

jh8186_solutionsHW2

Jerry Huang

2024-06-24

Library loaded

```
library(readr)
```

Problem 1 - Changing Minds on Gay Marriage?

Question 1

```
gay <- read_csv("gay.csv", show_col_types = FALSE)
baseline <- subset(gay, study == 1 & wave == 1 &
  treatment %in% c("Same-Sex Marriage Script by Gay Canvasser",
    "Same-Sex Marriage Script by Straight Canvasser",
    "No Contact"))
baseline_1 <- baseline$ssm[baseline$treatment == "No Contact"]
baseline_2 <- baseline$ssm[baseline$treatment == "Same-Sex Marriage Script by Gay Canvasser"]
baseline_3 <- baseline$ssm[baseline$treatment == "Same-Sex Marriage Script by Straight Canvasser"]

anova_result <- aov(ssm ~ treatment, data = baseline)
summary(anova_result)
```

```
##           Df Sum Sq Mean Sq F value Pr(>F)
## treatment    2      4    1.752   0.622  0.537
## Residuals 7419 20905    2.818
```

Why to check randomization? I check it to make sure the samples are all randomly selected. How to know if randomization was performed? I performed an Anova test if there are statistically significant differences in the baseline SSM scores among the three treatment groups. The result has shown a p-value of 0.54 which is higher than the $\alpha = 0.05$.

Question 2

```
wave2 <- subset(gay, study == 1 & wave == 2 &
  treatment %in% c("Same-Sex Marriage Script by Gay Canvasser",
    "Same-Sex Marriage Script by Straight Canvasser",
    "No Contact"))

No_contact <- wave2$ssm[wave2$treatment == "No Contact"]
Gay_Canvasser <- wave2$ssm[wave2$treatment == "Same-Sex Marriage Script by Gay Canvasser"]
Straight_Canvasser <- wave2$ssm[wave2$treatment == "Same-Sex Marriage Script by Straight Canvasser"]

mean_nc <- mean(No_contact)
```

```
mean_gay <- mean(Gay_Canvasser)
mean_straight <- mean(Straight_Canvasser)

ATE_gay <- mean_gay - mean_nc
ATE_straight <- mean_straight - mean_nc
ATE_gay
```

```
## [1] 0.09987463
```

```
ATE_straight
```

```
## [1] 0.122248
```

The average treatment effect of canvassing by gay canvassers is an increase of 0.0999 points in support for same-sex marriage compared to no contact. The average treatment effect of canvassing by straight canvassers is an increase of 0.122 points in support for same-sex marriage compared to no contact.

Question 3

```
recycle <- subset(gay, study == 1 & wave == 2 &
                  treatment %in% c("Recycling Script by Gay Canvasser",
                                   "Recycling Script by Straight Canvasser"))
mean_recycle_gay <- mean(recycle$ssm[recycle$treatment == "Recycling Script by Gay Canvasser"])
mean_recycle_straight <- mean(recycle$ssm[recycle$treatment == "Recycling Script by Straight Canvasser"])

ATE_gay_script <- mean_recycle_gay - mean_gay
ATE_straight__script <- mean_recycle_straight - mean_straight

ATE_gay_script
```

```
## [1] -0.03204239
```

```
ATE_straight__script
```

```
## [1] -0.1575845
```

I think the purpose of this treatment is to see whether the observed effects on support for same-sex marriage are specifically due to the content of the script (same-sex marriage script) or simply due to the act of canvassing by individuals (regardless of the script content). The results have shown a decrease in support for same-sex marriage when the canvassing message was about recycling.

Question 4

```
# Sub-setting all waves for study 1
nc <- subset(gay, subset = treatment=="No Contact" & study==1)
gc <- subset(gay, subset = treatment=="Same-Sex Marriage Script by Gay Canvasser" & study==1)
sc <- subset(gay, subset = treatment=="Same-Sex Marriage Script by Straight Canvasser" & study==1)

mean_nc <- tapply(nc$ssm, nc$wave, mean)
mean_gc <- tapply(gc$ssm, gc$wave, mean)
mean_sc <- tapply(sc$ssm, sc$wave, mean)

ATE_gay <- mean_gc - mean_nc
ATE_straight <- mean_sc - mean_nc
```

```
ATE_gay
```

```
##           1           2           3           4           5           6
## -0.01756893  0.09987463  0.08136612  0.09262577  0.14789543  0.08643548
##           7
##  0.05936835
```

```
ATE_straight
```

```
##           1           2           3           4           5           6
##  0.05694517  0.12224797  0.06040800  0.08719659  0.09855523  0.06351524
##           7
## -0.04253721
```

Wave 1 is the pre-test result of ssm. For gay group, we can see that the effect is largest in wave 5 and remains positive up to wave 7, indicating that the impact of canvassing by gay canvassers on support for same-sex marriage is lasting and significant over time. For Straight Canvassers: while there is an initial positive effect (Waves 2 to 6), the impact diminishes over time and becomes negative by Wave 7, indicating that the effect is not sustained in the long term.

