Formative Appendix

Zehao Qian

2024-03-06

Stata code explanation and R Code Implementaion for "Messages Designed to Increase Perceived Electoral Closeness Increase Turnout"

```
# ------------------------------------------------------------
## clear the environment var area
# rm(list = ls())
## clear all plots
# graphics.off()
## clear the console area
#cat("\014")
```

Read the data

- read DTA file
- Transfer it from Stata data file (*.dta) to csv format

```
# library(haven)
# data = read_dta(
#   "./data/APRCloseElections_Final_Publication_Replication_Dataset.dta"
# )
```

```
# write.csv(data, file = "./data/data.csv")
```

```
data = read.csv('./data/data.csv')
```

This Stata code is part of a replication file for a study on the effects of perceived electoral closeness on voter turnout. The code performs various statistical analyses and outputs the results. Here's a breakdown of the code, part by part:

Appendix Table A1: Balance Tests

1. Multinomial logistic regression (`mlogit`) is performed to test the balance of covariates across different treatment groups. The `robust` option is used for robust standard errors, and `baseoutcome(1)` sets the reference category for the dependent variable.
2. Local macros (`local`) are used to store statistics like p-values (`mlogitp`), degrees of freedom (`mlogitdf`), and chi-squared values (`mlogitchi`) from the model.

3. A note (`tablenotes2`) is prepared, summarizing the balance test results, and displayed using the `display` command.
4. `putexcel` commands are used to write the balance test results to an Excel file, creating a table with variable names, treatment groups, means, standard deviations, and the note prepared earlier.

```r
library(nnet)
mlogit_model <-
  multinom(
    a_phone_treat_relplacebo_passed ~ d_yearssincereg + d_yearssincereg_miss
    + d_electiondayage + d_gender_male + d_gender_unknown + d_race_black
    + d_race_latino + d_race_miss + d_race_other + d_genvotes + d_primvotes
    + d_specvotes,
    data = data,
    weights = data$weight_allstatestreats
  )
```

```
## # weights:  56 (39 variable)
## initial  value 215843.257546
## iter  10 value 215830.151671
## iter  20 value 215807.949291
## iter  30 value 215800.365621
## iter  40 value 215795.213077
## final  value 215794.860735
## converged
```

```r
summary(mlogit_model)
```

```
## Call:
## multinom(formula = a_phone_treat_relplacebo_passed ~ d_yearssincereg +
##     d_yearssincereg_miss + d_electiondayage + d_gender_male +
##     d_gender_unknown + d_race_black + d_race_latino + d_race_miss +
##     d_race_other + d_genvotes + d_primvotes + d_specvotes, data = data,
##     weights = data$weight_allstatestreats)
##
## Coefficients:
##     (Intercept) d_yearssincereg d_yearssincereg_miss d_electiondayage
## 1  -0.032964640    0.0007858479           0.08750893    -0.0008311004
## 10  0.000211515   -0.0007359491           0.10922055    -0.0003283361
## 11 -0.011546051   -0.0002820043           0.06450216    -0.0004212829
##    d_gender_male d_gender_unknown d_race_black d_race_latino d_race_miss
## 1     0.05736842       0.40488331  0.065524077    0.05439013   0.3158005
## 10    0.01014688       0.26755984  0.044889226   -0.14959959   0.2213804
## 11    0.03562879       0.09353999  0.009041932    0.09651350   0.5975183
##    d_race_other  d_genvotes   d_primvotes d_specvotes
## 1   -0.02982739 0.013687649 -0.0024667696 0.007241021
## 10  -0.12768551 0.006773196 -0.0002958713 0.008427966
## 11  -0.07600287 0.006020268  0.0022778769 0.007495242
##
## Std. Errors:
##    (Intercept) d_yearssincereg d_yearssincereg_miss d_electiondayage
## 1   0.03378481    0.0006259320           0.03460970     0.0005038792
## 10  0.03375436    0.0006285391           0.03442111     0.0005031458
## 11  0.03376136    0.0006276340           0.03475349     0.0005033578
```

```
##    d_gender_male d_gender_unknown d_race_black d_race_latino d_race_miss
## 1     0.01468116        0.1196274   0.03637734    0.06419893   0.1803780
## 10    0.01471265        0.1228912   0.03652982    0.06742543   0.1842329
## 11    0.01469106        0.1277490   0.03686184    0.06346600   0.1715563
##    d_race_other  d_genvotes d_primvotes d_specvotes
## 1    0.05888878 0.007074513 0.005794728 0.007762424
## 10   0.06025916 0.007072251 0.005796647 0.007761972
## 11   0.05961205 0.007070964 0.005791490 0.007758913
##
## Residual Deviance: 431589.7
## AIC: 431667.7
```

```r
library(dplyr)
```

```
##
##    'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
# Define the treatment groups and variables
treatments <-
  c("t_placebo", "t_info_only", "t_closeness_1", "t_closeness_2")
variables <-
  c(
    "d_yearssincereg",
    "d_yearssincereg_miss",
    "d_electiondayage",
    "d_gender_male",
    "d_gender_unknown",
    "d_race_black",
    "d_race_latino",
    "d_race_miss",
    "d_race_other",
    "d_genvotes",
    "d_primvotes",
    "d_specvotes"
  )

# Initialize an empty data frame for storing the summary statistics
summary_df <-
  data.frame(
    Variable = character(),
    Treatment = character(),
    Mean = numeric(),
    SD = numeric(),
    stringsAsFactors = FALSE
```

```
  )

# Loop through variables and treatments
for (var in variables) {
  for (treat in treatments) {
    treat_data <- data %>% filter(!!sym(treat) == 1)
    mean_val <-
      weighted.mean(treat_data[[var]], treat_data$weight_allstatestreats,
                    na.rm = TRUE)
    sd_val <- sd(treat_data[[var]], na.rm = TRUE)

    # Add the summary statistics to the data frame
    summary_df <-
      rbind(
        summary_df,
        data.frame(
          Variable = var,
          Treatment = treat,
          Mean = mean_val,
          SD = sd_val
        )
      )
  }
}

# Add observations count for each treatment
obs_counts <-
  sapply(treatments, function(treat)
    sum(data[[treat]] == 1, na.rm = TRUE))
obs_df <-
  data.frame(
    Variable = "Observations",
    Treatment = treatments,
    Mean = obs_counts,
    SD = NA
  )

# Combine summary statistics and observations count
final_df <- rbind(summary_df, obs_df)
TableA1_BalanceTests = final_df

# Save the data frame to a CSV file
write.csv(TableA1_BalanceTests,
          "./result/TableA1-BalanceTests.csv",
          row.names = FALSE)
```

```
# clear variables
remove(obs_df,
       summary_df,
       mean_val,
       obs_counts,
       sd_val,
       treat,
```

```
      treatments,
      var,
      variables,
      final_df)
```

