Problem Set 4: Linear Regression and Randomized Experiments

GOV 2003

Due at 11:59 pm (ET) on Oct 6, 2021

Instruction

Before you begin, please read the following instructions **carefully**:

- **No late submission is allowed** without prior approval from the instructors.
- **All answers should be typed up**. We recommend the use of `Rmarkdown`. A `Rmarkdown` template for this problem set is provided. Answers to analytical solutions should also be typed up.
- **A PDF copy of your answer** including your computer code should be uploaded to `Gradescope` before the deadline. **Do not submit the markdown file itself**.

Introduction

This problem set consists of two parts. In `Part A`, we will conduct regression analysis with a **robust variance estimator** and then will incorporate the **covariates** into our model. In `Part B`, we will investigate regression analysis for a randomized experiment under both **block** and **cluster** randomization.

For each part, we will investigate each of the two field experiments conducted in India by Gaikwad and Nellis (2021) which studies **the cause of underrepresentation of internal migrants**. In Treatment 1, to test whether bureaucratic hassle costs of updating voter registration cause this shortfall, the authors randomize the intervention of a door-to-door facilitation campaign to help migrants obtain a local voter identification card. In Treatment 2, to test the "supply side" hypothesis that political elites ignoring registered migrants causes this shortfall, the authors randomly inform the politicians that a voter registration drive had been carried out among internal migrant communities and estimate its impact on migrants' campaign exposure. You may read the "Research Design" section (pp. 7-9) for more details. (Note that the experimental sample is comprised of the individuals who wished to register to vote in their destination cities and the conclusion thus holds for this population of interest.)

Part A: Regression analysis

In this part, we will conduct regression analysis for Treatment 1 from Gaikwad and Nellis (2021). Here, we will compare "standard" variance estimator with the robust variance estimator and then incorporate covariates to our model.

Setup A

Let Y_i denote a binary outcome indicating whether migrant i voted in city in the 2019 Indian national elections (Lok Sabha elections) — $Y_i = 1$ if voted, and 0 if not. Let $T1_i$ denote a binary treatment of door-to-door facilitation campaign to help migrants obtain a local voter identification card — $T1_i = 1$ if treated, and 0 if not. Also, we have individual-level covariates, $\mathbf{X}_i$, which consists of not_voted_previously, female, age, muslim, sc_st, primary_edu, income, married, length_residence, and owns_home (see the details below). We will use the data frame named `data1` from `GaikwadNellis2021.RData` for this part.

Name	Description
id	a unique ID of migrants
voted_2019	= 1 if voted in 2019 elections
t1	= 1 if helped voter registration
not_voted_previously	= 1 if haven't voted previously
female	= 1 if self-identified female
age	age
muslim	= 1 if Muslim
sc_st	caste group, SC/ST
primary_edu	= 1 if received primary education
income	monthly household income in Rupees (INR 000s)
married	= 1 if currently married
length_residence	length of residence in current city
owns_home	= 1 if owns home

Question 1 (5 pts; 1pt for (a), 2pts for each (b) and (c))

(a) Write down a mathematical expression of the linear model without the covariates (that is, with just treatment as the only independent variable). Explain what each coefficient means. Fit the model and briefly discuss the results for each coefficient.

Hint: Recall how the coefficients can be expressed by the potential outcomes notation (see lecture slides p.4).

(b) Estimate the variance of the OLS estimator (i.e., the slope in your previous model) using each of two estimators — the standard variance estimator and robust variance estimator — and compare the results. Explain the reason why these two estimates are different. Which variance estimator would you prefer?

(c) Now, we will add the centered covariates, $\widetilde{\mathbf{X}}_i = (\mathbf{X}_i - \overline{\mathbf{X}})$, to our model. Write down a mathematical expression of the OLS estimator with full interactions. Compute the centered

covariates using `data1` and fit the fully interacted model. Estimate the variance of the OLS estimator from fully interacted model using robust variance estimator. Compare the results with the robust variance estimates from (b). Briefly discuss the difference.

