

```
library(knitr)
knit_hooks$set(crop=hook_pdfcrop)
```

```
library(ri2)
library(stats)
library(tidyverse)
require(lmtest)
require(sandwich)
library(data.table)
library(stargazer)
library(tinytex)
```

```
library(ivreg)
library(kableExtra)
library(dplyr)
```

```
library(sandwich)
library(lmtest)
library(stargazer)
library(dplyr)
```

## 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).

```
##
## The downloaded binary packages are in
## /var/folders/qj/2d4k07wn6m976hlrt3_s48nc0000gn/T//RtmpczJJRV/downloaded_packages
##
## The downloaded binary packages are in
## /var/folders/qj/2d4k07wn6m976hlrt3_s48nc0000gn/T//RtmpczJJRV/downloaded_packages
```

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

```
#setting seed to 9, which makes the dataset closely resemble the given summary dataset
set.seed(9)

#Following code builds small sub-datasets which are then weaved
# into the final unified datatable.
baseline <- data.table(N = 1:2463)
baseline[, ':='(Assignment="Baseline", Treated="No")]
baseline[, Turnout:=rbinom(2463, 1, .3008)]
```

```
Treat_d1 <- data.table(N = 1:512)
Treat_d1[, ':='(Assignment="Treatment", Treated="Yes")]
Treat_d1[, Turnout:=rbinom(512, 1, .3890)]

Treat_d0 <- data.table(N = 1:1898)
Treat_d0[, ':='(Assignment="Treatment", Treated="No")]
Treat_d0[, Turnout:=rbinom(1898, 1, .3160)]

Placebo_d1 <- data.table(N = 1:476)
Placebo_d1[, ':='(Assignment="Placebo", Treated="Yes")]
Placebo_d1[, Turnout:=rbinom(476, 1, .3002)]

Placebo_d0 <- data.table(N = 1:2108)
Placebo_d0[, ':='(Assignment="Placebo", Treated="No")]
Placebo_d0[, Turnout:=rbinom(2108, 1, .3145)]

#d <- rbind(baseline, Treat_d1, Treat_d0, Placebo_d1, Placebo_d0)
# d[, .(count = .N, avg = mean(Turnout)), by=list(Assignment, Treated)]
```

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.)

```
# binding all the temp dataframes above into a unified data.table
d <- data.table(rbind(baseline, Treat_d1, Treat_d0, Placebo_d1, Placebo_d0))

# And now printing a summary to mimic the same provided to us in the problem
d[, .(N = .N, Turnout = mean(Turnout)), by=list(Assignment, Treated)]
```

```
##      Assignment Treated      N      Turnout
## 1:   Baseline      No  2463  0.2976045
## 2:   Treatment     Yes   512  0.3984375
## 3:   Treatment     No  1898  0.3187566
## 4:    Placebo     Yes   476  0.3025210
## 5:    Placebo     No  2108  0.3149905
```

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

```
# 1st solution: Compliance rate as % of compliers of total assigned folks to treatment
compliance_rate_t <- d[d[, Assignment == "Treatment"], .N, by=Treated]$N[1] /
  d[Assignment == "Treatment", .N]

compliance_rate_t

## [1] 0.2124481

# Or other solution:
# (A) Calculate total folks assigned to treatment
total_treatment <- d[Assignment == "Treatment", .N]
# (B) Calculate total compliers among those assigned to treatment
total_treatment_complier <- filter(d, (Assignment=="Treatment" & Treated=="Yes"))[, .N]

compliance_rate_t <- total_treatment_complier / total_treatment
compliance_rate_t

## [1] 0.2124481
```

We show that the total compliance rate (or proportion of compliers) as per the both approaches outlined above come to: 0.2124481

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

```
total_placebo <- d[Assignment == "Placebo", .N]
total_placebo_complier <- filter(d, (Assignment=="Placebo" & Treated=="Yes"))[, .N]

compliance_rate_p <- total_placebo_complier / total_placebo
compliance_rate_p
```

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

Adopting the same approach from sub-question 2, we calculate the compliance rate among those assigned to placebo, is 0.1842105.

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

```
proportions_difference_test <- prop.test(x = c(total_treatment_complier,
                                              total_placebo_complier),
                                         n = c(total_treatment, total_placebo))
proportions_difference_test$p.value
```

```
## [1] 0.013605
```

We chose the two-proportions z-test because we are comparing two observed proportions. We are interested in finding whether the observed proportion of compliers in treatment assignment group is equal to the observed proportion of compliers in placebo assignment. Here the null hypothesis becomes  $H_0: p_A = p_B$  (where  $p_A$  = compliers in treatment assignment and  $p_B$  = compliers in placebo assignment).

