Problem Set 4

Emanuel Mejía

19/03/2024

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: … We want to estimate the ATE of our advertising, although we might have spillovers between subjects in treatment and control, so we would be computing a naive estimator instead. And we can expect this spillover to reduce the gap between the outcomes in treatment and control (since both groups are somehow exposed to our advertising), and therefore we could be underestimating the ATE (meaning that the naive estimate we computed will be lower than the actual ATE).

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: … In this scenario we want to estimate the ATE of bonuses on productivity, but once again we could be having some spillover effects (because let's face it, once we have anyone paying the next round in the company party with their bonus, everyone will know about it). I think there will be an overestimation of the ATE since those who didn't receive the bonus might produce even less due to an unfairness feeling, so our naive estimate will be higher than the true ATE be because the gap in production will be more than it should be.

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.

```r
recycling_itt <- 500/1500 - 600/3000
```

Answer: … We know that $ITT = E[Y_i(z=1)] - E[Y_i(z=0)]$. In this case we have $E[Y_i(z=1)] = \frac{500}{1500} = \frac{1}{3}$ and $E[Y_i(z=0)] = \frac{600}{3000} = \frac{1}{5}$ and therefore $ITT = 0.1333$

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.

```r
recycling_cace <- recycling_itt/(700/1500)
```

Answer: … In order to compute $CACE$ we first establish the fraction of compliers $\alpha = ITT_D = \frac{700}{1500}$ and then we compute $CACE = \frac{ITT}{\alpha} = 0.2857$

2.3 Mike's CACE

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

```r
cace_mike <- recycling_itt/(500/1500)
```

Answer: … In case Mike is correct and the reached houses were just 500, the computed $CACE$ would be higher than the original since our denominator would be smaller, which makes sense since we would be reaching the same effect by treating less subjects. Therefore $CACE_M = 0.4$

2.4 Andy's CACE

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

```r
cace_andy <- recycling_itt/(600/1500)
```

Answer: … On the other hand if Andy were correct, the computed $CACE$ would be still larger than the original (following the same logic as before), but not as much as Mike's. For this case $CACE_A = 0.3333$

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: … By not having the correct information on the amount of real compliers we were underestimating the true effect of our treatment by 0.1143 points because we reached the same effect with less treated subjects than the ones we expected.

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: ... In case Mike were right and we focus on the *ITT* we'd still be underestimating the effect, but in this case we'd be miscalculating it by 3 times!

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.)

```
d <- data.table(
  'id' = 1:summary_table[, sum(N)],
  'count' = rep(1),
  'assignment' = rep(as.factor(c("Baseline","Treatment","Placebo")),
                 times = c(
                   summary_table[Assignment == "Baseline", sum(N)],
                   summary_table[Assignment == "Treatment", sum(N)],
                   summary_table[Assignment == "Placebo", sum(N)]
                   )
                 ),
  'treat' = rep(c(0,1,0,1,0),
                 times = c(
                   summary_table[Assignment == "Baseline" & Treated == "No", N],
                   summary_table[Assignment == "Treatment" & Treated == "Yes", N],
                   summary_table[Assignment == "Treatment" & Treated == "No", N],
                   summary_table[Assignment == "Placebo" & Treated == "Yes", N],
                   summary_table[Assignment == "Placebo" & Treated == "No", N]
                   )
                 ),
  'turnout' = rep(rep(c(1,0),5),
                   times = c(
                     round(summary_table[1, N*Turnout],0),
                     round(summary_table[1, N*(1-Turnout)],0),
                     round(summary_table[2, N*Turnout],0),
                     round(summary_table[2, N*(1-Turnout)],0),
                     round(summary_table[3, N*Turnout],0),
                     round(summary_table[3, N*(1-Turnout)],0),
                     round(summary_table[4, N*Turnout],0),
                     round(summary_table[4, N*(1-Turnout)],0),
                     round(summary_table[5, N*Turnout],0),
                     round(summary_table[5, N*(1-Turnout)],0)
                     )
                   )
)

# Summary Table
d[, .(Count = sum(count), Turnout = mean(turnout)),
     keyby = .(assignment, treat)]
```

```
## Key: <assignment, treat>
##    assignment treat Count   Turnout
##        <fctr> <num> <num>     <num>
## 1:   Baseline     0  2463 0.3008526
## 2:    Placebo     0  2108 0.3145161
## 3:    Placebo     1   476 0.3004202
## 4:  Treatment     0  1898 0.3161222
## 5:  Treatment     1   512 0.3886719
```

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.

```
compliance_rate_t <- d[assignment == "Treatment", mean(treat)]
```

Answer: … The proportion of compliers on the treatment group is $E_t[d_i(1)] = 0.21$.