Answer 1

(a)

$$Y_i = \alpha + \tau T1_i + \epsilon_i$$

- Intercept $\alpha = \mathbb{E}[Y_i(0)]$ average control outcome.
- Slope $\tau = \mathbb{E}[Y_i(1) - Y_i(0)]$ the PATE.
- Error is deviation for control PO + treatment effect heterogeneity.

```
load("GaikwadNellis2021.RData")
model <- lm("voted_2019 ~ t1", data = data1)
model$coefficients
```

```
## (Intercept)          t1
##   0.1775701   0.2081442
```

(b)

```
# Estimates using "standard" variance estimator
# Note that I'm taking sqrt() to get SE instead of variance below:
## Option 1: Using lm()
summary(model)$coefficients[,"Std. Error"]
```

```
## (Intercept)          t1
##  0.01336897  0.01899640
```

```
# Or
sqrt(diag(vcov(model)))
```

```
## (Intercept)          t1
##  0.01336897  0.01899640
```

```
## Option 2: Using estimatr::lm_robust() (Note that this shows SE = sqrt(var))
library(estimatr)
lm_robust(as.formula("voted_2019 ~ t1"), data = data1, se_type = "classical")
```

```
##              Estimate Std. Error  t value     Pr(>|t|)  CI Lower  CI Upper   DF
## (Intercept) 0.1775701 0.01336897 13.28225 9.928388e-39 0.1513524 0.2037878 2118
## t1          0.2081442 0.01899640 10.95703 3.256582e-27 0.1708906 0.2453977 2118
```

```
# Estimates using robust variance estimator (HC2)
# Note that I'm taking sqrt() to get SE instead of variance below:
## Option 1: Using sandwich::vcovHC
sqrt(diag(sandwich::vcovHC(model, type = "HC2")))
```

```
## (Intercept)          t1
##  0.01168814  0.01903901
```

```r
## Option 2: Using estimatr::lm_robust() (Note that this shows SE = sqrt(var))
lm_robust(as.formula("voted_2019 ~ t1"), data = data1, se_type = "HC2")
```

```
##               Estimate Std. Error  t value     Pr(>|t|)  CI Lower  CI Upper   DF
## (Intercept) 0.1775701 0.01168814 15.19233 1.476260e-49 0.1546487 0.2004915 2118
## t1          0.2081442 0.01903901 10.93251 4.207034e-27 0.1708071 0.2454813 2118
```

```r
# Check balance n1, n0
n1 <- sum(data1$t1)
n0 <- sum(1-data1$t1)
n1 == n0
```

```
## [1] FALSE
```

Recall that "standard" variance estimator is biased (bias could be either positive or negative) whereas HC2 estimator is unbiased for $\mathbb{V}(\hat{\tau})$, and thus HC2 estimator is preferred.

(c)

$$\left(\widehat{\tau}_{\text{full}}, \widehat{\alpha}, \widehat{\beta}, \widehat{\gamma}\right) = \underset{(\alpha,\tau,\beta,\gamma)}{\arg\min} \sum_i \left(Y_i - \alpha - \tau T1_i - \widetilde{\mathbf{X}}_i'\beta - D_i\widetilde{\mathbf{X}}_i'\gamma\right)^2$$

```r
data1_centered <- data1
for (i in 4:13) {
  data1_centered[,i] <- data1_centered[,i] - mean(data1_centered[,i])
}
model_full <- lm("voted_2019 ~ t1 +
                 not_voted_previously + female +
                 age + muslim + sc_st +
                 primary_edu + income + married +
                 length_residence + owns_home +
                 t1*not_voted_previously + t1*female +
                 t1*age + muslim + t1*sc_st +
                 t1*primary_edu + t1*income + t1*married +
                 t1*length_residence + t1*owns_home", data = data1_centered)
# Abbreviated version (same result):
model_full <- lm("voted_2019 ~ . -uniqueid + t1*(. -uniqueid)", data = data1_centered)
model_full$coefficients["t1"]
```

```
##        t1
## 0.2033119
```

```r
# Estimates using robust variance estimator (HC2)
# Note that I'm taking sqrt() to get SE instead of variance below:
## Option 1: Using sandwich::vcovHC
sqrt(diag(sandwich::vcovHC(model_full, type = "HC2"))["t1"])
```

```
##         t1
## 0.01889695
```

```r
## Option 2: Using estimatr::lm_robust() (Note that this shows SE = sqrt(var))
model_full <- lm_robust(as.formula("voted_2019 ~ . -uniqueid + t1*(. -uniqueid)"), data = data1_centered)
tidy(model_full) %>% filter(term == "t1")
```

```
##   term  estimate  std.error statistic      p.value  conf.low conf.high   df
## 1   t1 0.2033119 0.01889695  10.75898 2.576507e-26 0.1662531 0.2403706 2098
##      outcome
## 1 voted_2019
```

OLS estimator with fully interacted model ($\widehat{\tau}_{\text{full}}$) has smaller variance.

