Experiments and Causality: Problem Set #4

Alex, Scott & Micah 12/9/2020

```
library(data.table)

library(sandwich)
library(lmtest)
```

```
## Loading required package: zoo

##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
library(knitr)
library(stargazer)
```

```
##
## Please cite as:

##  Hlavac, Marek (2018). stargazer: Well-Formatted Regression and Summary Statistics Tables.

##  R package version 5.2.2. https://CRAN.R-project.org/package=stargazer
```

1. Noncompliance in Recycling Experiment

Suppose that you want to conduct a study of recycling behavior. A number of undergraduate students are hired to walk door to door and provide information about the benefits of recycling to people in the treatment group. Here are some facts about how the experiment was actually carried out.

- 1,500 households are assigned to the treatment group.
- The undergrads tell you that they successfully managed to contact 700 households.
- The control group had 3,000 households (not contacted by any undergraduate students).
- The subsequent recycling rates (i.e. the outcome variable) are computed and you find that 500 households in the treatment group recycled. In the control group, 600 households recycled.

1. What is the ITT? Do the work to compute it, and store it into the object `recycling_itt`.

```
#itt = Yi(d(1))-Yi(d(0))/n
recycling_itt <- (500/1500)-(600/3000)
recycling_itt
```

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

2. What is the CACE? Do the work to compute it, and store it into the object `recycling_cace`.

```
# CACE = Yi(d(1))-Yi(d(0))/n compliers
recycling_cace <- recycling_itt/(700/1500)
recycling_cace
```

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

There appear to be some inconsistencies regarding how the undergraduates actually carried out the instructions they were given.

- One of the students, Mike, tells you that they actually lied about the the number of contacted treatment households and that the true number was 500.
- Another student, Andy, tells you that the true number was actually 600.

3. What is the CACE if Mike is correct?

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

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

4. What is the CACE if Andy is correct?

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

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

For the rest of this question, suppose that **in fact** Mike was telling the truth.

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

```
cace_dif <- cace_mike- recycling_cace
cace_dif
```

```
## [1] 0.1142857
```

> The false reporting changed the estimate for effectiveness by 0.1142857. This is a 11.4% change. Using the original CACE value you would underestimate the true CACE.

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

> Yes my answer would change. The ITT should actually stay the same since the denominator includes everyone regardless of whether they actually received the treatment or not. For this reason the **ITT would not change**. The CACE **would change** since it depends on how many people were actually treated or the compliance rate.

2. Fun with the placebo

The table below summarizes the data from a political science experiment on voting behavior. Subjects were randomized into three groups: a baseline control group (not contacted by canvassers), a treatment group (canvassers attempted to deliver an encouragement to vote), and a placebo group (canvassers attempted to deliver a message unrelated to voting or politics).

Assignment	Treated?	N	Turnout
Baseline	No	2463	0.3008
Treatment	Yes	512	0.3890
Treatment	No	1898	0.3160
Placebo	Yes	476	0.3002
Placebo	No	2108	0.3145

Evaluating the Placebo Group

1. Construct a data set that would reproduce the table. (Too frequently we receive data that has been summarized up to a level that is unuseful for our analysis. Here, we're asking you to "un-summarize" the data to conduct the rest of the analysis for this question.)

```
nrow <- sum(summary_table$N)
d <- data.table(
  id = 1:sum(summary_table$N)
)
d[1:2463, `:=` (assignment = 'Baseline', treated = 0, turnout = c(rep(0, 1723), rep(1, 740)))]
d[2464: sum(2463, 512), `:=` (assignment = 'Treatment', treated = 1, turnout = c(rep(0, 313), rep(1, 199)))]
d[2976: sum(2463, 512, 1898), `:=` (assignment = 'Treatment', treated = 0, turnout = c(rep(0, 1298), rep(1, 600)))]
d[4873 : sum(2463, 512, 1898, 476), `:=` (assignment = 'Placebo', treated = 1, turnout = c(rep(0, 334), rep(1, 143)))]
d[sum(2463, 512, 1898, 476, 1):.N, `:=` (assignment = 'Placebo', treated = 0, turnout = c(rep(0, 1445), rep(1, 663)))]
```