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.

```
compliance_rate_p <- d[assignment == "Placebo", mean(treat)]
```

Answer: … The proportion of compliers on the placebo group is $E_p[d_i(1)] = 0.18$.

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.

```
# Chi-square
proportions_difference_test <- prop.test(x=c(512,476),
                                          n=c(512+1898, 476+2108),
                                          conf.level=0.95
                                          )
p_prop <- proportions_difference_test$p.value

# Normal N(0,1)
prop_diff <- compliance_rate_t - compliance_rate_p
p <- d[assignment%in% c("Placebo","Treatment"), mean(treat)]
n1 <- d[assignment == "Treatment", sum(count)]
n2 <- d[assignment == "Placebo", sum(count)]

Z <- prop_diff / sqrt(p*(1-p)*((1/n1) + (1/n2)))
```

Answer: … We can compute this in a couple of ways:

- First we use the prop.test function in R which computes the proportion difference using a chi-squared test statistic. The null hypothesis is $H_0 : p_1 = p_2$ (meaning both proportions are equal). The result for this test, in summary, is a p-value $p = 0.0136$ which would mean to reject the null at a 95% confidence level, or fail to reject it at a 99% confidence level.

- On the other hand we can use a Z test statistic of the following form to compare these two proportions:

  $$Z = \frac{\hat{p_1} - \hat{p_2}}{\sqrt{\hat{p}(1-\hat{p})\left(\frac{1}{n_1} + \frac{1}{n_2}\right)}}$$

  Where $\hat{p} = \frac{Y_1+Y_2}{n1_+n_2}$. Our null hypothesis in this test is again $H_0 : p_1 = p_2$. This Z-statistic is then compared to a standard normal distribution. In this case $Z = 2.5031$ which is consistent with our previous result, meaning we would reject the null at a 95% confidence level and fail to reject it at a 99% confidence level.

In summary, at a 95% confidence level (although not at 99%) we would reject that both proportions are equal.

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: ... The previous results test that compliers in both groups are equivalent (which would be a direct consequence of randomization), ensuring there's no systemic difference between both groups. In this case I would recommend to assign groups again if possible because the current assignment doesn't seem to be fully randomized (it seems highly unlikely to get this mix by mere chance if both proportions were the same). In case this is not possible, we should take these results with caution and use conservative confidence levels to state any causal effect.

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?

```
itt_p <- d[assignment == "Placebo", mean(turnout)] - d[assignment == "Baseline", mean(turnout)]
cace_estimate <- itt_p / compliance_rate_p
```

Answer: ... In this case we have a placebo $ITT = 0.0111$ which gives us a placebo $CACE = 0.0601$. and taking into account that the baseline effect is 0.3009 this doesn't seem like having NO effect.

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 estimate the CACE. Provide a short narrative using inline R code.

```
itt        <- d[assignment == "Treatment", mean(turnout)] - d[assignment == "Baseline", mean(turnout)]
cace_means <- itt / compliance_rate_t
```

Answer: ... The CACE for this model should be computed using only treatment VS control groups so in this case the $CACE = 0.1444$.