Part B: Regression analysis with block and cluster

In this part, we will switch to Treatment 2 from Gaikwad and Nellis (2021) where the treatment is block randomized at the polling station level (that is, this is a block cluster design, with individuals in clusters of polling stations). Blocks of polling stations are determined by the city and the number of respondents in clusters (see Figure A9 below). We will first investigate the correct analysis of block randomized trials and then incorporate cluster randomized trials using `data2`.

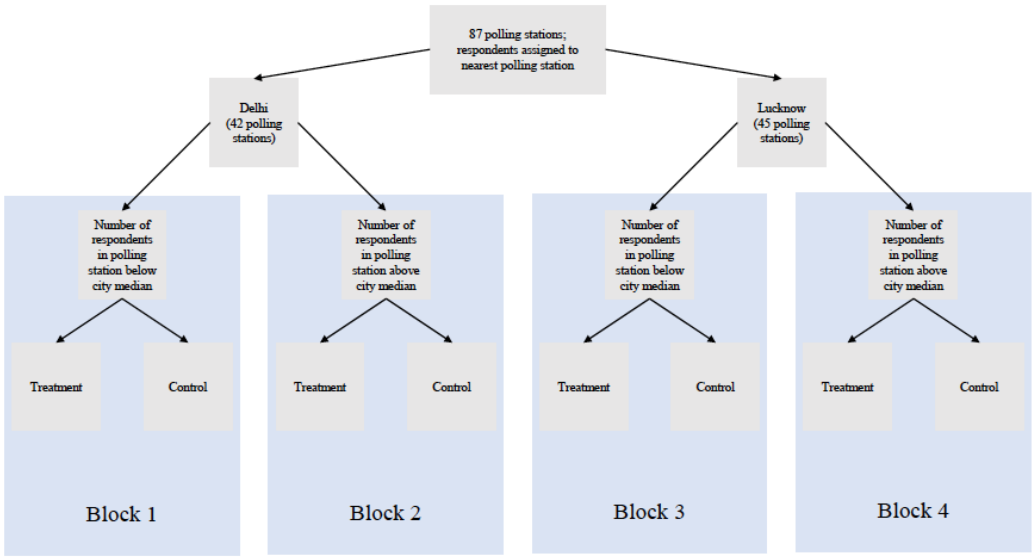

Figure A9: Flow diagram of T2 randomization blocks. Clusters (polling stations) are assigned to T2 treatment or control within four blocks.

block	# treated polling stations	# control polling stations
1	10	11
2	11	10
3	11	12
4	11	11

Question 2 (5 pts; 1pt for (a), 2pts for each (b) and (c))

To motivate the correct regression analysis of block randomized trials, we will do a thought experiment in this question. Please write your answer in words and mathematical expressions for this question. You don't have to fit the model using computational codes.

Suppose we have only two blocks, Block 2 and Block 4 from Figure A9, and we are interested in a polling station level inference. Let Y_k be an outcome variable of polling station k, which measures the campaign exposure of migrants in the polling station k. For simplicity, assume that the number of individual samples within each polling station is identical for those two blocks in this question (note that this might not be the case for Question 3). Let $T2_k$ denote a binary treatment variable of the polling station k — $T2_k = 1$ if the politicians in the neighborhood are informed that a voter registration drive had been carried out among internal migrant communities. We do not consider a covariate adjustment for this question.

Hint: Review the lecture slides p.15-16.

(a) Using a naive approach, let's consider the OLS estimator with the block FEs as below:

$$(\widehat{\tau}_{\text{b,fe}}, \widehat{\alpha}_1, \widehat{\alpha}_2) = \underset{(\tau,\alpha_1,\alpha_2)}{\arg\min} \sum_k \left(Y_k - \tau T2_k - \alpha_1 S_{k1} - \alpha_2 S_{k2}\right)^2$$

where $S_{k1} = 1$ if polling station k is in Block 2, and $S_{k2} = 1$ if polling station k is in Block 4. Under what conditions does $\widehat{\tau}_{\text{b,fe}}$ consistent for PATE? Briefly explain the implication and plausibility of each condition in this context.

(b) Write down a mathematical expression of weighted OLS estimator (similar to the expression in the previous part) where the weights are inverse of the propensity score. Briefly explain why this would be guaranteed as a correct analysis. Note that the weights should be the inverse of the probability of the **observed treatment** for a particular polling station, so $1/\Pr(T2_k = 1)$ for treated polling stations and $1/\Pr(T2_k = 0)$ for the control polling stations.

