Problem Set 4

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 <- (500/1500)-(600/3000)
```

> The ITT is 0.1333333

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

```
recycling_cace <- (500/600)-(600/3000)
```

> The CACE is 0.6333333

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 <- (500/500)-(600/3000)
```

> The CACE is 0.8

4. What is the CACE if Andy is correct?

```
cace_andy <- (500/600)-(600/3000)
```

> The CACE is 0.6333333

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?

> The false reporting led us to underestimate the CACE.

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

> On the contrary, the false reporting creates no impact on the ITT because the ITT only takes into account the assignment of the subjects in the treatment and control groups - it is blind to compliance.

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(
  'assignment' = c(rep('Baseline', 1722),
                   rep('Treatment',313),
                   rep('Treatment', 1298),
                   rep('Placebo', 333),
                   rep('Placebo', 1445)),
  'treatment'   = c(rep(0,1722),
                    rep(1,313),
                    rep(0,1298),
                    rep(1,333),
                    rep(0,1445)),
  'voted'          = c(rep(1, 741),
                       rep(0, 1722-741),
                       rep(1, 199),
                       rep(0, 313-199),
                       rep(1, 606),
                       rep(0, 1298-606),
                       rep(1, 143),
                       rep(0, 333-143),
                       rep(1, 663),
                       rep(0, 1445-663))
)
```

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

```
treatment_group = d[d$assignment=='Treatment']

compliance_rate_t <- mean(treatment_group$treatment)
```

> Compliance Rate of Treatment Group is 0.1942893

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

```
placebo_group = d[d$assignment=='Placebo']

compliance_rate_p <- mean(placebo_group$treatment)
```

> Compliance Rate of the Placebo Group is 0.1872891

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 <- t.test(treatment_group$treatment, placebo_group$treatment)
proportions_difference_test
```

```
##
##  Welch Two Sample t-test
##
## data:  treatment_group$treatment and placebo_group$treatment
## t = 0.51763, df = 3344.5, p-value = 0.6048
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.01951514  0.03351549
## sample estimates:
## mean of x mean of y
## 0.1942893 0.1872891
```

> The compliance rate of the binary ‘treatment’ variable for the treatment and control groups are equivalent to the mean of these vectors. For this reason, a simple t-test can detemine if the means of these two groups are significantly different from each other by comparing their means. In this case, the two compliance rates are not significantly different from each other.

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 treatment and control have similar noncompliance rates. Given the results of this test, I feel confident that we can move forward with the understanding that this assumption is met.’

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

```
baseline_group = d[d$assignment=='Baseline']
baseline_group = baseline_group[baseline_group$treatment==0]

placebo_group = d[d$assignment=='Placebo']
placebo_group = placebo_group[placebo_group$treatment==1]

cace_estimate <- mean(placebo_group$voted) - mean(baseline_group$voted)

cace_difference <- t.test(baseline_group$voted, placebo_group$voted)
cace_difference
```

```
##
##  Welch Two Sample t-test
##
## data:  baseline_group$voted and placebo_group$voted
## t = 0.029797, df = 469.15, p-value = 0.9762
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.05742319  0.05919151
## sample estimates:
## mean of x mean of y
## 0.4303136 0.4294294
```

> The CACE of receiving the placebo is -8.8415942^{-4}, which is not statistically significant. The estimate is consistent with the assumption that the placebo has no effect on turnout.

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 estiamate the CACE.

```
itt        <- mean(treatment_group$voted) - mean(placebo_group$voted)
cace_means <- itt/compliance_rate_t

print(itt)
```

```
## [1] 0.0702602
```

```
print(cace_means)
```

```
## [1] 0.3616268
```

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.

```
t_p_all = d[d$assignment == 'Treatment' | d$assignment == 'Placebo']

itt_model   <- lm(t_p_all$voted~t_p_all$assignment)
itt_d_model <- coef(itt_model)/mean(t_p_all$treatment==1)

print(itt_d_model)
```

```
##                (Intercept) t_p_all$assignmentTreatment
##                  2.3781669                   0.2432699
```

> Treatment corresponds with a 24.32% increase in turnout among compliers as compared to the placebo group.