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.

```
itt_model   <- lm(turnout ~ assignment, data = d)
itt_lm <- coef(itt_model)[3]
itt_d_model <- lm(treat ~ assignment, data = d)
itt_d_lm <- coef(itt_d_model)[3]
cace_lm <- itt_lm/itt_d_lm
```

Answer: ... In this case we have a linear model to compute the same amounts that we did above, so we get again an $ITT = 0.0307$ and an $\alpha = ITT_D = 0.2124$. And finally a $CACE = \frac{ITT}{ITT_D} = 0.1444$

3.9 Data subset estimator

When a design uses a placebo group, one additional way to estimate 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.

```
complier_subset <- d[assignment %in% c("Placebo", "Treatment") & treat ==1]
cace_subset_model <- lm(turnout ~ assignment, data = complier_subset)
summary(cace_subset_model)
```

```
##
## Call:
## lm(formula = turnout ~ assignment, data = complier_subset)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.3887 -0.3887 -0.3004  0.6113  0.6996
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)          0.30042    0.02173  13.823  < 2e-16 ***
## assignmentTreatment  0.08825    0.03019   2.923  0.00354 **
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 0.4742 on 986 degrees of freedom
## Multiple R-squared:  0.008591,   Adjusted R-squared:  0.007586
## F-statistic: 8.545 on 1 and 986 DF,  p-value: 0.003545
```

Answer: ... Using this method, we compare compliers in the treatment group and compliers in the placebo group. In this case we have a significant 0.0883 treatment effect at a 99% confidence level.

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: … Even when all three are finding what would seem like a significant treatment effect, the three methods are not giving the same results, in particular the one only using placebo VS treatment compliers (#9), but this might very well be because of what we said before that there seems to be a systemic difference between those assigned to placebo and those assigned to treatment.

4 Another Turnout Question

Let us start by showing some of the features about the data. There are 3,872,268 observations. Of these, 1,648,683 identify as Democrats (42.576676 percent); 934,392) identify as Republicans (24.1303546 percent); and, 1,289,193) neither identify as Democrat or Republican (33.2929694 percent).

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.

```r
# Simple model and Robust SE
mod_simple <- lm(vote ~ any_letter, data= d)
mod_simple$vcovHC_ <- vcovHC(mod_simple)
rob_SE_sim <- sqrt(diag(mod_simple$vcovHC_))

# Regression Table
stargazer(
  mod_simple,
  type = 'latex', header=F,
  se=list(rob_SE_sim),
  title="Simple Letter Model",
  dep.var.labels = "Voting Turnout",
  column.labels = c("Any Letter"),
  order="Constant",
  covariate.labels = c("(Intercept)","Any Letter")
  )
```

Table 1: Simple Letter Model

	Dependent variable:
	Voting Turnout
	Any Letter
(Intercept)	0.093***
	(0.0002)
Any Letter	0.005***
	(0.001)
Observations	3,872,268
R^2	0.00001
Adjusted R^2	0.00001
Residual Std. Error	0.291 (df = 3872266)
F Statistic	40.801*** (df = 1; 3872266)
Note:	*p<0.1; **p<0.05; ***p<0.01

Answer: ... By implementing a model that uses any letter as a treatment we find that turnout in the control group is 0.093 and it increases by 0.005 on letter receivers with a robust standard error of 0.001, so this result proves to be statistically significant.

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.

```
mod_letter <- lm(vote ~ treatment_f, data= d)
mod_letter$vcovHC_ <- vcovHC(mod_letter)
rob_SE_let <- sqrt(diag(mod_letter$vcovHC_))

stargazer(
  mod_simple, mod_letter,
  type = 'latex', header=F,
  se=list(rob_SE_sim, rob_SE_let),
  title="Letter-Specific Model",
  dep.var.labels = "Voting Turnout",
  column.labels = c("Any Letter", "Letter-Specific"),
  order="Constant",
  covariate.labels = c("(Intercept)","Any Letter","Election", "Partisan", "Top-Two")
  )
```

Table 2: Letter-Specific Model

	Dependent variable:	
	Voting Turnout	
	Any Letter	Letter-Specific
	(1)	(2)
(Intercept)	0.093***	0.093***
	(0.0002)	(0.0002)
Any Letter	0.005***	
	(0.001)	
Election		0.005***
		(0.002)
Partisan		0.005***
		(0.001)
Top-Two		0.004***
		(0.001)
Observations	3,872,268	3,872,268
R^2	0.00001	0.00001
Adjusted R^2	0.00001	0.00001
Residual Std. Error	0.291 (df = 3872266)	0.291 (df = 3872264)
F Statistic	40.801*** (df = 1; 3872266)	13.670*** (df = 3; 3872264)
Note:		*p<0.1; **p<0.05; ***p<0.01

Answer: … When looking at specific letter messages we have the following result. A base turnout rate of 0.093 for the control group and increases from this baseline of:

- 0.005 for the Election info message with a Robust SE of 0.002.
- 0.005 for the Partisan message with a Robust SE of 0.001.
- 0.004 for the Top-two info message with a Robust SE of 0.001.