(c) Write down a mathematical expression of OLS estimator (similar to the previous two parts) assuming the model of fully interacted block FEs with treatment. Briefly explain why this would be guaranteed as a correct analysis.

Answer 2

(a) If at least one of the following conditions holds, $\widehat{\tau}_{\text{b,fe}}$ would be consistent for PATE:

- Propensity scores are equal across blocks: Since $11/(11+10) \neq 11/(11+11)$, this condition cannot be satisfied.
- ATEs are equal across strata: Since the two blocks are from different cities we may suspect heterogeneous effects across strata.

(b)

$$(\widehat{\tau}_{\text{b,fe,w}}, \widehat{\alpha}_1, \widehat{\alpha}_2) = \underset{(\tau,\alpha_1,\alpha_2)}{\arg\min} \sum_k w_k \left(Y_k - \tau T2_k - \alpha_1 S_{k1} - \alpha_2 S_{k2}\right)^2$$

The weights are as below:

$$w_k = \left\{\frac{1}{p_1} T2_k + \frac{1}{1-p_1}\left(1 - T2_k\right)\right\} S_{k1} + \left\{\frac{1}{p_2} T2_k + \frac{1}{1-p_2}\left(1 - T2_k\right)\right\} S_{k2}$$

where $p_1 \frac{\sum_k S_{k1} T2_k}{\sum_k S_{k1}}$ and $p_2 \frac{\sum_k S_{k2} T2_k}{\sum_k S_{k2}}$. That is, the number of treated polling stations in block j divided by the number total polling stations in block j for each $j \in \{1, 2\}$.

Recall that the OLS estimator with the block fixed effects converges to a weighted average of block-specific effects (τ_j):

$$\widehat{\tau}_{\text{b,fe}} \xrightarrow{p} \frac{\sum_{j=1}^{J} \omega_j \tau_j}{\sum_{j=1}^{J} \omega_j} \quad \text{where} \quad \omega_j = w_j p_j (1 - p_j)$$

The weights we specified above will cancel out p_j and $(1 - p_j)$ of the ω_j and thus would converge to PATE.

(c)

$$(\widehat{\tau}_{\text{b,fe,full}}, \widehat{\alpha}_1, \widehat{\alpha}_2, \widehat{\gamma}) = \underset{(\tau, \alpha_1, \alpha_2, \gamma)}{\arg\min} \sum_k \left\{ Y_k - \tau T2_k \frac{S_{k2}}{q_2} - \alpha_1 S_{k1} - \alpha_2 S_{k2} - \gamma T2_k \left(S_{k1} - S_{k2} \frac{q_1}{q_2} \right) \right\}^2$$

where $q_1 = \frac{\sum_k S_{k1}}{\sum_k (S_{k1} + S_{k2})}$ and $q_2 = \frac{\sum_k S_{k2}}{\sum_k (S_{k1} + S_{k2})}$ — the proportion of polling stations in each block.

Consider the following regression model

$$Y_k = \alpha_1 S_{k1} + \alpha_2 S_{k2} + \tau_1 T2_k S_{k1} + \tau_2 T2_k S_{k2} + \epsilon_k.$$

Observe that this is equivalent to running separate Y_k on each blocks (i.e., $Y_k = \alpha_1 + \tau_1 T2_k + \epsilon_k$ if $S_{k1} = 1$ and $Y_k = \alpha_2 + \tau_2 T2_k + \epsilon_k$ if $S_{k2} = 1$). Thus $\hat{\tau}_1$ and $\hat{\tau}_2$ are consistent for within-block ATE. Note that transforming the parameter τ_2 to $\tau = q_1 \tau_1 + q_2 \tau_2$, with inverse transformation $\tau_2 = (\tau - q_1 \tau_1)/q2$, yields the fully interacted model specified above. Thus $\widehat{\tau}_{\text{b,fe,full}}$ is identical to $q_1 \widehat{\tau}_1 + q_2 \widehat{\tau}_2$ and therefore is consistent for $\tau_{\text{b}} = q_1 \tau_1 + q_2 \tau_2$ from Neyman's approach.

