Experiments and Causality: Problem Set #4

Alex, Scott & Micah

12/9/2020

```
library(data.table)

library(sandwich)
library(lmtest)

library(stargazer)
library(ggplot2)
library(magrittr)

library(knitr)
```

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`.

```
recycling_itt <- 'fill this in'
```

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

```
recycling_cace <- 'fill this in'
```

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 <- 'fill this in'
```

4. What is the CACE if Andy is correct?

```
cace_andy <- 'fill this in'
```

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?

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

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.)

```
d <- data.table('fill this in')
```

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

```
compliance_rate_t <- 'fill this in'
```

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

```
compliance_rate_p <- 'fill this in'
```

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 <- 'fill this in'
```

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?

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

```
cace_estimate <- 'fill this in'
```

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        <- 'fill this in'
cace_means <- 'fill this in'
```

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   <- 'fill this in'
itt_d_model <- 'fill this in'
```

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 <- 'fill this in'
```

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?

3. Optional Turnout in Dorms

This question is optional.

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.

```
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?

```
dorm_model <- 'fill this in'
```

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 <- 'fill this in'
n_control   <- 'fill this in'
```

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

```
dorm_room_ate <- 'fill this in'
```

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

```
dorm_room_ci <- 'fill this in with a length-two vector; first number 2.5%, second number 97.5%'
```

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 <- 'fill this in'
```

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?

```
dorm_room_cace <- 'fill this in'
```

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?

4. Another Turnout Question

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

This question allows you to scope the level of difficulty that you want to take on.

- If you keep the number of rows at 100,000 this is pretty straightforward, and you should be able to complete your work on the r.datahub.
- But, the real fun is when you toggle on the full dataset; in the full dataset there are about 4,000,000 rows that you have to deal with. This is too many to work on the r.datahub. But if you're writing using `data.table` and use a docker image or a local install either on your own laptop or a cloud provider, you should be able to complete this work.

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_of_rows <- 100000
# number_of_rows <- Inf

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

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.

```
setnames(
  x = d,
  old = c("age.in.14", "Party", "Gender", "block.num", "treatment.assign", "yvar"),
  new = c("age",       "party", "gender", "block",     "treatment",        "vote")
)
```

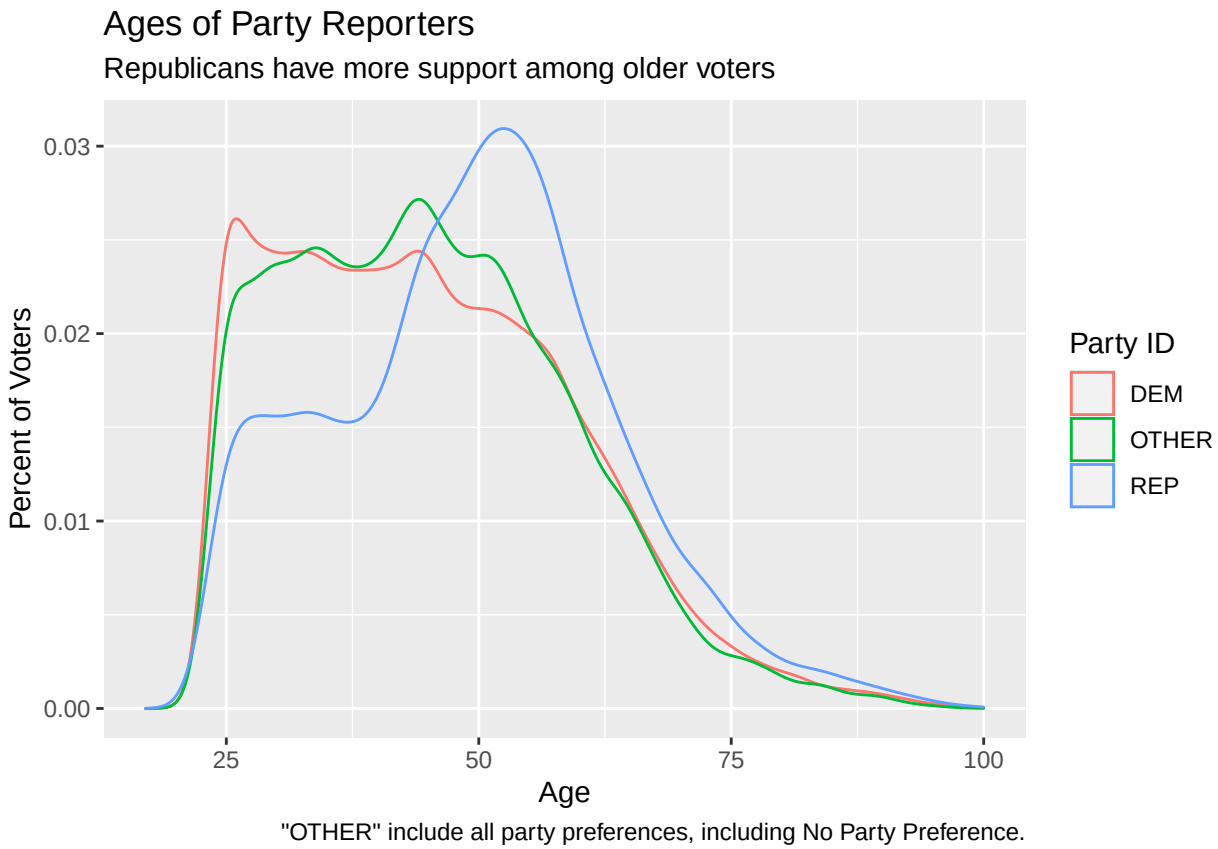
```
three_party_labeler <- function(x) {
  party <- ifelse(
    x == 'DEM', 'DEM',
    ifelse(
      x == 'REP', 'REP',
      'OTHER'))
  return(party)
}

d[ , three_party := three_party_labeler(party)]
```

```
d[ , treatment_f := factor(treatment)]
d[ , any_letter  := treatment_f != 'Control' ]
```

Let's start by showing some of the features about the data. There are 100,000 observations. Of these, 53,412 identify as Democrats (53.412 percent); 12,444) identify as Republicans (12.444 percent); and, 34,144) neither identify as Democrat or Republican (34.144 percent).

```
d %>%
  ggplot() +
  aes(x = age, color = three_party) +
  geom_density() +
  scale_x_continuous(limits = c(17, 100)) +
  labs(
    title = 'Ages of Party Reporters',
    subtitle = 'Republicans have more support among older voters',
    x = 'Age', y = 'Percent of Voters',
    color = 'Party ID',
    caption = '"OTHER" include all party preferences, including No Party Preference.'
  )
```

```
## Warning: Removed 66 rows containing non-finite values (stat_density).
```



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.

```
model_simple <- d[ , lm(vote ~ any_letter)]
model_simple$rse <- sqrt(diag(vcovHC(model_simple)))

stargazer(
  model_simple, type = 'text',
  se = list(model_simple$rse)
)
```

```
##
## ================================================
##                         Dependent variable:
##                     ---------------------------
##                                  vote
## ------------------------------------------------
## any_letter                       0.001
##                                 (0.005)
##
## Constant                       0.085***
```

```
##                                (0.001)
##
## ------------------------------------------------
## Observations                    100,000
## R2                              0.00000
## Adjusted R2                    -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
```

There is about a one-half of one percentage point effect of being sent any of these letters – people who receive letters are 0.05 percentage points more likely to vote. On the one hand, that's a really small effect; but, on the other hand, if such a letter had been sent to every person in the study, it would mean 500 more people would have voted in this election.

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(vote ~ treatment_f)]
model_letters$rse <- sqrt(diag(vcovHC(model_letters)))

stargazer(
  model_simple, model_letters,
  type = 'text',
  se = list(model_simple$rse, model_letters$rse),
  covariate.labels  = c('Any Letter', 'Election Info', 'Partisan', 'Top-Two Info', 'Intercept')
  )
```

```
##
## ========================================================================
##                                  Dependent variable:
##                      ------------------------------------------
##                                         vote
##                                (1)                    (2)
## ------------------------------------------------------------------------
## Any Letter                    0.001
##                              (0.005)
##
## Election Info                                       -0.004
##                                                     (0.010)
##
## Partisan                                             0.001
##                                                     (0.007)
##
## Top-Two Info                                         0.004
##                                                     (0.007)
##
## Intercept                   0.085***               0.085***
##                              (0.001)                (0.001)
##
## ------------------------------------------------------------------------
```

```
## Observations               100,000               100,000
## R2                         0.00000               0.00000
## Adjusted R2               -0.00001              -0.00003
## Residual Std. Error  0.279 (df = 99998)    0.279 (df = 99996)
## F Statistic         0.077 (df = 1; 99998) 0.154 (df = 3; 99996)
## ============================================================
## Note:                                 *p<0.1; **p<0.05; ***p<0.01
```

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_letters, model_simple)
model_anova
```

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

We see that the p-value that we generate from this comparison is 0.8246298. A p-value of this level is above the threshold to reject the null hypothesis that all of the letters have the same effect, and so we fail to reject this null. This suggests that the increased flexibility does **not** produce any better fit in the model; that is, while each of these treatment effects are significantly different from zero, they are not significantly different from one another. We'll test this further in the next question part.

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.

```
d[ , table(treatment_f)]
```

```
## treatment_f
##       Control Election info      Partisan  Top-two info
##         96100           825          1530          1545
```

```
model_partisan_vs_info <- d[treatment_f %in% c('Partisan', 'Election info'), lm(vote ~ treatment_f)]
model_top_two_vs_info  <- d[treatment_f %in% c('Top-two info', 'Election info'), lm(vote ~ treatment_f)]

get_rse <- function(model) {
  sqrt(diag(vcovHC(model)))
}

stargazer(
  model_partisan_vs_info, model_top_two_vs_info,
  se = list(get_rse(model_partisan_vs_info), get_rse(model_top_two_vs_info)),
  type = 'text'
)
```

```
##
## ================================================================
##                                  Dependent variable:
##                              ---------------------------
##                                         vote
##                                (1)                (2)
## ----------------------------------------------------------------
## treatment_fPartisan            0.005
##                               (0.012)
##
## treatment_fTop-two info                           0.007
##                                                  (0.012)
##
## Constant                      0.081***           0.081***
##                               (0.010)            (0.010)
##
## ----------------------------------------------------------------
## Observations                   2,355              2,370
## R2                             0.0001             0.0002
## Adjusted R2                   -0.0003            -0.0003
## Residual Std. Error      0.278 (df = 2353)  0.281 (df = 2368)
## F Statistic             0.177 (df = 1; 2353) 0.380 (df = 1; 2368)
## ================================================================
## Note:                                *p<0.1; **p<0.05; ***p<0.01
```

These are testing for differences between two different sets of treatments, and the test is reporting that there is **very** little difference between how different treatment message are affecting people's vote. All of this is to say, it *sure* seems like receiving any letter causes people to react about the same as receiving any other letter. To be honest, when we were writing this paper, and doing this analysis, this caused us to really, really closely look at whether our randomization had somehow gone wrong, although we couldn't fine any such effect.

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?

   So many blocks! There are 283 if we count them all.

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?

```
# model_block_fx  <- d[ , .(vote, any_letter, block)][ , lm(vote ~ any_letter + factor(block))]
# i thought i could get it... nope.
# Error: vector memory exhausted (limit reached?)
```

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.

```
model_block_average <- d[ , block_mean := mean(vote), by = .(block)][ , lm(vote ~ any_letter + block_mea
f_test_results      <- anova(model_block_average, model_simple, test = "F")
```

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?

   It feels *so* much like a bad control! But, if it were a bad control, then it would change what we estimate for the treatment effect. The thing is, it hasn't here.

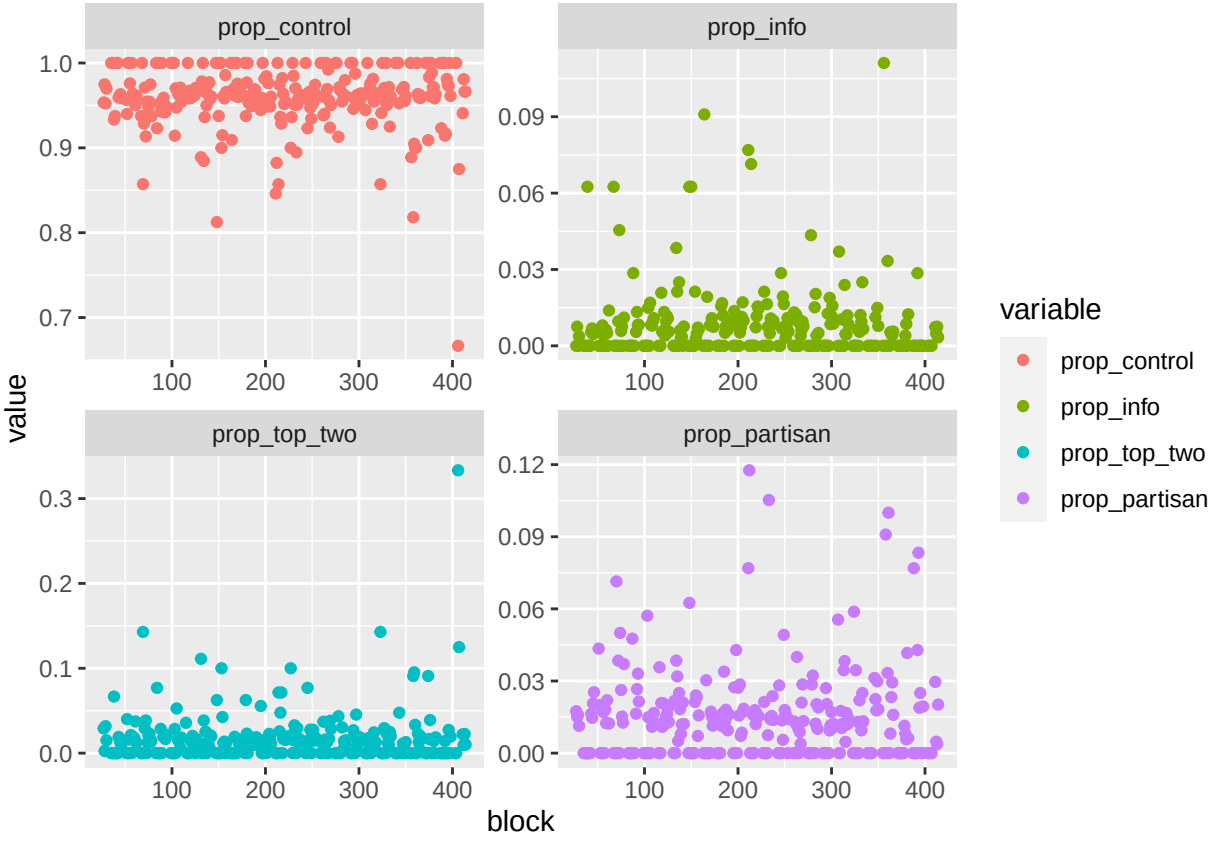
```
stargazer(
  model_simple, model_block_average,
  style = 'all',
  type = 'text'
)
```

```
##
## ================================================================================================
##                                              Dependent variable:
##                     ---------------------------------------------------------------------------
##                                                     vote
##                                  (1)                                  (2)
## ------------------------------------------------------------------------------------------------
## any_letter                        0.001                                0.001
##                                  (0.005)                              (0.004)
##                                 t = 0.277                            t = 0.333
##                                 p = 0.782                            p = 0.740
## block_mean                                                           1.000***
##                                                                      (0.017)
##                                                                     t = 57.847
##                                                                     p = 0.000
## Constant                        0.085***                             -0.0001
##                                  (0.001)                              (0.002)
##                                 t = 94.392                           t = -0.034
##                                  p = 0.000                            p = 0.973
## ------------------------------------------------------------------------------------------------
## Observations                      100,000                              100,000
## R2                                0.00000                               0.032
## Adjusted R2                      -0.00001                               0.032
## Residual Std. Error        0.279 (df = 99998)                   0.274 (df = 99997)
## F Statistic         0.077 (df = 1; 99998) (p = 0.782) 1,673.167*** (df = 2; 99997) (p = 0.000)
## ================================================================================================
## Note:                                                          *p<0.1; **p<0.05; ***p<0.01
```

   Why not? Well, one way of seeing why not is that when we designed the randomization, we made it so that the blocks were all randomized into the same proportion of treatment.

```
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')
```



This just means that **even if** including this block average were a bad control (it could be because it is measured after treatment) there is no covariance between the block number and the treatment assignment rate in it – which means that there is no problem with using this variable in the regression.

Another way of seeing this might be using another method that doesn't rely on inverting the matrix, and instead uses a different algorithm. One such method might be using Maximum Likelihood Estimation; in this particular case, it still isn't going to work because it is likely to swamp the system.

A third way is using a method described by Simen Gaure (https://cran.r-project.org/web/packages/lfe/vignettes/lfehow.pdf) and implemented in the package `lfe`.

```
model_lfe <- d[ , lfe::felm(vote ~ any_letter | factor(block))]
```

```
stargazer(
  model_simple, model_block_average, model_lfe,
  type = 'text'
)
```

```
##
## ================================================================================================
##                                              Dependent variable:
##                      ------------------------------------------------------------------------
##                                                    vote
##                                          OLS                                      felm
##                               (1)                        (2)                        (3)
## ------------------------------------------------------------------------------------------------
## any_letter                   0.001                      0.001                      0.001
##                             (0.005)                    (0.004)                    (0.004)
##
## block_mean                                             1.000***
##                                                        (0.017)
##
## Constant                   0.085***                    -0.0001
##                             (0.001)                    (0.002)
##
## ------------------------------------------------------------------------------------------------
## Observations                100,000                    100,000                    100,000
## R2                          0.00000                     0.032                      0.032
## Adjusted R2                -0.00001                     0.032                      0.030
## Residual Std. Error  0.279 (df = 99998)        0.274 (df = 99997)      0.275 (df = 99716)
## F Statistic         0.077 (df = 1; 99998) 1,673.167*** (df = 2; 99997)
## ================================================================================================
## Note:                                                          *p<0.1; **p<0.05; ***p<0.01
```

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.

2. As we're writing this question, summer 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?*