All three seem statistically significant.

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?

```
test_letter_effects <- anova(
  mod_simple,
  mod_letter,
  test = "F")

test_letter_effects
```

```
## Analysis of Variance Table
##
## Model 1: vote ~ any_letter
## Model 2: vote ~ treatment_f
##    Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1 3872266 327616
## 2 3872264 327616  2  0.017723 0.1047 0.9006
```

Answer: … For this F-Statistic we have a non-significant p-value $p = 0.9006$ meaning we're not gaining explaining power from this "broken down" model, we're not really getting different treatment effects for the different messages.

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.

```
dif_test_1 <- t.test(d[treatment_f == "Partisan", vote], d[treatment_f == "Election info", vote])
dif_test_1
```

```
##
##  Welch Two Sample t-test
##
## data:  d[treatment_f == "Partisan", vote] and d[treatment_f == "Election info", vote]
## t = 0.1305, df = 59805, p-value = 0.8962
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.003856263  0.004406390
## sample estimates:
##  mean of x  mean of y
## 0.09838448 0.09810942
```

```
dif_test_2 <- t.test(d[treatment_f == "Top-two info", vote], d[treatment_f == "Election info", vote])
dif_test_2
```

```
##
##  Welch Two Sample t-test
##
## data:  d[treatment_f == "Top-two info", vote] and d[treatment_f == "Election info", vote]
## t = -0.23204, df = 59622, p-value = 0.8165
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.004615132  0.003638048
## sample estimates:
##  mean of x  mean of y
## 0.09762088 0.09810942
```

Answer: … When comparing the effect of two letters we can do it in different ways, for instance we first do a t.test for difference in means of two different treatments.

- When comparing Partisan VS Election Info messages we get a p-value of 0.89617368736641 meaning we fail to reject the null that these two have the same effects.
- When comparing Top-two info VS Election Info messages we get a p-value of 0.81650581218432 meaning we fail to reject the null that these two have the same effects.

Another way to test it is build confidence intervals around the coefficients from the table above for every different message by taking the coefficient and adding/subtracting about two standard errors from it. By doing this we can easily see that all 3 confidence intervals contain the coefficients for the other kind of letters, meaning the same as above, we fail to reject the null that the difference between these coefficients is zero.

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?

```
block_count <- d[, .(uniqueN(block))]
```

Answer: … For this particular experiment we have 382 blocks.

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?

Answer: … In this particular model we're trying to assess the specific fixed effects for the blocks in the data. This seems like too much work because as we said there are 382 blocks and for each one of these our model is trying to determine the turnout coefficient with lower MSE while controlling for everything else (the MSE should be minimized for every variable including the treatment), getting a number that best represents turnout for each block. Since the treatment is randomized this shouldn't change the coefficient of our treatment effect, but it could narrow the Standard Errors, and also getting better estimates.

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.

```
block_avg <- d[ , .(
  block_average = mean(vote)),
  by = .(block)
  ]

d <- merge(d, block_avg, by = "block")

mod_blck_avg <- lm(vote ~ any_letter + block_average, data= d)
mod_blck_avg$vcovHC_ <- vcovHC(mod_blck_avg)
rob_SE_blc <- sqrt(diag(mod_blck_avg$vcovHC_))

test_blck_avg_effects <- anova(
  mod_simple,
  mod_blck_avg,
  test = "F")

test_blck_avg_effects
```

```
## 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 3872266 327616
## 2 3872265 315228  1     12388 152170 < 2.2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
```

Answer: ... By doing this work-around we’re already giving our model an answer to not compute the representative numbers for each block, and as we obtain an F-test significant p-value $p = 0$. Meaning this model is explaining more variation in the data and we’re indeed improving performance.