Question 5

```
baseline_st2 <- subset(gay, subset = study == 2 & wave == 1 &
                        treatment %in% c("Same-Sex Marriage Script by Gay Canvasser",
                                          "No Contact"))
baseline_st2_nc <- baseline_st2$ssm[baseline_st2$treatment == "No Contact"]
baseline_st2_gc <- baseline_st2$ssm[baseline_st2$treatment == "Same-Sex Marriage Script by Gay Canvasser"]

# ATE
ATE_gay_s2 <- mean(baseline_st2_gc) - mean(baseline_st2_nc)
ATE_gay_s2
```

```
## [1] 0.001653782
```

The ATE is very close to zero, suggesting that the mean baseline support for same-sex marriage is nearly identical between the treatment group and the control group. This is because we expect both groups to be randomized before the experiment begins.

Question 6:

```
wave2_st2 <- subset(gay, subset = study == 2 & wave == 2 &
                    treatment %in% c("Same-Sex Marriage Script by Gay Canvasser",
                                      "No Contact"))
wave2_st2_nc <- wave2_st2$ssm[wave2_st2$treatment == "No Contact"]
wave2_st2_gc <- wave2_st2$ssm[wave2_st2$treatment == "Same-Sex Marriage Script by Gay Canvasser"]

# ATE
ATE_gay_s2 <- mean(wave2_st2_gc) - mean(wave2_st2_nc)
ATE_gay_s2
```

```
## [1] 0.1238937
```

Yes, the results of Study 2 are consistent with those of Study 1 because both results are positive. The positive ATE suggests that there is a causal effect of the treatment. Specifically, being canvassed by a gay canvasser

with a same-sex marriage script increases support for same-sex marriage compared to not being contacted at all.

Question 7

```
nc_st2 <- subset(gay, subset = treatment=="No Contact" & study==2)
gc_st2 <- subset(gay, subset = treatment=="Same-Sex Marriage Script by Gay Canvasser" & study==2)

mean_nc_st2 <- tapply(nc_st2$ssm, nc_st2$wave, mean)
mean_gc_st2 <- tapply(gc_st2$ssm, gc_st2$wave, mean)

ATE_gay_st2 <- mean_gc_st2 - mean_nc_st2
ATE_gay_st2
```

```
##           1           2           3           4           7
## 0.001653782 0.123893673 0.150420627 0.125173822 0.307062649
```

From this result, we can see the effect remains positive from wave 1 to wave 7, indicating a lasting and long term effects.

Problem 2 - Election Fraud in Russia

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
##
## The following objects are masked from 'package:stats':
##
##   filter, lag
##
## The following objects are masked from 'package:base':
##
##   intersect, setdiff, setequal, union
```

Question 1

```
load("fraud.RData")

# Calculating the vote share for each row
russia2011$vote_share <- russia2011$votes / russia2011$turnout

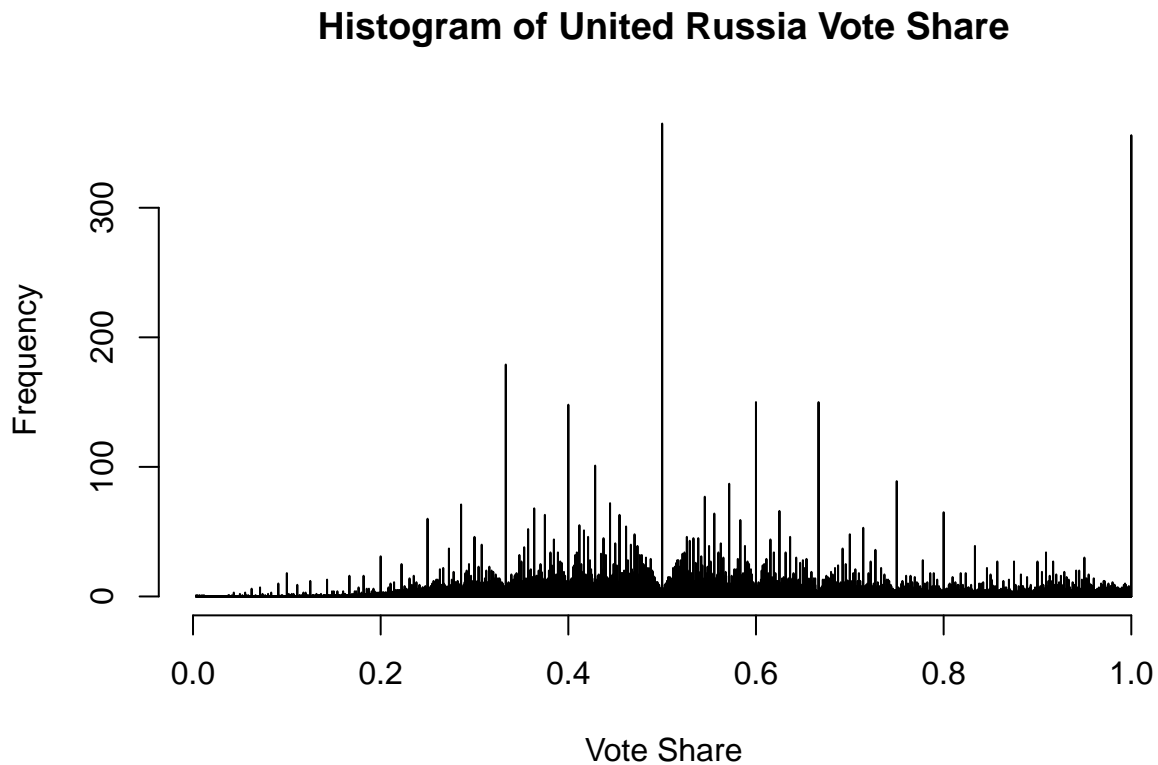
# Filtering vote share proportion in terms of frequency
number_vote_share <- russia2011 %>%
  group_by(vote_share) %>%
  summarise(freq = n()) %>%
  arrange(desc(freq))