Setup B

Let $i \in \{1, \ldots, m_k\}$ with $\sum_k = 1969$ denote the individual unit in cluster k, where $k \in \{1, \ldots, 87\}$ denote the polling station (cluster), and $j \in \{1, 2, 3, 4\}$ denote the block. Let Y_{ik} denote a binary outcome indicating whether migrant i in cluster k voted in city in the 2019 Indian national elections. Let $T2_{ik}$ denote a binary treatment of informing the politicians that a voter registration drive had been carried out among internal migrant communities — $T2_{ik} = 1$ if cluster k treated, and 0 if not. Also, we have individual-level covariates, $\mathbf{X}_i$, which consists of politicians_visits, female, age, muslim, sc_st, primary_edu, income, married, length_residence, and owns_home (see the details below). We will use the data frame named `data2` from `GaikwadNellis2021.RData` for this part.

Name	Description
id	a unique ID of migrants
block	a unique ID of blocks
cluster	a unique ID of polling stations
voted_2019	= 1 if voted in 2019 elections
t2	= 1 if informed politicians
politician_visits	= 1 if politicians visited basti in the past year
female	= 1 if self-identified female
age	age
muslim	= 1 if Muslim
sc_st	caste group, SC/ST
primary_edu	= 1 if received primary education
income	monthly household income in Rupees (INR 000s)
married	= 1 if currently married
length_residence	length of residence in current city
owns_home	= 1 if owns home

Question 3 (5 pts; 1pt for (a), 2pts for each (b) and (c))

Hint: You can use `lm_robust` function from `estimatr` package.

(a) To address the block design, write down a mathematical expression of weighted OLS estimator for individual-level inference under cluster design where the weight is inverse of the propensity score (as in question 2(b)). Compute the weights using `data2`. Fit the model and briefly discuss the results.

Hint: Be sure to use the correct robust variance estimator for this design throughout the questions (a) - (c).

(b) Now, we will add the individual-level covariates to the model. Write down a mathematical expression of the OLS estimator with (uninteracted) centered covariates. Fit the model and briefly discuss the results.

(c) Alternatively, consider a regression with fully interacted covariates. Write down a mathematical expression of the OLS estimator with interacted centered covariates. Fit the model and briefly discuss the results.

Answer 3

(a) Observe that $T2_{ik} = T2_k$ since treatment assignment is at cluster level.

$$(\widehat{\tau}, \widehat{\alpha}_1, \widehat{\alpha}_2, \widehat{\alpha}_3, \widehat{\alpha}_4) = \underset{(\tau,\alpha_1,\alpha_2,\alpha_3,\alpha_4)}{\arg\min} \sum_{k=1}^{87} \sum_{i=1}^{m_k} w_{ik} \left(Y_{ik} - \tau T2_k - \sum_{j=1}^{4} \alpha_1 S_{kj} \right)^2$$

Here, the weight w_{ik} is

$$w_{ik} = \sum_{j=1}^{4} \left\{ \frac{1}{p_j} T2_k + \frac{1}{1-p_j}(1 - T2_k) \right\} S_{kj}$$

where $p_j = \frac{\sum_{k=1}^{87} S_{kj} T2_k}{\sum_{k=1}^{87} S_{kj}}$ and $S_{kj} = 1$ if cluster k is in block j and 0 otherwise.

```
pj.df <- data2 %>%
  select(t2, block, cluster) %>%
  distinct() %>%
  group_by(block) %>%
  summarise(pj = mean(t2))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
data2_weight <- data2 %>%
  left_join(pj.df, by = "block") %>%
  mutate(weights = t2/(pj) + (1-t2)/(1-pj))

library(estimatr)
model_bcls <- lm_robust(as.formula("voted_2019 ~ t2"),
                                data = data2_weight,
                                fixed_effects = block,
                                clusters = cluster,
                                weights = weights,
                                se_type = "stata")
model_bcls
```

```
##       Estimate Std. Error   t value  Pr(>|t|)    CI Lower   CI Upper DF
## t2 -0.02993385 0.02456589 -1.218513 0.2263612 -0.07876922 0.01890152 86
```

```
# Alternatively, you may also use fixest::feols()
library(fixest)
model_bcls2 <- feols(voted_2019 ~ t2 | block,
                 weights = ~ weights,
                 cluster = ~ cluster,
                 data = data2_weight)

tidy(model_bcls2) %>% filter(term == "t2")
```

```
## # A tibble: 1 x 5
##   term  estimate std.error statistic p.value
```

```
##   <chr>    <dbl>     <dbl>     <dbl>   <dbl>
## 1 t2     -0.0299    0.0246     -1.22   0.226
```