With a p-value of 0.013605 we note that we reject the null at 5% significance level and note that both proportions are statistically significantly different.

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?

One of the critical assumptions in our experiments is that the percentage of compliers in treatment is same as the percentage of compliers in placebo. However, as seen from the solution above, the compliance rates in treatment and placebo are statistically significantly different at the 5% confidence level. This could be due to a variety of reasons that should be analyzed as part of the experiment design, in order to move ahead.

We should check the three core assumptions of randomized control trials first (i.e perfect randomization, excludability and non-inference) and address if there could be breakdowns in either of these assumptions.

Most importantly, we should address if there are inherent biases that can mess with perfect randomizations. For example, did the canvassers stick to their script of not mentioning local issues with the subjects that could color their opinion on how they vote? Or we should check if the canvassers adopted any different approach while administering the non-treatment placebo to the subjects? Did the canvassers know beforehand that they were administering placebos, and hence were not motivated enough in giving the placebo to the subjects (with as much enthusiasm as they were giving treatments)?

These are some of the points we should review and possibly address before moving ahead with the experiment.

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

```
# 1st approach: using ivreg() to calculate using 2SLR approach:
d_baseline_placebo <- filter(d, (Assignment=="Placebo") | (Assignment=="Baseline"))
cace_estimate <- d_baseline_placebo[,
```

```

                                ivreg(Turnout ~ Treated, ~ Assignment)]
summary(cace_estimate)$coefficients

##              Estimate Std. Error t value      Pr(>|t|)
## (Intercept) 0.29760455 0.009295186 32.01706 6.215321e-205
## TreatedYes  0.08191145 0.070520283  1.16153 2.454812e-01
## attr(,"df")
## [1] 5045
## attr(,"nobs")
## [1] 5047

# Or using an alternate approach to come to the same solution:
placebo_itt <- coef(d_baseline_placebo[, lm(Turnout ~ as.factor(Assignment))])[2]
placebo_itt_d <- coef(d_baseline_placebo[, lm(factor(Treated) ~ Assignment))][2]

## Warning in model.response(mf, "numeric"): using type = "numeric" with a factor
## response will be ignored

## Warning in Ops.factor(y, z$residuals): '-' not meaningful for factors
placebo_itt_d

## AssignmentPlacebo
##              0.1842105

cace_estimate <- placebo_itt / placebo_itt_d
cace_estimate

## as.factor(Assignment)Placebo
##              0.08191145

```

We see that the contact by canvassers increased turnout among placebo compliers by 0.0819114 percentage points. Although, this is subject to considerable sampling uncertainty. This still leads us to believe that the placebo is having some impact on ATE and is not zero (and there is perhaps something wrong with the design of the experiment that needs to be rectified).

## Estimate the CACE Several Ways

- Using a difference in means (i.e. not a linear model), compute the ITT using the appropriate groups' data. Then, divide this ITT by the appropriate compliance rate to produce an estimate of the CACE.

```

# an alternate approach (NOT ASKED IN THE QUESTION HERE):
# d_baseline_treatment <- filter(d, (Assignment=="Treatment") | (Assignment=="Baseline"))
# itt <- coef(d_baseline_treatment[, lm(Turnout ~ as.factor(Assignment))])[2]
# itt

# Approach asked for here (difference in means and then dividing by
# take-out rate to calculate CACE:
itt <- d[Assignment == "Treatment", mean(Turnout)] -
  d[Assignment == "Baseline", mean(Turnout)]
cace_means <- itt / compliance_rate_t
cace_means

## [1] 0.1792442

```

Calculated ITT and CACE: 0.0380801 and 0.1792442.

- 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.

```
d_baseline_treatment <- filter(d, (Assignment=="Treatment") | (Assignment=="Baseline"))

itt_model <- coef(d_baseline_treatment[, lm(Turnout ~ as.factor(Assignment))])[2]
itt_model
```

```
## as.factor(Assignment)Treatment
##                                0.0380801
```

```
itt_d_model <- coef(d_baseline_treatment[, lm(factor(Treated) ~ Assignment))][2]
```

```
## Warning in model.response(mf, "numeric"): using type = "numeric" with a factor
## response will be ignored
```

```
## Warning in Ops.factor(y, z$residuals): '-' not meaningful for factors
```

```
itt_d_model
```

```
## AssignmentTreatment
##                0.2124481
```

```
itt_model / itt_d_model
```

```
## as.factor(Assignment)Treatment
##                                0.1792442
```

This is an alternate approach of calculating Causal ATE. We see that using the 1st stage and 2nd stage linear models (which is equivalent to using the 2-stage linear regression model or a reduced form model approach) the answer is exactly same as the earlier approach of calculating CACE via a difference in means approach. In both approaches, we obtain the CACE as 0.1792442.

9. When a design uses a placebo group, one additional way to estimate the CACE is possible – subset to include only compliers in the treatment and placebo groups, and then estimate a linear model. Produce that estimate here.

```
d_treatment_placebo <- filter(d, ((Assignment=="Treatment") |
                                   (Assignment=="Placebo")) & (Treated=="Yes"))
```

```
cace_subset_model <- d_treatment_placebo[, ivreg(Turnout ~ Assignment)]
cace_subset_model
```

```
##
## Call:
## ivreg(formula = Turnout ~ Assignment)
##
## Coefficients:
##      (Intercept)  AssignmentTreatment
##           0.30252             0.09592
```

```
cace_subset_model <- d_treatment_placebo[, lm(Turnout ~ Assignment)]
```

```
summary(cace_subset_model)
```

```
##
## Call:
## lm(formula = Turnout ~ Assignment)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
```

```
## -0.3984 -0.3984 -0.3025  0.6016  0.6975
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      0.30252    0.02181  13.874 < 2e-16 ***
## AssignmentTreatment 0.09592    0.03029   3.167  0.00159 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4757 on 986 degrees of freedom
## Multiple R-squared:  0.01007,    Adjusted R-squared:  0.009063
## F-statistic: 10.03 on 1 and 986 DF,  p-value: 0.00159
```

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?

Step 9 is computing the average treatment effect on the compliers (which is called as ATET a.k.a ATE on the treated). Whereas, the CACE is the complier ATE with some additional noise due to inclusion on non-compliers. However, the placebo design ATET allows us to get unbiased estimates of the CACE (by increasing the precision and reducing standard errors).

In theory, both approaches should give us the similar estimates of CACE (in perfectly randomized and administered experiments where the placebo design is not impacting the treatment group and vice versa). In this case, however, we see that the ATET is much different from the treatment CACE. This was expected, as we saw from the proportions difference test that the compliance rates in treatment and placebo groups were significantly different at the 5% confidence level.

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?

Conventional designs are preferred when compliance rate of compliers in treatment are greater than 50%. In this case, since we have complier rates at 21% (well below the 50% threshold), we should prefer the placebo design approach for a more efficient estimate of the treatment effect.

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?

```
#computing std error of control vs treatment using the ivreg() model
d_baseline_treatment <- filter(d, (Assignment=="Treatment") | (Assignment=="Baseline"))

cace_estimate <- d_baseline_treatment[,
                                     ivreg(Turnout ~ Treated, ~ Assignment)]
summary(cace_estimate)

##
## Call:
## ivreg(formula = Turnout ~ Treated | Assignment)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.4768 -0.2976 -0.2976  0.7024  0.7024
##
## Coefficients:
```

```

##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.297605   0.009372  31.755  < 2e-16 ***
## TreatedYes   0.179244   0.062728   2.857  0.00429 **
##
## Diagnostic tests:
##               df1  df2 statistic p-value
## Weak instruments    1 4871   664.140 <2e-16 ***
## Wu-Hausman         1 4870    2.225   0.136
## Sargan              0  NA         NA     NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4651 on 4871 degrees of freedom
## Multiple R-Squared:  0.0003125,    Adjusted R-squared:  0.0001072
## Wald test: 8.165 on 1 and 4871 DF,  p-value: 0.004288

```

In the {control vs. treatment} design, we are calculating a regular version of CACE. In the {placebo vs. treatment} we are calculating a version of ATET. We know that the ATET provides a shrinkage in standard error due to the elimination of never-takers. Therefore, standard error of conventional CACE should be  $1/(\text{compliance rate})^{(1/2)}$  more than the standard error of ATET.

As seen in the illustrative calculations, we see precisely that (std error of CACE is 0.0627 and std error of ATET is 0.03029).