head(number_vote_share, 10)

## # A tibble: 10 x 2
##   vote_share freq
##   <dbl> <int>
```

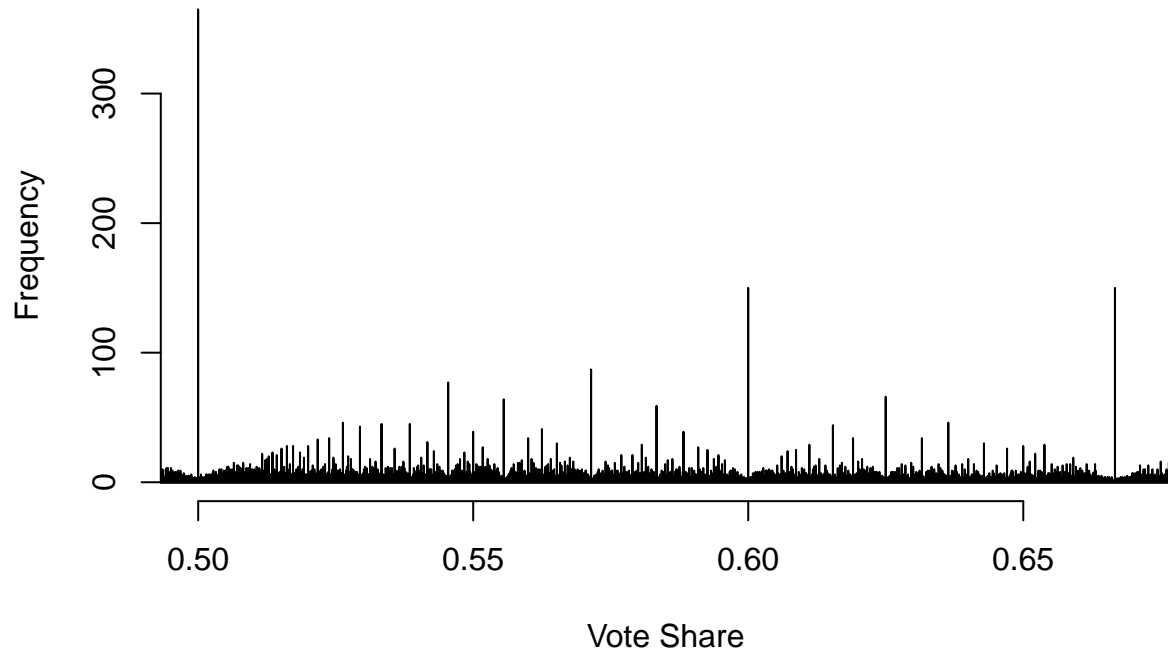
```
## 1      0.5      365
## 2      1       356
## 3     0.333    179
## 4      0.6     150
## 5     0.667    150
## 6      0.4     148
## 7     0.429    101
## 8      0.75     89
## 9     0.571     87
## 10     0.545     77
```