Table 2: Differences in Election and Turnout Context Across States

1. `sum` commands calculate the turnout rates for placebo subjects in various states. Local macros store these proportions.
2. Another `putexcel` set of commands creates an Excel table summarizing these turnout rates, along with information about the primary date, number of congressional districts, and the number of contested and uncontested primaries.

```
# Calculate turnout rates for placebo subjects by state
turnout_rates <- data %>%
  filter(t_placebo == 1) %>%
  group_by(vf_state) %>%
  summarise(Turnout_Rate = mean(voted_2014_primary, na.rm = TRUE) * 100) %>%
  filter(vf_state %in% c("MA", "MI", "MN", "MO", "NH", "TN", "WI"))

# Define the state context information
state_context <- data.frame(
  State = c("Massachusetts", "Michigan", "Minnesota",
            "Missouri", "New Hampshire", "Tennessee", "Wisconsin"),
  Primary_Date = c("September 9", "August 5", "August 12",
                   "August 5", "September 9", "August 7", "August 12"),
  Number_of_Congressional_Districts = c(9, 14, 8, 8, 2, 9, 8),
  Democratic_Contested = c(2, 5, 1, 4, 0, 3, 3),
  Republican_Contested = c(1, 8, 2, 6, 2, 8, 5),
  Democratic_Uncontested = c(7, 9, 2, 3, 0, 5, 0),
  Republican_Uncontested = c(2, 6, 1, 1, 0, 1, 0)
)

# Merge turnout rates with the state context information
final_table <- cbind(state_context, turnout_rates)

# Rename and reorder columns to match the desired output
final_table <- final_table %>%
  select(State, Turnout_Rate,Primary_Date,
         Number_of_Congressional_Districts,
         Democratic_Contested, Republican_Contested,
         Democratic_Uncontested,
         Republican_Uncontested)
Table2_StateContext = final_table
# Save the table as a CSV file
write.csv(Table2_StateContext, "./result/Table2-StateContext.csv",
          row.names = FALSE)
```

```
remove(turnout_rates, state_context, final_table)
```

Tables 3, A3, A5, and A7: Various Regressions and Proportions

- Variables `close350not2500` and `ageunder50` are generated to represent treatment conditions and age groups, respectively.
- The dataset is structured for panel data analysis using `xtset strata`.
- A series of regression analyses are conducted to assess the impact of closeness treatments on voter turnout, interactions with age, and other covariates. Results are outputted to Excel files.
- The `include` command suggests that another Stata script (`Closeness_SubProgramPRTestRegression.do`) is called multiple times to run specific regression models.

```
data_T3 = data %>%
  mutate(
    close350not2500 = case_when(t_closeness_2 == 1 ~ 0,
                                t_closeness_1 == 1 ~ 1,
                                TRUE ~ NA_real_),
    ageunder50 = as.numeric(d_electiondayage < 50)
  )
```

```
calculate_proportions <- function(subset_data) {
  # Ensure 'subset_data' is filtered based on our model
  # restrictions before passing to this function
  prop_test <-
    prop.test(
      x = sum(subset_data$voted_2014_primary[subset_data$close350not2500 == 1],
              na.rm = TRUE),
      n = sum(subset_data$close350not2500 == 1,
              na.rm = TRUE),
      p = mean(subset_data$voted_2014_primary[subset_data$close350not2500 == 0],
               na.rm = TRUE)
    )  # Setting correct=FALSE for a continuity correction
  return(prop_test)
}
# Regression Function
# This function will perform both linear and logistic regression
# based on the type parameter.
perform_regression <- function(subset_data, formula, type = "linear") {
  if (type == "linear") {
    # Linear regression
    model <- lm(formula, data = subset_data)
    tidy_model <- broom::tidy(model)
  } else if (type == "logistic") {
    # Logistic regression
    model <- glm(formula, data = subset_data, family = binomial())
    tidy_model <- broom::tidy(model)
  }
  return(tidy_model)
  # print(tidy_model)
}
```

```
# Analysis for the Entire Sample
# No filter applied for the entire sample
entire_sample <- data_T3
```

```r
# Proportion test
prop_test_entire <-
  prop.test(
    x = sum(entire_sample$voted_2014_primary[entire_sample$close350not2500 == 1],
            na.rm = TRUE),
    n = sum(entire_sample$close350not2500 == 1,
            na.rm = TRUE),
    p = mean(entire_sample$voted_2014_primary[entire_sample$close350not2500 == 0],
             na.rm = TRUE)
  )

# Linear regression
lm_entire <-
  lm(voted_2014_primary ~ close350not2500, data = entire_sample)

# Logistic regression
logit_entire <-
  glm(voted_2014_primary ~ close350not2500,
      data = entire_sample,
      family = binomial())

# Output the results (to console for simplicity, but you can write them
# to files or tables as needed)
summary(prop_test_entire)
```

```
##             Length Class  Mode
## statistic   1      -none- numeric
## parameter   1      -none- numeric
## p.value     1      -none- numeric
## estimate    1      -none- numeric
## null.value  1      -none- numeric
## conf.int    2      -none- numeric
## alternative 1      -none- character
## method      1      -none- character
## data.name   1      -none- character
```

```r
summary(lm_entire)
```

```
##
## Call:
## lm(formula = voted_2014_primary ~ close350not2500, data = entire_sample)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.2601 -0.2601 -0.2446  0.7399  0.7554
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)     0.244585   0.004738  51.617   <2e-16 ***