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.

```
t_p_compliers = d[(d$assignment == 'Treatment' | d$assignment == 'Placebo') & d$treatment == 1]

cace_subset_model <- lm(t_p_compliers$voted~t_p_compliers$assignment)

summary(cace_subset_model)
```

```
##
## Call:
## lm(formula = t_p_compliers$voted ~ t_p_compliers$assignment)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.6358 -0.4294  0.3642  0.3642  0.5706
##
## Coefficients:
##                                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)                      0.42943    0.02680  16.021  < 2e-16 ***
## t_p_compliers$assignmentTreatment  0.20635    0.03851   5.359 1.17e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4891 on 644 degrees of freedom
## Multiple R-squared:  0.04269,    Adjusted R-squared:  0.0412
## F-statistic: 28.72 on 1 and 644 DF,  p-value: 1.168e-07
```

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 in 7, 8, and 9 are not the same as their CACE equal 0.36, 0.24, and 0.21 respectively. This gives me doubt as to whether thes estimators are working well, or perhaps that the assumptions underlying these models are potentially not being met. This could provide different estimates.

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.

(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 (https://drivendata.github.io/cookiecutter-data-science/#links-to-related-projects-and-references).)

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).

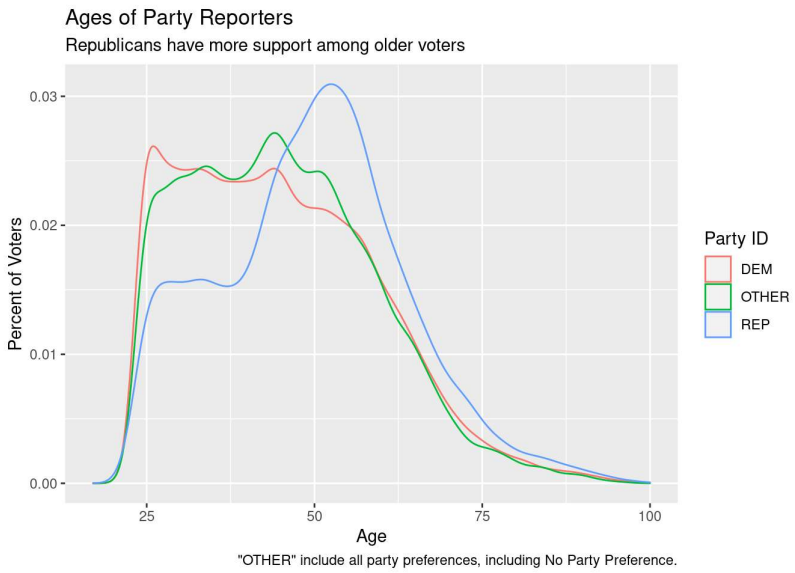
While you're at it, you might as well also modify your `.gitignore` to ignore the data folder. Because you're definitely going to have the data rejected when you try to push it to github. And every time that happens, it is a 30 minute rabbit hole to try and re-write git history.

```
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).



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.

```
mod_1_x = d$treatment != 'Control'
mod_1 = lm(d$vote ~ mod_1_x)

mod_1.vcovHC <- vcovHC(mod_1)
mod_ci <- coefci(mod_1, vcov.=mod_1.vcovHC, level=0.95)
mod_ci
```

```
##                    2.5 %     97.5 %
## (Intercept)  0.083128508 0.08665297
## mod_1_xTRUE -0.007720026 0.01024624
```

> There is no significant difference between the control group and the group of individuals who received any kind of treatment.

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.

```
mod_2 = lm(d$vote ~ d$treatment)

mod_2.vcovHC <- vcovHC(mod_2)
mod2_ci <- coefci(mod_2, vcov.=mod_2.vcovHC, level=0.95)
mod2_ci
```

```
##                                2.5 %     97.5 %
## (Intercept)               0.08312851 0.08665297
## d$treatmentElection info -0.02242419 0.01506696
## d$treatmentPartisan      -0.01280409 0.01557163
## d$treatmentTop-two info  -0.01051080 0.01807560
```

> None of the independent treatment choices have a significant treatment effect on voter turnout as compared to the control group.

3. Does the increased flexibility 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?

```
anova <- aov(d$vote~d$treatment_f, data = d)
summary(anova)
```

```
##                  Df Sum Sq Mean Sq F value Pr(>F)
## d$treatment_f     3      0 0.01198   0.154  0.927
## Residuals     99996   7772 0.07773
```

> This test yields an F-Value of 0.154 and a p-value of 0.927 which indicates that it is very likely that the variation we see is not due to the effect of the independent variable, but instead due to chance.

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.

```
part_elec <- d[(d$treatment == 'Partisan' | d$treatment == 'Election info')]
toptwo_elec <- d[(d$treatment == 'Top-two info' | d$treatment == 'Election info')]

part_elec_test <- lm(part_elec$vote ~ part_elec$treatment)
summary(part_elec_test)
```

```
##
## Call:
## lm(formula = part_elec$vote ~ part_elec$treatment)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.08627 -0.08627 -0.08627 -0.08121  0.91879
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  0.081212   0.009687   8.383   <2e-16 ***
## part_elec$treatmentPartisan 0.005062   0.012019   0.421    0.674
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2782 on 2353 degrees of freedom
## Multiple R-squared:  7.54e-05,   Adjusted R-squared:  -0.0003496
## F-statistic: 0.1774 on 1 and 2353 DF,  p-value: 0.6736
```

```
toptwo_elec_test <- lm(toptwo_elec$vote ~ toptwo_elec$treatment)
summary(toptwo_elec_test)
```

```
##
## Call:
## lm(formula = toptwo_elec$vote ~ toptwo_elec$treatment)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.08867 -0.08867 -0.08867 -0.08121  0.91879
##
## Coefficients:
##                                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)                       0.081212   0.009768   8.314   <2e-16 ***
## toptwo_elec$treatmentTop-two info 0.007461   0.012098   0.617    0.537
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2806 on 2368 degrees of freedom
## Multiple R-squared:  0.0001606,  Adjusted R-squared:  -0.0002617
## F-statistic: 0.3803 on 1 and 2368 DF,  p-value: 0.5375
```

> These estimates show no significant difference between either pair of treatments when comparing by group assignment.

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?

```
##
## SAVE YOUR CODE: before you run the next lines, because it's going
##                 to crash you if you're on the r.datahub.
##                 ... but why does it crash you?
##
#model_block_fx  <- d[ , .(vote, any_letter, block)][ , lm(vote ~ any_letter + factor(block))]
# Error: vector memory exhausted (limit reached?)
```

The block would have accounted for the fact that groups that received each type of letter might be different from one another, or experience different treatment effects based on the letter they received.

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[, block_average := .(avg_turnout_by_block = mean(vote)), by = .(block)]

blockavg_mod = lm(d$vote~d$treatment+d$block_average)

summary(blockavg_mod)
```

```
##
## Call:
## lm(formula = d$vote ~ d$treatment + d$block_average)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.49994 -0.10055 -0.06839 -0.04144  0.98597
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                -5.727e-05  1.714e-03  -0.033    0.973
## d$treatmentElection info -2.837e-03  9.589e-03  -0.296    0.767
## d$treatmentPartisan       1.880e-03  7.067e-03   0.266    0.790
## d$treatmentTop-two info   3.416e-03  7.033e-03   0.486    0.627
## d$block_average           1.000e+00  1.729e-02  57.845   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2742 on 99995 degrees of freedom
## Multiple R-squared:  0.03238,    Adjusted R-squared:  0.03234
## F-statistic: 836.6 on 4 and 99995 DF,  p-value: < 2.2e-16
```

```
anova_2 = aov(d$vote~d$treatment+d$block_average, data=d)

summary(anova_2)
```

```
##                    Df Sum Sq Mean Sq  F value Pr(>F)
## d$treatment         3      0    0.01    0.159  0.924
## d$block_average     1    252  251.67 3346.083 <2e-16 ***
## Residuals       99995   7521    0.08
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Yes, adding the blocking greatly improves the causal model as shown by the large F value and low Pr(>F) value.

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 coefficients and standard errors have become smaller as compared to when we didn't include the block_average measure. This is ok to do because it is reasonable to assume that groups of different turnout averages might respond to treatment differently, and blocking by this factor can be a good way to account for this.

```
stargazer(mod_1, mod_2, blockavg_mod, type='text')
```

```
##
## ============================================================================================
##                                                  Dependent variable:
##                         --------------------------------------------------------------------
##                                                         vote
##                                  (1)                     (2)                     (3)
## --------------------------------------------------------------------------------------------
## mod_1_x                         0.001
##                                (0.005)
##
## treatmentElection info                                  -0.004                  -0.003
##                                                        (0.010)                 (0.010)
##
## treatmentPartisan                                       0.001                   0.002
##                                                        (0.007)                 (0.007)
##
## treatmentTop-two info                                   0.004                   0.003
##                                                        (0.007)                 (0.007)
##
## block_average                                                                  1.000***
##                                                                                (0.017)
##
## Constant                       0.085***                0.085***                -0.0001
##                                (0.001)                 (0.001)                 (0.002)
##
## --------------------------------------------------------------------------------------------
## Observations                   100,000                 100,000                 100,000
## R2                             0.00000                 0.00000                  0.032
## Adjusted R2                    -0.00001                -0.00003                 0.032
## Residual Std. Error      0.279 (df = 99998)     0.279 (df = 99996)      0.274 (df = 99995)
## F Statistic             0.077 (df = 1; 99998) 0.154 (df = 3; 99996) 836.640*** (df = 4; 99995)
## ============================================================================================
## Note:                                                       *p<0.1; **p<0.05; ***p<0.01
```