```
# Breaks = length(unique(vote_share)) sets the breaks to the number of unique values in vote share, so
hist(russia2011$vote_share, breaks = length(unique(russia2011$vote_share)), main = "Histogram of United Russia Vote Share")
```



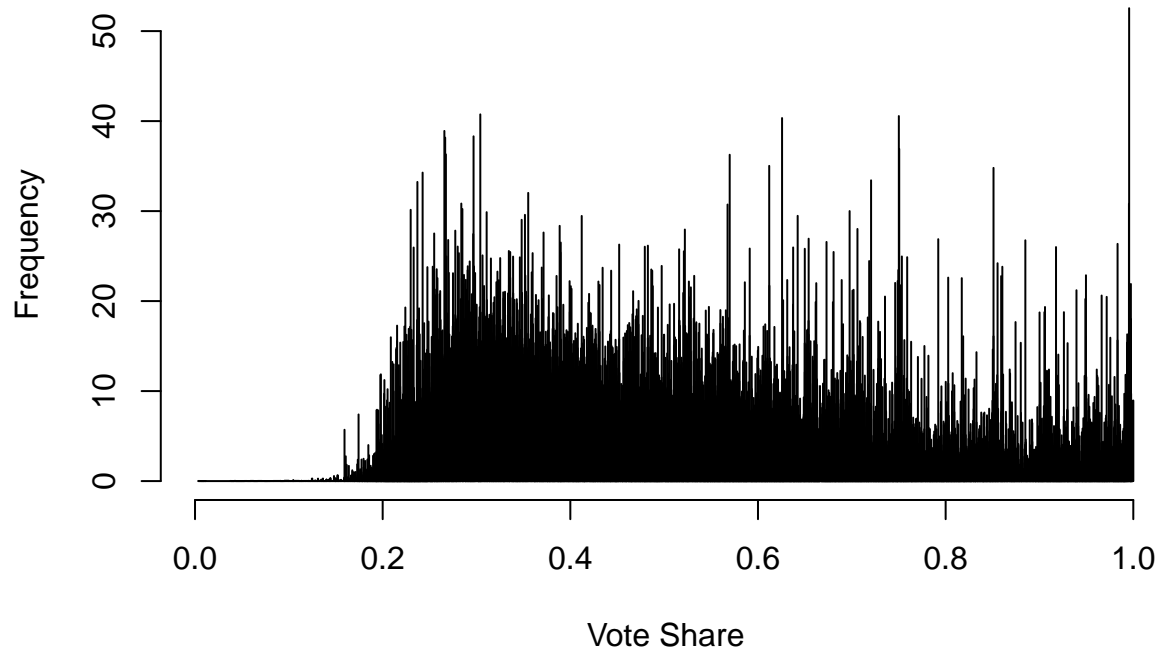
```
hist(russia2011$vote_share, breaks = length(unique(russia2011$vote_share)), main = "Histogram of United Russia Vote Share")
```

Histogram of United Russia Vote Share



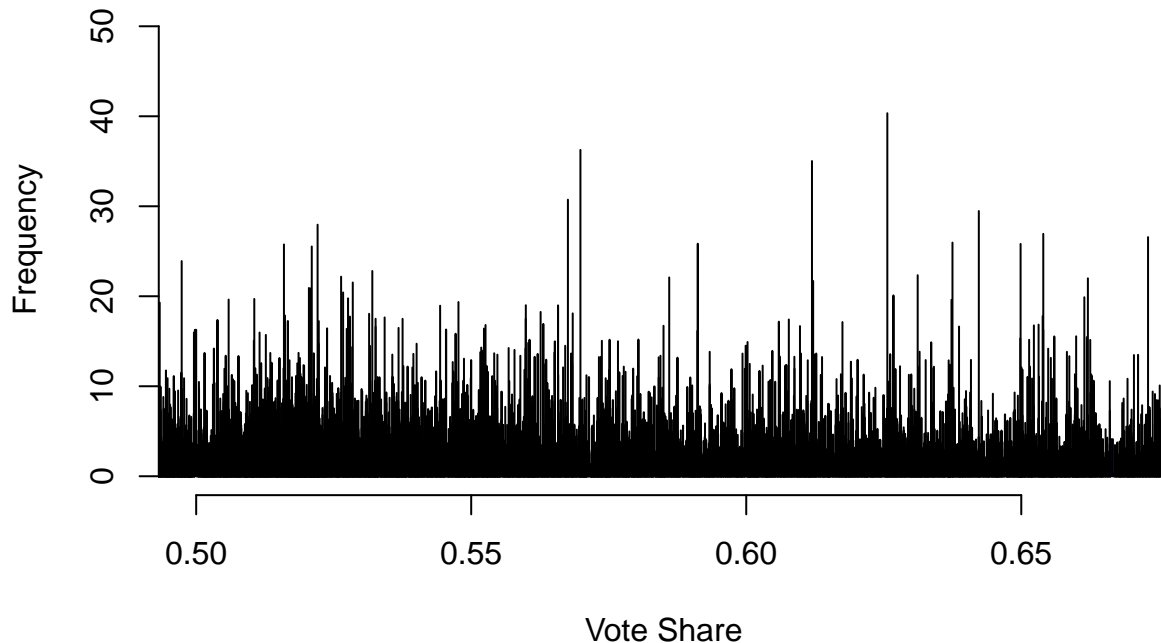
```
hist(russia2011$vote_share, breaks = unique(russia2011$vote_share), main = "Histogram of United Russia
```

Histogram of United Russia Vote Share



```
hist(russia2011$vote_share, breaks = unique(russia2011$vote_share), main = "Histogram of United Russia
```

Histogram of United Russia Vote Share



`= length(unique(vote_share))` sets the breaks to the number of unique values in vote share, so that I can visualize better how many times a certain fraction has happened. From the graphs we can see that the frequencies of fractions with low numerators and denominators such as $1/2$ and $2/3$ are high. From the general histogram we can see that the distribution of vote share is not even.

Question 2