## close350not2500 0.015560   0.006691   2.325   0.0201 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 0.4343 on 16853 degrees of freedom
##   (    22078    )
## Multiple R-squared:  0.0003208,  Adjusted R-squared:  0.0002615
## F-statistic: 5.408 on 1 and 16853 DF,  p-value: 0.02006
```

```r
summary(logit_entire)
```

```
##
## Call:
## glm(formula = voted_2014_primary ~ close350not2500, family = binomial(),
##     data = entire_sample)
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)     -1.12771    0.02538 -44.432   <2e-16 ***
## close350not2500  0.08249    0.03548   2.325   0.0201 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 19044  on 16854  degrees of freedom
## Residual deviance: 19039  on 16853  degrees of freedom
##   (    22078    )
## AIC: 19043
##
## Number of Fisher Scoring iterations: 4
```

```r
remove(lm_entire,
       prop_test_entire,
       logit_entire,
       entire_sample)
```

```r
# Filter for Massachusetts
ma_sample <- data_T3 %>% filter(vf_state == "MA")

# Proportion test for Massachusetts
prop_test_ma <-
  prop.test(
    x = sum(ma_sample$voted_2014_primary[ma_sample$close350not2500 == 1],
            na.rm = TRUE),
    n = sum(ma_sample$close350not2500 == 1, na.rm = TRUE),
    p = mean(ma_sample$voted_2014_primary[ma_sample$close350not2500 == 0],
             na.rm = TRUE)
  )

# Linear regression for Massachusetts
lm_ma <- lm(voted_2014_primary ~ close350not2500, data = ma_sample)

# Logistic regression for Massachusetts
logit_ma <-
  glm(voted_2014_primary ~ close350not2500,
```

```r
      data = ma_sample,
      family = binomial())

# Output the results for Massachusetts
summary(prop_test_ma)
```

```
##             Length Class  Mode
## statistic   1      -none- numeric
## parameter   1      -none- numeric
## p.value     1      -none- numeric
## estimate    1      -none- numeric
## null.value  1      -none- numeric
## conf.int    2      -none- numeric
## alternative 1      -none- character
## method      1      -none- character
## data.name   1      -none- character
```

```r
summary(lm_ma)
```

```
##
## Call:
## lm(formula = voted_2014_primary ~ close350not2500, data = ma_sample)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.2719 -0.2719 -0.2507  0.7281  0.7492
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)      0.25075    0.01202  20.859   <2e-16 ***
## close350not2500  0.02116    0.01700   1.245    0.213
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4394 on 2669 degrees of freedom
##   (   3274    )
## Multiple R-squared:  0.00058,    Adjusted R-squared:  0.0002055
## F-statistic: 1.549 on 1 and 2669 DF,  p-value: 0.2134
```

```r
summary(logit_ma)
```

```
##
## Call:
## glm(formula = voted_2014_primary ~ close350not2500, family = binomial(),
##     data = ma_sample)
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)     -1.09462    0.06312 -17.342   <2e-16 ***
## close350not2500  0.10967    0.08813   1.244    0.213
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 3068.7  on 2670  degrees of freedom
## Residual deviance: 3067.1  on 2669  degrees of freedom
##   (   3274    )
## AIC: 3071.1
##
## Number of Fisher Scoring iterations: 4
```

```r
remove(lm_ma,
       prop_test_ma,
       logit_ma,
       ma_sample)
```

Although I try to define the function, I cannot get the same results of Proportion and SE as the stata's.

Table 3. Turnout by Closeness Experimental Condition in Phone Field Experiment.