2. Estimate the proportion of compliers by using the data on the treatment group.

```
complier <- d[assignment == 'Treatment' & treated == 1, .N]
total <- d[assignment == 'Treatment', .N]
compliance_rate_t <- complier/total

compliance_rate_t
```

```
## [1] 0.2125363
```

3. Estimate the proportion of compliers by using the data on the placebo group.

```
compliance_rate_p <- d[assignment == 'Placebo' & treated == 1, .N]/d[assignment == 'Placebo', .N]
compliance_rate_p
```

```
## [1] 0.1845261
```

4. Are the proportions in parts (1) and (2) statistically significantly different from each other? Provide *a test* and n description about why you chose that particular test, and why you chose that particular set of data.

```
proportions_difference_test <- prop.test(
  x = c(d[assignment == 'Treatment' & treated == 1, .N], d[assignment == 'Placebo' & treated == 1, .N]),
  n = c(d[assignment == 'Treatment', .N], d[assignment == 'Placebo', .N]))

proportions_difference_test
```

```
##
##  2-sample test for equality of proportions with continuity correction
##
## data:  c(d[assignment == "Treatment" & treated == 1, .N], d[assignment ==  out of c(d[assignment == "Treatment", .N], d[assignmen
## X-squared = 5.9849, df = 1, p-value = 0.01443
## alternative hypothesis: two.sided
## 95 percent confidence interval:
##  0.005461982 0.050558438
## sample estimates:
##    prop 1    prop 2
## 0.2125363 0.1845261
```

> I decided to use a Z-test since the number of people is sufficiently large for the central limit to stand. For this reason a Z test is appropriate. I choose the subset of the data that was used in the calculation of the proportion. The x variable in the test is the number of people who complied and n is the total number of people. In this case, the results from a Z-test and a T-test should be very similar since the number of individuals is large.

5. 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?

> This tests the assumption that the rate of non-compliance does not vary accross groups. This test shows that non-compliance in fact does vary accross groups. Going forward, I would want to check and see if there were any particular causes which led to non-compliance accross groups.

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

```
itt <- (d[assignment == "Placebo" & turnout == 1, .N]/d[assignment == "Placebo", .N]) - (d[assignment == "Baseline" & turnout == 1,
itt_d <- (d[assignment == "Placebo" & treated == 1, .N]/d[assignment == "Placebo", .N])
cace_estimate <- itt/itt_d



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

```
## [1] 0.06152099
```

> No it seems like there actually is **somewhat of an effect**. If there were no effect we'd espect the estimate to be closer to 0. A p-value, a standard error and or a confidence interval would improve the ability to reason if this is a real effect.

Estimate the CACE Several Ways

7. 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.

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

```
## [1] 0.144969
```

8. 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 <- d[assignment == 'Treatment'|assignment == 'Baseline', lm(turnout ~ assignment)]
itt_d_model <- d[, lm(treated ~ assignment)]
cace_model_estimate <- coef(itt_model)[2]/coef(itt_d_model)[3] #check why 3 not 2
cace_model_estimate
```

```
## assignmentTreatment
##            0.144969
```

> Not surprisignly this gives identical results to the ITT computed above. The `cace_model_estimate` shows the causal effect of treatment on turnout.

9. 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.

```
cace_subset_model <- d[assignment != 'Baseline' & treated == 1, lm(turnout ~ assignment)]
cace_subset_model
```

```
##
## Call:
## lm(formula = turnout ~ assignment)
##
## Coefficients:
##         (Intercept)  assignmentTreatment
##             0.29979              0.08888
```

10. 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?