(b)

$$\left(\widehat{\tau}, \widehat{\alpha}_1, \widehat{\alpha}_2, \widehat{\alpha}_3, \widehat{\alpha}_4, \widehat{\beta}\right) = \underset{(\tau,\alpha_1,\alpha_2,\alpha_3,\alpha_4,beta)}{\arg\min} \sum_{k=1}^{87} \sum_{i=1}^{m_k} w_{ik} \left(Y_{ik} - \tau T2_k - \sum_{j=1}^{4} \alpha_1 S_{kj} - \widetilde{\mathbf{X}}'_{ik}\beta \right)^2$$

```
data2_centered <- data2_weight
for (i in 6:15) {
  data2_centered[,i] <- data2_centered[,i] - mean(data2_centered[,i])
}

model_bcls_cov <- lm_robust(as.formula("voted_2019 ~ t2 +
                politician_visits + female +
                age + muslim + sc_st +
                primary_edu + income + married +
                length_residence + owns_home"),
                            data = data2_weight,
                            fixed_effects = block,
                            clusters = cluster,
                            weights = weights,
                            se_type = "stata")
tidy(model_bcls_cov) %>% filter(term == "t2")
```

```
##   term    estimate std.error statistic   p.value    conf.low  conf.high df
## 1   t2 -0.03483687 0.0233624 -1.491151 0.1395809 -0.08127978 0.01160605 86
##      outcome
## 1 voted_2019
```

```
# Alternatively, you may also use fixest::feols()
model_bcls_cov2 <- feols(voted_2019 ~ t2 +
                politician_visits + female +
                age + muslim + sc_st +
                primary_edu + income + married +
                length_residence + owns_home | block,
                weights = ~ weights,
                cluster = ~ cluster,
                data = data2_weight)

tidy(model_bcls_cov2) %>% filter(term == "t2")
```

```
## # A tibble: 1 x 5
##   term  estimate std.error statistic p.value
##   <chr>    <dbl>     <dbl>     <dbl>   <dbl>
## 1 t2     -0.0348    0.0234     -1.49   0.140
```

(c)

$$\left(\widehat{\tau}, \widehat{\alpha}_1, \widehat{\alpha}_2, \widehat{\alpha}_3, \widehat{\alpha}_4, \widehat{\beta}, \widehat{\gamma}\right) = \underset{(\tau,\alpha_1,\alpha_2,\alpha_3,\alpha_4,beta,\gamma)}{\arg\min} \sum_{k=1}^{87} \sum_{i=1}^{m_k} w_{ik} \left(Y_{ik} - \tau T2_k - \sum_{j=1}^{4} \alpha_1 S_{kj} - \widetilde{\mathbf{X}}'_{ik}\beta - T2_k \widetilde{\mathbf{X}}'_{ik}\gamma \right)^2$$

```
model_bcls_full <- lm_robust(as.formula("voted_2019 ~ t2 +
                 politician_visits + female +
                 age + muslim + sc_st +
                 primary_edu + income + married +
                 length_residence + owns_home +
                 t2*(politician_visits + female +
                 age + muslim + sc_st +
                 primary_edu + income + married +
                 length_residence + owns_home)"),
                              data = data2_weight,
                              fixed_effects = block,
                              clusters = cluster,
                              weights = weights,
                              se_type = "stata")
tidy(model_bcls_full) %>% filter(term == "t2")
```

```
##   term    estimate std.error statistic   p.value   conf.low conf.high df
## 1   t2 -0.05558463 0.1126659 -0.493358 0.6230171 -0.2795571 0.1683878 86
##      outcome
## 1 voted_2019
```

```
# Alternatively, you may also use fixest::feols()
model_bcls_full2 <- feols(voted_2019 ~ t2 +
                 politician_visits + female +
                 age + muslim + sc_st +
                 primary_edu + income + married +
                 length_residence + owns_home +
                 t2*(politician_visits + female +
                 age + muslim + sc_st +
                 primary_edu + income + married +
                 length_residence + owns_home) | block,
                 weights = ~ weights,
                 cluster = ~ cluster,
                 data = data2_weight)

tidy(model_bcls_full2) %>% filter(term == "t2")
```

```
## # A tibble: 1 x 5
##   term  estimate std.error statistic p.value
##   <chr>    <dbl>     <dbl>     <dbl>   <dbl>
## 1 t2     -0.0556     0.113    -0.493   0.623
```

References

Gaikwad, N. and Nellis, G. (2021). Overcoming the political exclusion of migrants: Theory and experimental evidence from india. *American Political Science Review*, page 1–18.