	(1)	(2)	(3)	(4)	(5)
Sample	Proportion Voting 350 Votes Treatment	Proportion Voting 2500 Votes Treatment	Difference of Proportions (350 votes - 2500 Votes) [Standard error]	Regression Estimate of Difference (350 Votes - 2500 Votes) [Standard Error]	Number of Observations (350 Votes, 2500 Votes)
Entire sample	.260	.245	.016 [.007]	.012 [.005]	(8453, 8402)
State = Massachusetts	.272	.251	.021 [.017]	.013 [.014]	(1335, 1336)
State = Michigan	.233	.222	.012 [.019]	−.005 [.014]	(1050, 993)
State = Minnesota	.178	.165	.013 [.011]	.010 [.010]	(2295, 2372)
State = Missouri	.376	.343	.033 [.028]	.033 [.024]	(585, 568)
State = New Hampshire	.321	.306	.015 [.030]	.034 [.024]	(474, 484)
State = Tennessee	.429	.414	.015 [.028]	.011 [.022]	(653, 636)
State = Wisconsin	.257	.250	.007 [.014]	.010 [.011]	(2061, 2013)
No competitive house primary	.250	.235	.016 [.008]	.012 [.007]	(5867, 5881)
Either house primary competitive	.282	.268	.015 [.012]	.010 [.010]	(2586, 2521)
Ever voters (Have voted before)	.267	.252	.015 [.007]	.011 [.006]	(8168, 8106)
Have voted in primary	.524	.515	.009 [.013]	.013 [.011]	(3164, 3034)
Have voted, but never in primary	.105	.095	.010 [.006]	.010 [.006]	(5004, 5072)
No prior history of voting	.056	.044	.012 [.018]	.033 [.017]	(285, 296)

Table A2. Analysis of Intent to Vote in Election

```r
library(tidyr)
# Filter out rows with NA in either intend_to_vote_maybeYes or close350not2500
filtered_data <- data_T3 %>%
  filter(!is.na(intend_to_vote_maybeYes) & !is.na(close350not2500))

# Now perform the proportion test with the filtered data
prop_test_result <-
  prop.test(x = table(
    filtered_data$intend_to_vote_maybeYes,
    filtered_data$close350not2500
  ))

# Print the result
print(prop_test_result)
```

```
##
##  2-sample test for equality of proportions with continuity correction
##
## data:  table(filtered_data$intend_to_vote_maybeYes, filtered_data$close350not2500)
## X-squared = 3.666, df = 1, p-value = 0.05553
## alternative hypothesis: two.sided
## 95 percent confidence interval:
##  -0.0003500128  0.0303199351
## sample estimates:
##    prop 1    prop 2
## 0.5069162 0.4919312
```

```r
remove(filtered_data,
       prop_test_result)
```

```r
# install.packages("survey")
# install.packages("lmtest")
library(survey)
```

```
## Warning:   'survey' R 4.3.3

##      grid

##      Matrix

##
##    'Matrix'

## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack

##      survival

##
##    'survey'
```

```
## The following object is masked from 'package:graphics':
##
##     dotchart
```

```
library(sandwich)

# Create a survey design object to handle weights
design <- svydesign(ids = ~1, data = data_T3, weights = ~weight_allstatestreats)

# Fit a linear regression model using the survey design
model <- svyglm(intend_to_vote_maybeYes ~ close350not2500 + strata, design = design)

# Obtain robust standard errors using the vcovHC function from the sandwich package
robust_se <- sqrt(diag(vcovHC(model, type = "HC1")))

# You can print the coefficients and their robust standard errors like this:
coefs <- coef(model)
se <- robust_se

# Print the coefficients and robust standard errors
print(coefs)
```

```
##     (Intercept) close350not2500          strata
##     0.512296941     0.012433450     0.001245765
```

```
print(se)
```

```
##     (Intercept) close350not2500          strata
##     0.002422479     0.002184156     0.000050626
```

```
remove(design,
       model,
       robust_se,
       coefs,
       se)
```

Figure 1 and Table A4: Comparative Effectiveness of Different Treatments

- Regression analyses compare the effectiveness of different treatments on voter turnout.
- `lincom` commands are used to compare the treatment effects.
- Variables for plotting (`tvar`, `beta`, `beta_lowci`, `beta_hici`) are prepared, and a figure is generated using the `twoway` command, showing the estimated treatment effects with confidence intervals.
- The figure is exported as a PDF, and unnecessary variables are dropped.

```
library(survey)
library(lmtest)
```

```
## Warning:   'lmtest' R 4.3.3
```

```
##      zoo
```

```
##
##     'zoo'

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

```r
library(sandwich)
library(ggplot2)

design <-
  svydesign(
    ids = ~ 1,
    data = data,
    weights = ~ weight_allstatestreats
  )

# Fit the model
model <-
  svyglm(voted_2014_primary ~ t_info_only + t_closeness_1
         + t_closeness_2 + strata,
         design = design)



# Obtain and print summary statistics
summary(model)
```

```
##
## Call:
## svyglm(formula = voted_2014_primary ~ t_info_only + t_closeness_1 +
##     t_closeness_2 + strata, design = design)
##
## Survey design:
## svydesign(ids = ~1, data = data, weights = ~weight_allstatestreats)
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.1761223  0.0056162  31.360  < 2e-16 ***
## t_info_only   0.0222691  0.0059142   3.765 0.000167 ***
## t_closeness_1 0.0317539  0.0063258   5.020  5.2e-07 ***
## t_closeness_2 0.0206772  0.0063115   3.276 0.001053 **
## strata        0.0014185  0.0001088  13.040  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.1844696)
##
## Number of Fisher Scoring iterations: 2
```

```r
# Calculate robust standard errors
robust_se <- sqrt(diag(vcovHC(model, type = "HC1")))
```

```r
# Extract coefficients and confidence intervals
coefs <- coef(model)
cis <- confint(model)

# Create a data frame for plotting
plot_data <- data.frame(
  Treatment = c(
    "Election Reminder",
    "Closeness (350 Votes)",
    "Closeness (2500 Votes)"
  ),
  Effect = c(coefs["t_info_only"],
             coefs["t_closeness_1"],
             coefs["t_closeness_2"]),
  LowerCI = c(cis["t_info_only", 1],
              cis["t_closeness_1", 1],
              cis["t_closeness_2", 1]),
  UpperCI = c(cis["t_info_only", 2],
              cis["t_closeness_1", 2],
              cis["t_closeness_2", 2])
)
```

```r
ggplot(plot_data, aes(x = Treatment, y = Effect)) +
  geom_col() +
  geom_errorbar(aes(ymin = LowerCI, ymax = UpperCI), width = 0.2) +
  ylab("Estimated Treatment Effect (Relative to Placebo)") +
  xlab("") +
  theme_minimal()
```

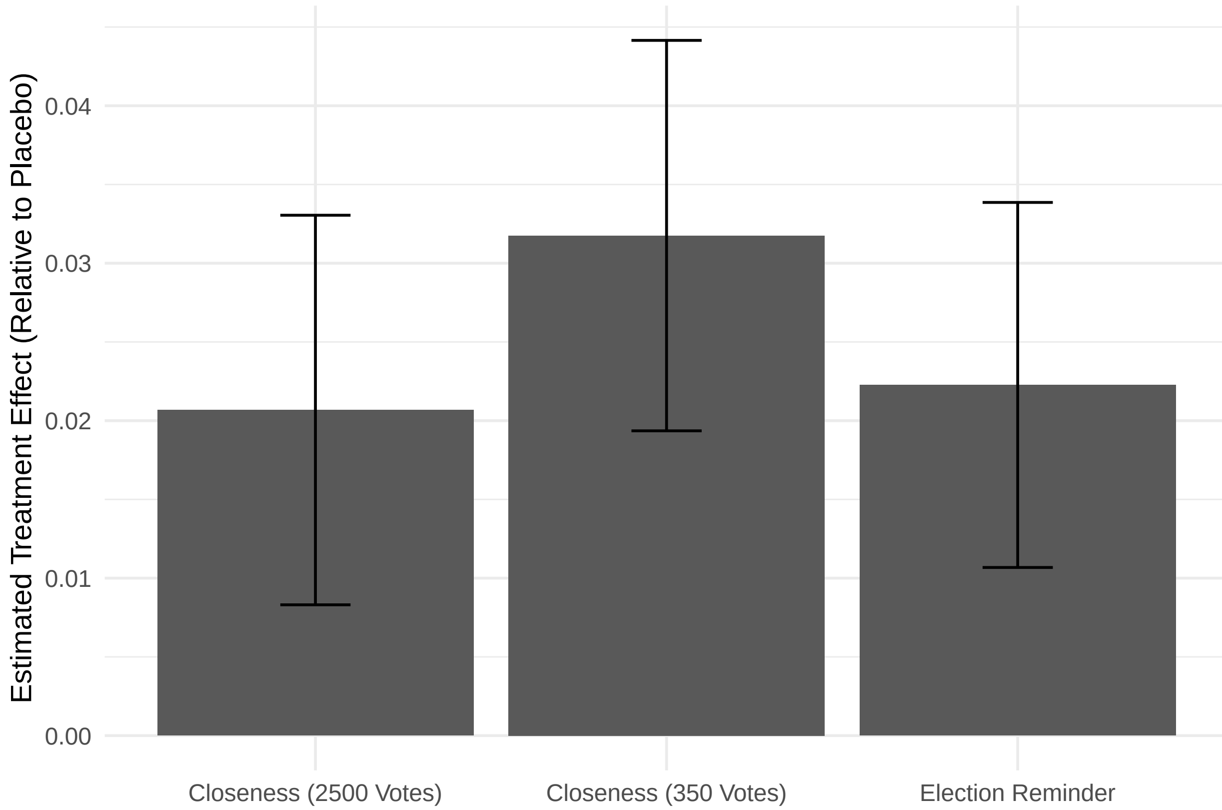

Table A6: Proportion Voting by Experimental Conditions, State, and Strata

- Proportions of voting by experimental condition, state, and voter strata are calculated and outputted to an Excel file.
- The `foreach` and `forvalues` loops iterate over treatment conditions, states, and voter history categories to summarize the data.

```
library(dplyr)

# Calculate for the entire sample by treatment
results_entire <- data %>%
  group_by(treatment = coalesce(t_placebo, t_info_only,
                               t_closeness_1, t_closeness_2)) %>%
  summarize(
    Proportion_Voting = mean(voted_2014_primary, na.rm = TRUE),
    N = n(),
    .groups = 'drop'
  ) %>%
  mutate(Group = "Entire Sample")

# Calculate for each state as an example
results_state <- data %>%
  filter(vf_state == "MA") %>%  # Replace "MA" with other states as needed
  group_by(treatment = coalesce(t_placebo, t_info_only,
                               t_closeness_1, t_closeness_2)) %>%
```

```r
  summarize(
    Proportion_Voting = mean(voted_2014_primary, na.rm = TRUE),
    N = n(),
    .groups = 'drop'
  ) %>%
  mutate(Group = "State=Massachusetts")

# Combine results
combined_results <- bind_rows(results_entire, results_state)

# Add more states and strata as needed
```

```r
write.csv(combined_results,
          "./result/table6-ProportionVotingByCondition.csv",
          row.names = FALSE)
```

Table A8: Relationship Between Intention to Vote and Actual Turnout

- A regression model explores the relationship between the intention to vote and actual turnout, considering the experimental condition and other covariates.
- Results are outputted to an Excel file.

```r
library(survey)
library(lmtest)
library(sandwich)

# Define the survey design object
design <-
  svydesign(
    ids = ~ 1,
    data = data_T3,
    weights = ~ weight_allstatestreats
  )

# Fit the regression model with interaction terms
model <-
  svyglm(voted_2014_primary ~ intend_to_vote_maybeYes * close350not2500
         + strata,
         design = design)

# Calculate robust standard errors
robust_se <- sqrt(diag(vcovHC(model, type = "HC1")))
```

```r
# Extract model coefficients
coefs <- coef(model)

# Combine coefficients and robust SEs
results <- cbind(Estimate = coefs, `Robust SE` = robust_se)

# Convert to data frame for output
results_df <- as.data.frame(results)
```

```r
# Define the note to add to the CSV
# Note: OLS regression coefficients with robust standard errors in brackets.
# Dependent variable is voted in 2014 primary election (Yes = 1, No = 0).
# Model includes state x voter history x district competitiveness
# fixed effects. Weighted analysis. ***p<0.01; **p<0.05; *p<0.1.

# Write the results and note to a CSV file
write.csv(results_df, "./result/TableA8-VoteIntent_Turnout.csv", row.names = TRUE)
```