```
# Number of simulations
num_simulations <- 1000

# Initialize a list to store the simulated vote shares
simulated_vote_shares <- list()

# Conduct Monte Carlo simulations
set.seed(123) # For reproducibility
for (i in 1:num_simulations) {
  # Simulate vote shares using the binomial distribution
  simulated_shares <- rbinom(n = nrow(russia2011),
                             size = russia2011$turnout,
                             prob = russia2011$vote_share) / russia2011$turnout
  simulated_vote_shares[[i]] <- simulated_shares
}

# Combine the results into a single vector
simulated_vote_shares <- unlist(simulated_vote_shares)

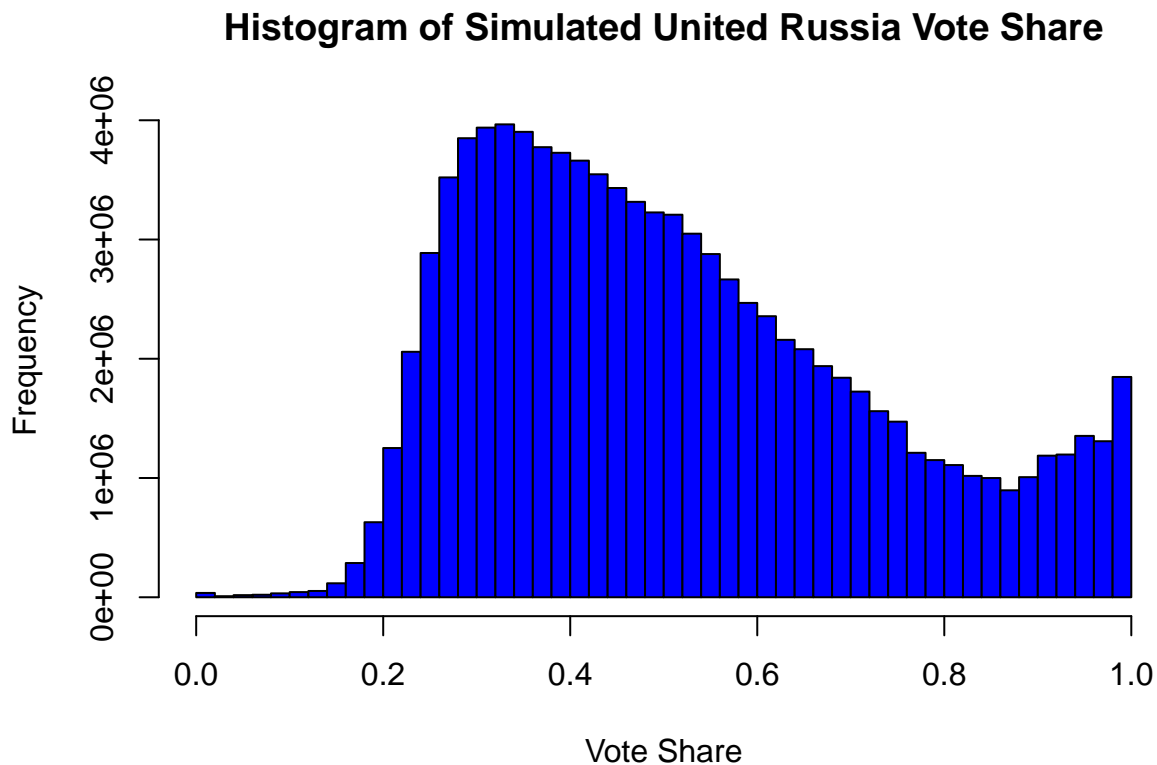
# Calculate the frequency of each vote share
simulated_vote_shares_rounded <- round(simulated_vote_shares * 100) / 100
vote_share_freq <- table(simulated_vote_shares_rounded)
vote_share_freq_df <- as.data.frame(vote_share_freq)
```

```
colnames(vote_share_freq_df) <- c("vote_share", "frequency")

# Identify the 10 most frequent vote share values
top_10_vote_shares <- vote_share_freq_df[order(-vote_share_freq_df$frequency), ][1:10, ]
top_10_vote_shares

##      vote_share frequency
## 34         0.33  1985604
## 35         0.34  1980161
## 33         0.32  1977594
## 36         0.35  1963608
## 32         0.31  1960899
## 31          0.3   1944775
## 37         0.36  1939456
## 39         0.38  1914419
## 30         0.29  1904868
## 40         0.39  1868714

# Plot the histogram of the simulated vote share values
hist(simulated_vote_shares_rounded, breaks = 50,
     main = "Histogram of Simulated United Russia Vote Share",
     xlab = "Vote Share", ylab = "Frequency", col = "blue", border = "black")
```



The distribution from the Monte Carlo simulation provides a baseline for what the vote share distribution should look like under normal conditions (assuming the binomial distribution model is correct).

Question 3

```
set.seed(123)
```



```

# Define the fractions of interest
fractions <- c(1/2, 1/3, 3/5, 2/3)

# Calculate observed proportions
observed_proportions <- sapply(fractions, function(f) {
  mean(russia2011$vote_share == f)
})