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?

```
stargazer(
  mod_simple, mod_blck_avg,
  type = 'latex', header=F,
  se=list(rob_SE_sim, rob_SE_blc),
  title="Block Average VS Simple Model",
  dep.var.labels = "Voting Turnout",
  column.labels = c("Any Letter", "Block Average"),
  order="Constant",
  covariate.labels = c("(Intercept)","Any Letter","Block Avg")
  )
```

Table 3: Block Average VS Simple Model

	Dependent variable:	
	Voting Turnout	
	Any Letter	Block Average
	(1)	(2)
(Intercept)	0.093***	−0.0002
	(0.0002)	(0.0003)
Any Letter	0.005***	0.005***
	(0.001)	(0.001)
Block Avg		1.000***
		(0.003)
Observations	3,872,268	3,872,268
R^2	0.00001	0.038
Adjusted R^2	0.00001	0.038
Residual Std. Error	0.291 (df = 3872266)	0.285 (df = 3872265)
F Statistic	40.801*** (df = 1; 3872266)	76,106.350*** (df = 2; 3872265)
Note:		*p<0.1; **p<0.05; ***p<0.01

Answer: … This does feel like using a bad control because "block average turnout" should move with the treatment in case we do have a treatment effect. Although since we're randomizing the assignment to treatment, it seems like we still have an effect of the same magnitude over each average. On the other hand, we can see that our intercept is now almost zero (although that slightly negative intercept could be catching the error on the averages, because when adding fixed effects we're doing something more difficult than just averaging) and block average's coefficient is 1, so we're doing some sort of fixed effects and perhaps we could indeed use it.

5 Optional Turnout in Dorms

5.1 Use Linear Regressions

1. Estimate the ITT using a linear regression on the appropriate subset of data. Notice that there are two `NA` in the data. Just na.omit to remove these rows so that we are all working with the same data. Given the ways that randomization was conducted, what is the appropriate way to construct the standard errors?

```
d <- na.omit(d)
dorm_model <- lm(turnout ~ treatment_group, data = d)
exp_itt <- dorm_model$coefficients[2]
summary(dorm_model)
```

```
##
## Call:
## lm(formula = turnout ~ treatment_group, data = d)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.8006  0.1994  0.1994  0.1994  0.3313
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)      0.66867    0.01162  57.521   <2e-16 ***
## treatment_group  0.13193    0.01422   9.278   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4246 on 4020 degrees of freedom
## Multiple R-squared:  0.02096,    Adjusted R-squared:  0.02072
## F-statistic: 86.08 on 1 and 4020 DF,  p-value: < 2.2e-16
```

Narrative: . . . In this case we have a very significant $ITT = 0.1319$, although we know that there's still work to do to test if this is indeed a valid causal inference.

5.2 Use Randomization Inference

1. How many people are in treatment and control? Does this give you insight into how the scientists might have randomized? As usual, include a narrative sentence after your code.

```
n_treatment <- d[treatment_group == 1, .N]
n_control   <- d[treatment_group == 0, .N]

dorm <- d[, .(students = .N), keyby = .(dormid, treatment_group)]
dorm[, .(dorms = .N), keyby = .(students, treatment_group)]
```

```
## Key: <students, treatment_group>
##     students treatment_group dorms
##        <int>           <int> <int>
##  1:        1               1     1
##  2:        2                0     4
##  3:        2                1     8
##  4:        3                0    19
##  5:        3                1    45
##  6:        4                0   298
##  7:        4                1   576
##  8:        5                0     2
##  9:        5                1     8
## 10:        6                0    10
## 11:        6                1    32
## 12:        7                0     1
```

```r
dorm[, .(dorms = .N), keyby = .(treatment_group)]
```

```
## Key: <treatment_group>
##    treatment_group dorms
##              <int> <int>
## 1:               0   334
## 2:               1   670
```

Narrative: . . . We have 2688 students in treatment and 1334 students in control. Which seems like having 1 student in control per 2 students in treatment. Although since this has been done by dormitory it seems like perhaps it’s more like 2 dorms in treatment per 1 dorm in control (by looking at the distribution in dormitories it seems like it since we have 334 in control and 670 in treatment).

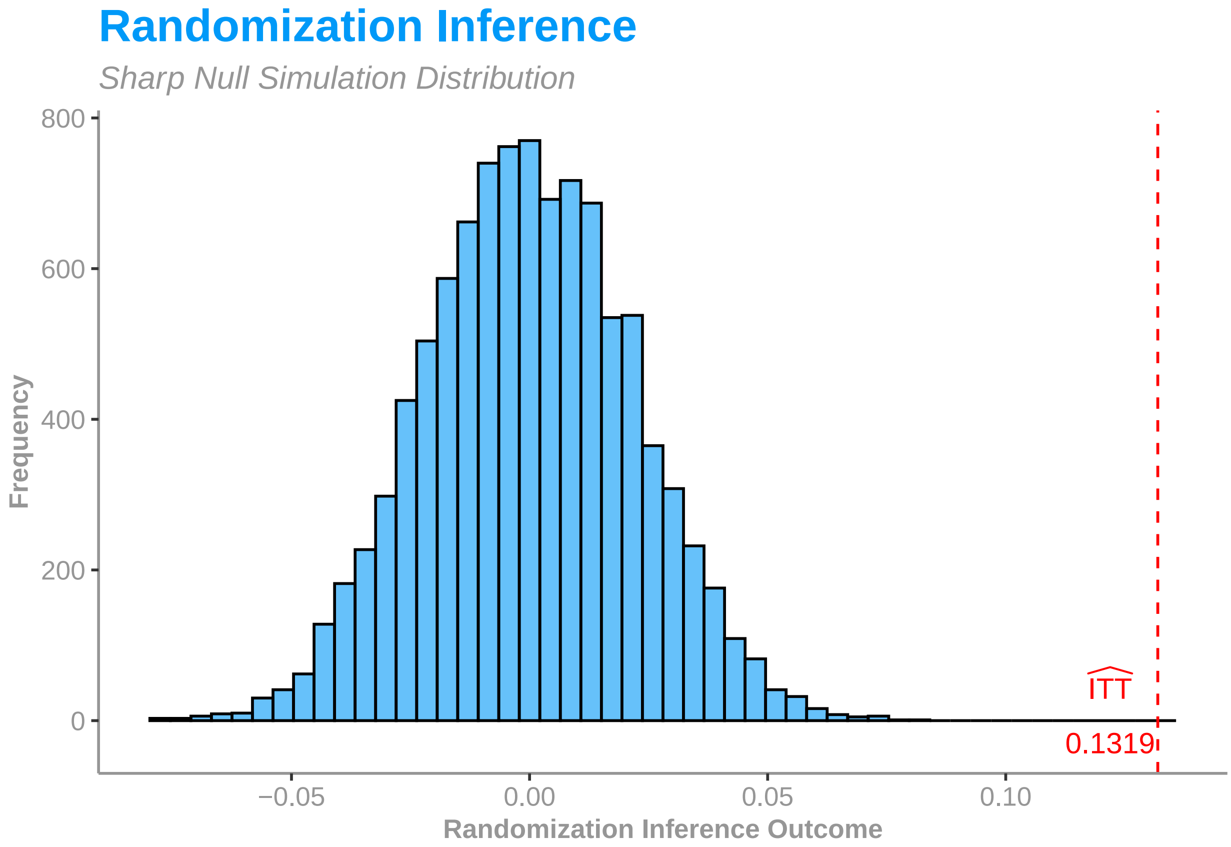
2. Write an algorithm to conduct the Randomization Inference. Be sure to take into account the fact that random assignment was clustered by dorm room.

```r
# Simulation under the sharp null
ri_simulation <- function(simulations = 10000){
  vec_itt <- NA
  for(sim in 1:simulations) {
    dorm_sampl <- dorm[, .(dormid, assignment = sample(treatment_group))]
    d_sim <- merge(d, dorm_sampl, by = "dormid")
    vec_itt[sim] <- d_sim[ , .(turnout = mean(turnout)),
                           keyby = .(assignment)][ ,diff(turnout)]
  }
  return(vec_itt)
}

#Distribution under sharp null
sharp_dist <- ri_simulation()

sim_dist_hist <-  ggplot() +
  geom_histogram(
    aes(sharp_dist),
    fill = "#0099F8",
    color="black",
    bins = 50,
    alpha = 0.6) +
  geom_vline(xintercept = exp_itt, linetype = "dashed", color = "red") +
  annotate(geom='text', x=exp_itt - 0.01, y=50,
           label= TeX("$\\widehat{ITT}", output='character'),
           parse=TRUE, color= "red", size = 4) +
  annotate(geom='text', x=exp_itt - 0.01, y=-30,
           label= round(exp_itt,4),
           parse=TRUE, color= "red", size = 4) +
  labs(
    title = "Randomization Inference",
    subtitle = "Sharp Null Simulation Distribution",
    x = "Randomization Inference Outcome",
    y = "Frequency"
  ) +
  theme_classic() +
  theme(
    plot.title = element_text(color = "#0099F8",
                              size = 17,
                              face = "bold"),
    plot.subtitle = element_text(color="#969696",
                                   size = 12,
                                   face = "italic"),
    axis.title = element_text(color = "#969696",
                              size = 10,
                              face = "bold"),
    axis.text = element_text(color = "#969696", size = 10),
    axis.line = element_line(color = "#969696")
  )

sim_dist_hist
```



3. What is the value that you estimate for the treatment effect?

```
# estimating complier rate
itt_d_mod <- lm(treated ~ treatment_group, data = d)
itt_d_dorm <- coef(itt_d_mod)[2]

dorm_room_cace <- exp_itt / itt_d_dorm
```

Narrative: . . . In this case we have a complier rate $\alpha = 0.89$ and therefore a $CACE = \frac{ITT}{ITT_D} = 0.1489$.

4. What are the 2.5% and 97.5% quantiles of this distribution?

```
dorm_room_ci <- quantile(
  sharp_dist,
  probs = c(0.025,0.975)) # there's a built-in to pull these.
```

Narrative: . . . Under the sharp null we tested above, the quantiles 2.5% and 97.5% are in the positions -0.0421 and 0.0424 respectively.

5. What is the p-value that you generate for the test: How likely is this treatment effect to have been generated if the sharp null hypothesis were true.

```r
p_value <- sum(
    abs(sharp_dist) > dorm_room_cace
  ) / length(sharp_dist)
```

Narrative: ... To find a CACE this extreme under the sharp null would be extremely unlikely with a p-value estimated at $p = 0$.

6. Assume that the leaflet (which was left in case nobody answered the door) had no effect on turnout. Estimate the CACE either using ITT and ITT_d or using a set of linear models. What is the CACE, the estimated standard error of the CACE, and the p-value of the test you conduct?

```r
se_vanilla <- coeftest(dorm_model)
se_vanilla
```

```
##
## t test of coefficients:
##
##                 Estimate Std. Error t value  Pr(>|t|)
## (Intercept)     0.668666   0.011625 57.5215 < 2.2e-16 ***
## treatment_group 0.131930   0.014220  9.2781 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
se_cluster <- coeftest(dorm_model, vcov = vcovCL(dorm_model, cluster = d[ , dormid]))
se_cluster
```

```
##
## t test of coefficients:
##
##                 Estimate Std. Error t value  Pr(>|t|)
## (Intercept)     0.668666   0.020241 33.0349 < 2.2e-16 ***
## treatment_group 0.131930   0.023271  5.6692 1.536e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
dorm_room_cace
```

```
## treatment_group
##       0.1489402
```

Narrative: ... In case we know that the leaflet had no effect, we already know that the $CACE = \frac{ITT}{ITT_D} = 0.1489$. And to compute the Standard Error we know that $SE(\widehat{CACE}) \approx \frac{SE(\widehat{ITT})}{ITT_D}$. But in this case we have to use clustered standard errors to compute $SE(\widehat{ITT}) = 0.0233$ (compare above the difference between vanilla and cluster approaches). And therefore our $SE(\widehat{CACE}) \approx 0.0263$.

7. What if the leaflet that was left actually *did* have an effect? Is it possible to estimate a CACE in this case? Why or why not?

Narrative: ... It would be very difficult to estimate a CACE because we wouldn't know the complier rate, and we'll also be assuming that the administered treatment had the same effect which is not necessarily true.