> The results for 7 and 8 are the same but the results for 9 are not. I think the method in 9 is giving a different result because it is utilizing the placebo set rather than the baseline. In this case there were some differences between the placebo group and the baseline that should not be there. The **difference** in the compliance rate between the placebo group and the treatment group is the source of the issue.

11. In class we discussed that the rate of compliance determines whether one or another design is more efficient. (You can review the textbook expectation on page 162 of *Field Experiments*)). Given the compliance rate in this study, which design *should* provide a more efficient estimate of the treatment effect?

> Since the compliance ratio or ITT_d is less than 50% the placebo design is more efficient for the study.

12. When you apply what you've said in part (11) against the data that you are working with, does the {placebo vs. treatment} or the {control vs. treatment} comparison produce an estimate with smaller standard errors?

```
#for baseline vs treatment
itt_model$vcovCL_val <- vcovCL(itt_model)
itt_model_test <- coeftest(x = itt_model, level = 0.95, vcov. =itt_model$vcovCL_val)
#itt_model_test
#itt_model_test[2,2]/coef(itt_d_model)[3] # pulls the SE

#for placebo vs treatment
cace_subset_model$vcovCL_val <- vcovCL(cace_subset_model)
cace_subset_model_test <- coeftest(x = cace_subset_model, level = 0.95, vcov. = cace_subset_model$vcovCL_val)
#cace_subset_model_test[2,2]
```

> The placebo vs treatment should produce smaller standard errors. The placebo design improves efficiency of the estimate thus narrowing the standard errors. However, in this case the SE is actually higher for the placebo group due to non-compliance. To show this, this is the standard error for the baseline vs treatment 0.0626615 vs placebo vs treament 0.0300995.

3. Turnout in Dorms

Guan and Green report the results of a canvassing experiment conduced in Beijing on the eve of a local election. Students on the campus of Peking University were randomly assigned to treatment or control groups.

- Canvassers attempted to contact students in their dorm rooms and encourage them to vote.
- No contact with the control group was attempted.
- Of the 2,688 students assigned to the treatment group, 2,380 were contacted.
- A total of 2,152 students in the treatment group voted; of the 1,334 students assigned to the control group, 892 voted.
- One aspect of this experiment threatens to violate the exclusion restriction. At every dorm room they visited, even those where no one answered, canvassers left a leaflet encouraging students to vote.

```
library(sandwich)
library(lmtest)
library(data.table)
d <- fread('https://ucb-mids-w241.s3-us-west-1.amazonaws.com/Guan_Green_CPS_2006.csv')
d
```

```
##       turnout treated   dormid treatment_group
##    1:       0       0  1010101               0
##    2:       0       0  1010101               0
##    3:       0       0  1010101               0
##    4:       0       0  1010102               0
##    5:       0       0  1010102               0
##   ---
## 4020:       1       1 24033067               1
## 4021:       1       1 24033068               1
## 4022:       1       1 24033068               1
## 4023:       1       1 24033068               1
## 4024:       1       1 24033068               1
```

Here are definitions for what is in that data:

- `turnout` did the person turn out to vote?
- `treated` did someone at the dorm open the door?
- `dormid` a unique ID for the door of the dorm
- `treatment_group` whether the dorm door was assigned to be treated or not

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?

```
#dont think a subset is neccessary since no placebo group, need clustered standard errors for dorm
d <- na.omit(d)
dorm_model <- d[, lm(turnout ~ treatment_group)]
dorm_model$vcovCL_val <- vcovCL(dorm_model, cluster = d[,dormid])
dorm_model_test <- coeftest(x = dorm_model, level = 0.95, vcov. = dorm_model$vcovCL_val)
dorm_model_conf <- coefci(x = dorm_model, level = 0.95, vcov. = dorm_model$vcovCL_val)
dorm_model_test
```

```
## 
## 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
```

```
dorm_model_conf
```

```
##                      2.5 %    97.5 %
## (Intercept)     0.62898177 0.7083496
## treatment_group 0.08630481 0.1775543
```

> I used robust clustered standard errors since the experiment was conducted at the dorm level rather than the individual level.