# Calculate simulated proportions
simulated_proportions <- sapply(fractions, function(f) {
  mean(simulated_vote_shares == f)
})

# Combine into a data frame for comparison
comparison_df <- data.frame(
  Fraction = fractions,
  Observed = observed_proportions,
  Simulated = simulated_proportions
)

print(comparison_df)

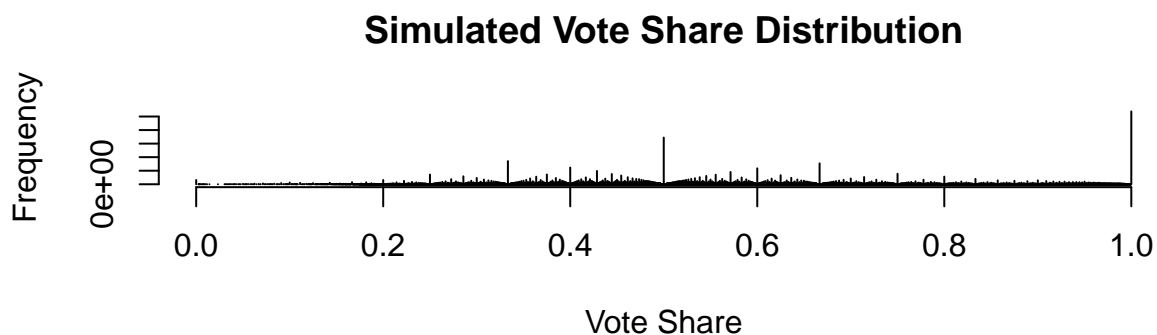
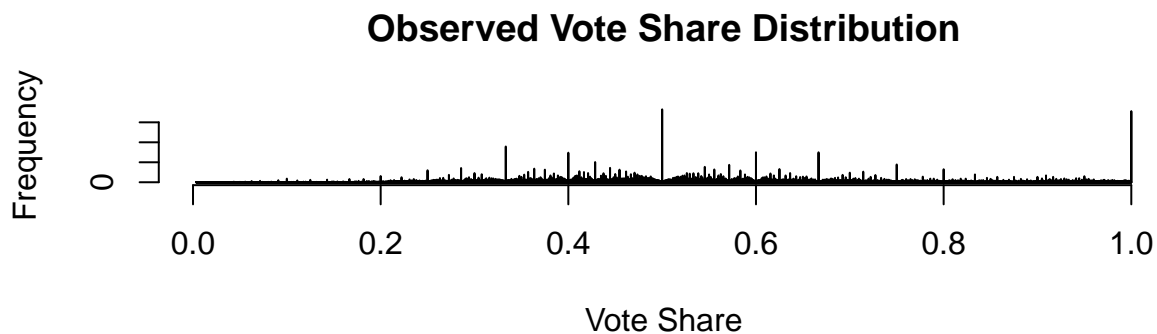
##      Fraction      Observed      Simulated
## 1 0.5000000 0.003842307 0.003627338
## 2 0.3333333 0.001884310 0.001797452
## 3 0.6000000 0.001579030 0.001237539
## 4 0.6666667 0.001579030 0.001627644

# Plot histograms for observed data
par(mfrow = c(2, 1)) # Set up a 2x1 plotting area

# Histogram of observed data
hist(russia2011$vote_share, breaks = length(unique(russia2011$vote_share)),
     main = "Observed Vote Share Distribution",
     xlab = "Vote Share",
     col = "blue",
     border = "black")

# Plot histograms for simulated data
hist(simulated_vote_shares, breaks = length(unique(simulated_vote_shares)),
     main = "Simulated Vote Share Distribution",
     xlab = "Vote Share",
     col = "green",
     border = "black")

```



```
# Reset plotting area
par(mfrow = c(1, 1))
```

We can see from the observed value and simulated value that they are quite similar, and the distributions of them look very similar but simulated distribution is shown to be less extreme as what we observed.

Problem 3

```
library(readxl)
```

Part a

```
gotv <- read_xlsx("gotv_individual.xlsx")
gotv$sex <- as.integer(grepl("female", gotv$sex))

gotv$age <- (2006 - gotv$yob)

df <- gotv %>%
  group_by(hh_id) %>%
  summarize(
    treatment = first(treatment),
    hh_size = first(hh_size),
    sex = mean(sex, na.rm = TRUE),
    p2000 = mean(p2000, na.rm = TRUE),
    g2000 = mean(g2000, na.rm = TRUE),
    p2002 = mean(p2002, na.rm = TRUE),
    g2002 = mean(g2002, na.rm = TRUE),
    p2004 = mean(p2004, na.rm = TRUE),
```

```

g2004 = mean(g2004, na.rm = TRUE),
p2004_mean = mean(p2004_mean, na.rm = TRUE),
g2004_mean = mean(g2004_mean, na.rm = TRUE),
voted = mean(voted, na.rm = TRUE),
age = mean(age, na.rm = TRUE)
)

```

- As shown in the code above.
- As shown in the code above.
- I grouped the data by the `hh_id` and took the mean of the numerical columns.
- The authors analyzed household because family members may influence the votes for each other. For instance, if the elders (mom or dad) read the mail, they may influence their children to vote for certain candidates.

Part b

```

mean_samples <- df %>%
  group_by(treatment) %>%
  summarise(
    hh_size = mean(hh_size, na.rm = TRUE),
    sex = mean(sex, na.rm = TRUE),
    p2000 = mean(p2000, na.rm = TRUE),
    g2000 = mean(g2000, na.rm = TRUE),
    p2002 = mean(p2002, na.rm = TRUE),
    g2002 = mean(g2002, na.rm = TRUE),
    p2004 = mean(p2004, na.rm = TRUE),
    age = mean(age, na.rm = TRUE)
  )
mean_samples

## # A tibble: 5 x 9
##   treatment hh_size  sex p2000 g2000 p2002 g2002 p2004  age
##   <chr>      <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 Civic Duty    1.91 0.503 0.266 0.865 0.410 0.836 0.416  51.2
## 2 Control      1.91 0.502 0.265 0.866 0.409 0.834 0.417  51.3
## 3 Hawthorne    1.91 0.503 0.263 0.867 0.412 0.836 0.419  51.2
## 4 Neighbors    1.91 0.505 0.263 0.865 0.406 0.835 0.423  51.3
## 5 Self         1.91 0.501 0.263 0.863 0.410 0.835 0.421  51.2

```

Yes, the means are very similar across all treatment groups. This similarity indicates that the randomization process effectively distributed participants across treatment groups.

Part c

```

ATE_mean_votes <- df %>%
  group_by(treatment) %>%
  summarize(
    voted = mean(voted, na.rm = TRUE) ) %>%
  mutate(ATE = voted - voted[treatment == "Control"])

ATE_mean_votes

## # A tibble: 5 x 3
##   treatment voted  ATE

```

```
##   <chr>      <dbl> <dbl>
## 1 Civic Duty 0.325 0.0206
## 2 Control    0.304 0
## 3 Hawthorne  0.332 0.0275
## 4 Neighbors  0.389 0.0848
## 5 Self       0.357 0.0526
```

We are investigating the treatment on encouraging votes, so the value of ATE is in terms of votes. Two Assumptions: Ignorability is assumed to be satisfied in this experiment because of randomization. It allows us to ensure that treatment assignment is independent of potential outcomes.

Consistency is assumed to be satisfied in this experiment because of SUTVA. It allows us to ensure that each treatment unit in the groups will not interfere each other during the experiment. The potential outcomes of individuals are consistent with what they actually received.

Part d:

```
tau_hat <- ATE_mean_votes$ATE[ATE_mean_votes$treatment == "Neighbors"]

# Variance
var_neighbors <- var(df$voted[df$treatment == "Neighbors"], na.rm = TRUE)
var_control <- var(df$voted[df$treatment == "Control"], na.rm = TRUE)

n_neighbors <- sum(df$treatment == "Neighbors", na.rm = TRUE)
n_controls <- sum(df$treatment == "Control", na.rm = TRUE)

var_tau_hat <- (var_neighbors / n_neighbors) + (var_control / n_controls)
z_stat <- sqrt(n_neighbors + n_controls)*(tau_hat - 0) / sqrt(var_tau_hat)

# Calculate the p-value for a two-tailed test
# pnorm(abs(z_stat)) gives the cumulative probability up to |z_stat|
# 1 - pnorm(abs(z_stat)) gives the right-tail probability
# 2 * (1 - pnorm(abs(z_stat))) gives the two-tailed p-value
p_value <- 2 * (1 - pnorm(abs(z_stat)))
```

$\hat{\tau}$: 0.08478808 $Var[\hat{\tau}]$: 1.156835e-05 Z-score: 8635.506 P-value: 0

Part e:

```
library(ggplot2)

observed_z_stat <- sqrt(n_neighbors + n_controls) * tau_hat / sqrt(var_tau_hat)

# Sharp Null Simulation
num_iterations <- 1000

# Initialize a vector to store the Z-statistics
z_stats <- numeric(num_iterations)

# Perform the randomization test
for (i in 1:num_iterations) {
  # Shuffle the treatment labels
  df_shuffled <- df %>%
    mutate(treatment = sample(treatment))
```

```

# Calculate the means for shuffled data
mean_neighbors_shuffled <- mean(df_shuffled$voted[df_shuffled$treatment == "Neighbors"], na.rm = TRUE)
mean_control_shuffled <- mean(df_shuffled$voted[df_shuffled$treatment == "Control"], na.rm = TRUE)

# Calculate the shuffled ATE
tau_hat_shuffled <- mean_neighbors_shuffled - mean_control_shuffled

# Calculate the variance for shuffled data
var_neighbors_shuffled <- var(df_shuffled$voted[df_shuffled$treatment == "Neighbors"], na.rm = TRUE)
var_control_shuffled <- var(df_shuffled$voted[df_shuffled$treatment == "Control"], na.rm = TRUE)

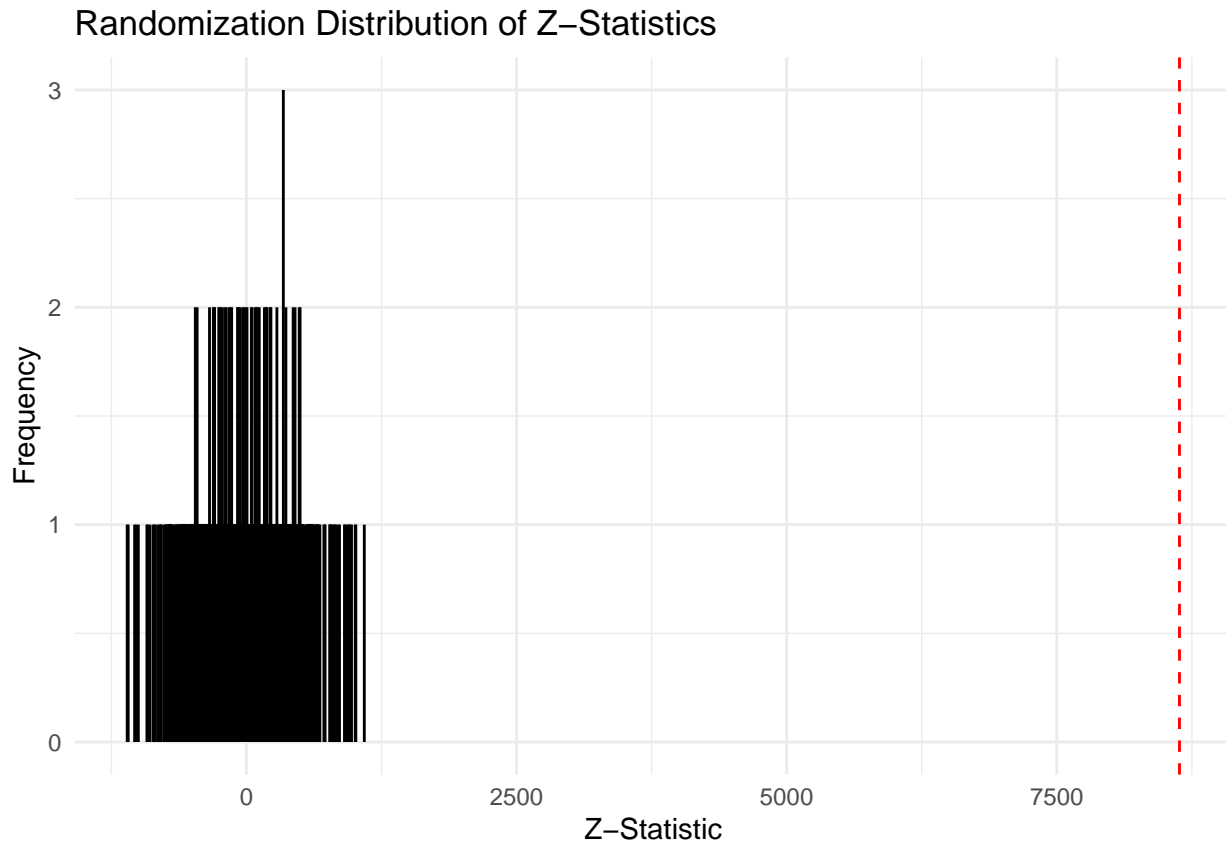
# Calculate the variance of the shuffled ATE
var_tau_hat_shuffled <- (var_neighbors_shuffled / n_neighbors) + (var_control_shuffled / n_controls)

# Calculate the Z-statistic for the shuffled data
z_stat_shuffled <- sqrt(n_neighbors + n_controls) * tau_hat_shuffled / sqrt(var_tau_hat_shuffled)

# Store the Z-statistic
z_stats[i] <- z_stat_shuffled
}

# Create a histogram of the Z-statistics from the simulations
ggplot(data.frame(z_stats), aes(x = z_stats)) +
  geom_histogram(binwidth = 0.1, fill = "blue", alpha = 0.7, color = "black") +
  labs(title = "Randomization Distribution of Z-Statistics",
       x = "Z-Statistic",
       y = "Frequency") +
  geom_vline(xintercept = observed_z_stat, color = "red", linetype = "dashed") +
  theme_minimal()

```



```
# Calculate the two-sided p-value
p_value_randomization <- mean(abs(z_stats) >= abs(observed_z_stat))

observed_z_stat
```

```
## [1] 8635.506
```

```
p_value_randomization
```

```
## [1] 0
```

The two-sided p-value result is 0.

Part f:

The p-values from both questions are 0. This result implies that there is a strong casual effect in neighbors' treatment under both hypothesis testing. The observed test statistic is extremely unlikely under the null hypothesis for both tests, which is $ATE = 0$ and $Y_i(\text{neighbors}) = