Yes
Observations	100,000	100,000
R^2	0.00000	0.032
Adjusted R^2	−0.00001	0.032
Residual Std. Error	0.279 (df = 99998)	0.274 (df = 99997)
F Statistic	0.077 (df = 1; 99998)	1,673.167*** (df = 2; 99997)
Note:		*p<0.1; **p<0.05; ***p<0.01

Answer: My initial inclination was that this approach does *seem* like using a bad control in the regression (but that was not my *final* conclusion). Taking the observed average voter turnout for each block and applying it as a RHS variable in the regression formula (`lm` model call) originally struck me as something we wouldn't want to do to estimate the treatment effect of any letter on voting turnout, since voter turnout is the dependent/outcome variable (a post-treatment variable) in this study. This approach initially felt a bit like "target encoding" in ML, where the modeler encodes a new feature with the average of the target/outcome variable to try to boost the accuracy of their model's predictions — a practice which is not always advisable and carries some risks. *However, I ultimately think our "clever workaround" is O.K. and does not constitute a bad control in this instance.* The main reason is because the documentation for the data used in this study (included above) states that the `block.num` represents "a block indicator for blocked random assignment." Assuming block random assignment (randomization) was executed cleanly, we can feel confident that average voter turnout within each block does not systematically differ across or co-vary with the various treatment/election letter groups. That keeps it from being a bad control and instead makes it a useful control variable. It's interesting to see that neither the treatment coefficient nor the RSE for `any_letter` changes within our "clever workaround" model compared to our original "simple" causal model. However, the constant term definitely changes; virtually all of the "signal" within that term is now pulled into the coefficient for `block_average` and what's left is a value that's essentially zero (i.e., not statistically significant from zero). What seems to be happening here — if I'm interpreting the coefficients in the table correctly and the context of the problem setup — is that, when we incorporate the `block_average` variable, this helps us control for existing voter behavior (or turnout tendencies, obstacles, etc.) within each "block". The `any_letter` variable shows the marginal/additional increase in voter turnout (beyond what already exists within each block) generated by sending any type of letter with election-related messaging. In other words, the `any_letter` coefficient represents the additional turnout expected *on top of* what's already expected within each block. And, as it turns out, the *blocks* — not the treatment letters — end up explaining most of

the differences in turnout. This result suggests that the marginal/additional effect of letters on turnout is quite small from a pure percentage point standpoint — only 0.15% points different from the control group with blocked fixed effects accounted for — and statistically non-significant. Granted, this is a *positive* number, but it's still small... Perhaps this is what Hill and Kousser (2015) meant by claiming it's possible to *increase* the probability that someone votes in the California primary election simply by sending them a letter in the mail. The final model coefficients *do* (technically) seem to support that claim, though it's difficult to put a lot of faith in the conclusion given the observed non-significant results. (Based on the published paper, it seems like the additional data the authors used — beyond the first 100,000 records featured in this assignment — establishes a more pronounced treatment effect with actual statistical significance.) However, I acknowledge that in tight elections, a small but *positive* anticipated turnout increase might carry a substantive degree of *practical* importance to political scientists, researchers, and politicians. Indeed, I could see the findings from this research still being significant in the "how to boost civic engagement in elections" sense, which certainly matters practically.

```
setkeyv(x = d, cols = 'block')

d[block > 0 , .(
      prop_control  = mean(treatment_f == 'Control'),
      prop_info     = mean(treatment_f == 'Election info'),
      prop_top_two  = mean(treatment_f == 'Top-two info'),
      prop_partisan = mean(treatment_f == 'Partisan')),
    keyby = .(block)] %>%
  melt(data = ., id.vars = 'block') %>%
    ggplot() +
      aes(x = block, y = value, color = variable) +
      geom_point() +
      facet_wrap(facets = vars(variable), nrow = 2, ncol = 2, scales = 'free')
```