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 ususal, include a narrative setence after your code.

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

```
## [1] 2688
```

```
n_control
```

```
## [1] 1334
```

> There are 2688 people in the treatment group and 1334 in control. This gives some limitted insight into their randomization strategy. I presume that used a probability sample to shuffle people into the control or treatment group. It was most likely intentional to have more people in the treatment group.

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.

```
#unique_dorms <- d[, unique(dormid)]
#treated_dorms <- length(d[treatment_group == 1, unique(dormid)])

clustered_randomization <- function(data) {
      unique_dorms <- data[, unique(dormid)]
      n_treated_dorms <- length(data[treatment_group ==1, unique(dormid)])
      treated_ids <- sample(x = unique_dorms,
                            size = n_treated_dorms,
                            replace = FALSE)
      return(as.numeric(data[,dormid] %in% treated_ids))

}
ri_dist <- replicate(1000, d[, .(turnout_mean = mean(turnout)),
                             keyby = clustered_randomization(d)]
                     [,diff(turnout_mean)])
```

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

```
dorm_room_ate <- mean(d[treatment_group == 1, turnout]) - mean(d[treatment_group == 0, turnout])
dorm_room_ate
```

```
## [1] 0.1319296
```

> The estimated ATE is 0.1319296

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

```
dorm_room_ci <- quantile(ri_dist, probs = c(0.025, 0.975))
dorm_room_ci
```

```
##        2.5%       97.5% 
## -0.04443170  0.04287029
```

> The quantiles range from -0.0444317, 0.0428703 which includes zero. Since zero is included in the range, that means the results are not significant. This makes sense since this distribution was generated under the sharp null.

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.

```
p_value <- mean(abs(ri_dist) > dorm_room_ate)
p_value
```

```
## [1] 0
```

> This indicates that the ATE we calculated is highly significant with a zero P-value. This P-value was generated by comparing the ATE against the sharp null hpyothesis.

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?

```
 library("ivreg")
ivyreg <- d[, ivreg(turnout ~ treated |treatment_group)]
ivyreg$vcovCL_val <- vcovCL(ivyreg, cluster = d[,dormid])
ivyreg_test <- coeftest(x = ivyreg, level = 0.95, vcov. = ivyreg$vcovCL_val)
ivyreg_test
```

```
##
## t test of coefficients:
##
##             Estimate Std. Error t value  Pr(>|t|)
## (Intercept) 0.668666   0.020239 33.0390 < 2.2e-16 ***
## treated     0.148940   0.026308  5.6614 1.607e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

> I decided to use instrumental variable by two-stage least squares to greatly facilitate the ease of calculating the standard error and the p-value. The coefficient for the treatment variable is 0.1489402 with a standard error of 0.0263 and a highly significant P-value.

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?

> If the leaflet did have an effect the ITT_d or compliance rate would be nearly impossible to interpret. This would mean that people in the treatment group who did not receive the intended intervention, contact with a convasser, received a partial treatment instead. This would make it nearly impossible to know the effect of contact with a convasser. This would lead to an odd middle ground where non-compliers essentially received a partial treatment. For this reason I think the CACE would be a misleading statistic. This would lead to a bias in the CACE where its effect is overestimated.

4. Another Turnout Question

We're sorry; it is just that the outcome and treatment spaces are so clear!

Hill and Kousser (2015) report that it is possible to increase the probability that someone votes in the California *Primary Election* simply by sending them a letter in the mail. This is kind of surprising, because who even reads the mail anymore anyways? (Actually, if you talk with folks who work in the space, they'll say, "We know that everybody throws our mail away; we just hope they see it on the way to the garbage.")

Can you replicate their findings? Let's walk through them.

```
 number_rows <- 3872268 # you should change this for your answer. full points for full data

d <- data.table::fread(
  input = 'https://ucb-mids-w241.s3-us-west-1.amazonaws.com/hill_kousser_analysis_file.csv',
  nrows = number_rows)
```

(As an aside, you'll note that this takes some time to download. One idea is to save a copy locally, rather than continuing to read from the internet. One problem with this idea is that you might be tempted to make changes to this cannonical data; changes that wouldn't be reflected if you were to ever pull a new copy from the source tables. One method of dealing with this is proposed by Cookiecutter data science.)

Here's what is in that data.

- `age.bin` a bucketed, descriptive, version of the `age.in.14` variable
- `party.bin` a bucketed version of the `Party` variable
- `in.toss.up.dist` whether the voter lives in a district that often has close races
- `minority.dist` whether the voter lives in a majority minority district, i.e. a majority black, latino or other racial/ethnic minority district
- `Gender` voter file reported gender
- `Dist1-8` congressional and data districts
- `reg.date.pre.08` whether the voter has been registered since before 2008
- `vote.xx.gen` whether the voter voted in the `xx` general election
- `vote.xx.gen.pri` whether the voter voted in the `xx` general primary election
- `vote.xx.pre.pri` whether the voter voted in the `xx` presidential primary election
- `block.num` a block indicator for blocked random assignment.
- `treatment.assign` either "Control", "Election Info", "Partisan Cue", or "Top-Two Info"
- `yvar` the outcome variable: did the voter vote in the 2014 primary election

These variable names are horrible. Do two things:

- Rename the smallest set of variables that you think you might use to something more useful. (You can use `data.table::setnames` to do this.)
- For the variables that you think you might use; check that the data makes sense;

```
d <- setnames(d, c('age.bin', 'party.bin', 'in.toss.up.dist', 'minority.dist', 'Gender', 'Dist1-8',
                   'reg.date.pre.08', 'vote.xx.gen', 'vote.xx.gen.pri', 'vote.xx.pre.pri',
                   'block.num', 'treatment.assign', 'yvar'), c('age_group', 'party_bin',
                                                               'swing_district', 'majority_minority',
                                                               'gender', 'district', 'registered_pre_08','voted_20_general',
                                                               'voted_20_primary', 'voted_20_pres_primary', 'block_no', 'treatment',
dplyr::count(d, age_group, sort = TRUE)
```

```
## # A tibble: 7 x 2
##   age_group      n
##       <int>  <int>
## 1         1 883315
## 2         2 854591
## 3         3 773773
## 4         4 691890
## 5         5 426799
## 6         6 241831
## 7         0     69
```

```
dplyr::count(d, party_bin, sort = TRUE)
```

```
## # A tibble: 4 x 2
##   party_bin       n
##       <int>   <int>
## 1         2 1648683
## 2         3  956876
## 3         1  934392
## 4         4  332317
```

```
dplyr::count(d, voted, sort = TRUE)
```

```
## # A tibble: 2 x 2
##   voted       n
##   <int>   <int>
## 1     0 3510931
## 2     1  361337
```

```
dplyr::count(d, swing_district, sort = TRUE)
```

```
## # A tibble: 2 x 2
##   swing_district       n
##            <int>   <int>
## 1              0 2925362
## 2              1  946906
```

```
dplyr::count(d, treatment, sort = TRUE)
```

```
## # A tibble: 4 x 2
##   treatment           n
##   <chr>           <int>
## 1 Control       3722672
## 2 Partisan        59857
## 3 Top-two info    59854
## 4 Election info   29885
```

> What do the age groups signify? Seems like there is an issue with the treatment variable.

When you make these changes, take care to make these changes in a way that is reproducible. In doing so, ensure that nothing is positional indexed, since the orders of columns might change in the source data).

While you're at it, you might as well also modify your `.gitignore` to ignore the data folder. Because you're definitely going to have the data rejected when you try to push it to github. And every time that happens, it is a 30 minute rabbit hole to try and re-write git history.

Some questions!

1. **A 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.

```
d[, treatment_binary := ifelse(treatment != 'Control', 1, 0)]
model_simple <- d[, lm(voted ~ treatment_binary)]
coeftest(model_simple, vcov = vcovHC(model_simple, type = "HC0"))
```

```
##
## t test of coefficients:
##
##                    Estimate Std. Error  t value  Pr(>|t|)
## (Intercept)      0.09312478 0.00015062 618.2815 < 2.2e-16 ***
## treatment_binary 0.00489923 0.00078340   6.2538 4.006e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
coefci(model_simple, level = 0.95, vcov. = vcovHC, 'treatment_binary')
```

```
##                        2.5 %      97.5 %
## treatment_binary 0.003363791 0.006434678
```

> The results for this model are highly significant with a P-value well below the 0.05 significance level. This indicates that sending letters does in fact have an impact on voting turnout. I included heterogenous robust standard errors and confidence intervals in case the residuals are not normally distributed arround the regression line.

2. **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.

```
model_letters <- d[, lm(voted ~ treatment)]
coeftest(model_letters, vcov = vcovHC(model_letters, type = "HC0"))
```

```
##
## t test of coefficients:
##
##                          Estimate Std. Error  t value  Pr(>|t|)
## (Intercept)            0.09312478 0.00015062 618.2815 < 2.2e-16 ***
## treatmentElection info 0.00498464 0.00172728   2.8858 0.0039038 **
## treatmentPartisan      0.00525971 0.00122664   4.2879 1.804e-05 ***
## treatmentTop-two info  0.00449610 0.00122248   3.6779 0.0002352 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
coefci(model_letters, level = 0.95, vcov. = vcovHC)
```

```
##                              2.5 %      97.5 %
## (Intercept)            0.092829570 0.093419985
## treatmentElection info 0.001599119 0.008370165
## treatmentPartisan      0.002855505 0.007663907
## treatmentTop-two info  0.002100048 0.006892153
```

> The coefficient for each one of the treatments is actually very small with the partisan treatment having the biggest effect. The partisan treatment increases voter turnout by 0.525%. However, each coefficient is highly significant.

3. 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?

```
model_anova <- anova(model_simple, model_letters, test = "F")
model_anova
```

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

> The more complex model does not increase performance. The P-value for the more complex model is not significant and the RSS has stayed the same. This means the increased complexity in the second model does not lead to any improvements.

4. **More Specific Treatment 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.

```
model_partisan_vs_info <- d[treatment == 'Partisan' | treatment == 'Election info',lm(voted ~ treatment) ]
model_partisan_vs_info_se <- coeftest(model_partisan_vs_info, vcov = vcovHC(model_partisan_vs_info, type = "HC0"))
model_partisan_vs_info_ci <- coefci(model_partisan_vs_info, level = 0.95, vcov. = vcovHC)
model_partisan_vs_info_se
```

```
##
## t test of coefficients:
##
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)       0.09810942 0.00172070 57.0171   <2e-16 ***
## treatmentPartisan 0.00027506 0.00210779  0.1305   0.8962
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
model_partisan_vs_info_ci
```

```
##                          2.5 %     97.5 %
## (Intercept)        0.094736747 0.10148209
## treatmentPartisan -0.003856293 0.00440642
```

```
model_top_two_vs_info  <- d[treatment == "Top-two info" | treatment == "Election info", lm(voted ~ treatment)]
model_top_two_vs_info_se <- coeftest(model_top_two_vs_info, vcov = vcovHC(model_top_two_vs_info, type = "HC0"))
model_top_two_vs_info_ci <- coefci(model_top_two_vs_info, level = 0.95, vcov. = vcovHC)
model_top_two_vs_info_se
```

```
##
## t test of coefficients:
##
##                          Estimate  Std. Error t value Pr(>|t|)
## (Intercept)            0.09810942  0.00172070  57.017   <2e-16 ***
## treatmentTop-two info -0.00048854  0.00210537  -0.232   0.8165
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
model_top_two_vs_info_ci
```

```
##                              2.5 %      97.5 %
## (Intercept)            0.094736747 0.101482092
## treatmentTop-two info -0.004615161 0.003638077
```

> For `model_partisan_vs_info` the coefficient for `treatmentPartisan` is slightly larger than for `Election info`. However the results are not significant.
> For `model_top_two_vs_info` the coefficient for `Top-two info` is slightly smaller than it is for `Election info` however the results are also not significant.

> This can also be seen in the confidence interval which contains zero.

5. **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?

> There are **382 block ids**. This is a lot of blocks!

6. **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?

```
#commented it out since it crashes
#model_block_fx  <- d[,lm(voted ~ treatment_binary + as.factor(block_no))]
```

> We would be able to see if different blocks have an impact on voter turnout. Unfortunately their design includes far too many blocks and therefore this is impossible to do.

6. 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.

```
d[, mean_turnout_block := mean(voted), keyby = block_no]
model_block_average <- d[, lm(voted ~ treatment_binary+ mean_turnout_block)]
coeftest(model_block_average, vcov = vcovHC(model_block_average, type = "HC0"))
```

```
##
## t test of coefficients:
##
##                       Estimate  Std. Error  t value  Pr(>|t|)
## (Intercept)        -0.00019035  0.00030036  -0.6338    0.5262
## treatment_binary    0.00492381  0.00076824   6.4092 1.463e-10 ***
## mean_turnout_block  1.00000141  0.00343213 291.3650 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
f_test_results <- anova(model_simple, model_block_average, test = 'F')
f_test_results
```

```
## Analysis of Variance Table
##
## Model 1: voted ~ treatment_binary
## Model 2: voted ~ treatment_binary + mean_turnout_block
##    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
```

> This does seem to improve the performance of the model. The p-value for the model with all the additional blocks is highly significant. This is indicative that the block fixed effects are in fact predictive of turnout.

7. 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?

> The coefficient for the treatment did change but to a very small degree. From what I know about the blocking mechanism I do not think it's a bad control. I do not think the blocking variable is associated with both the treatment and the outcome. However it does feel kind of strange since it includes the average turnout per block which feels like it's associated with the outcome. Surprisingly the standdard errors for the model with the block average turnout did not change. I believe it is acceptable to include these values because they are not associated with both the outcome and the treatment. They are just associated with the outcome. Therefore, this does not open a back door pathway.

Consider Designs

Determine the direction of bias in estimating the ATE for each of the following situations when we randomize at the individual level. Do we over-estimate, or underestimate? Briefly but clearly explain your reasoning.

1. Suppose that you're advertising games – Among Us? – to try and increase sales, and you individually randomly-assign people into treatment and control. After you randomize, you learn that some treatment-group members are friends with control-group members IRL.

> This is an issue because the people in the treatment group will likely communicate with people in the control group. In this case I think we'd likely underestimate

> since the treatment effect will be dampened since the treatment group will likely communicate with the control group making the treatment effect far less clear.

2. As we're writing this question, end-of-year bonuses are being given out in people's companies. (This is not a concept we 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 might happen to your estimated treatment effects if people learn about the bonuses that others have received?*

> Besides this idea being horribly unfair, people who did not receive the bonus will likely interpret it as if they did something wrong. This could lead to an underestimate of the ATE since the effect would likely be the opposite if those in the control group did not know their peers received the bonus. For example, I think those in the control group will likely work harder because they would be worried about getting fired. If they didn't learn about the treatment, I think the effect would be the opposite. However, there is also the possiblity that this would lead to an overestimate. If workers get discouraged by the **random** nature of bonusses they might be incentivized to work less if the reward is the same. I think this could be biased in either direction. Basically, this treatment could be biased in either direction. The direction largely depends on how the employees react.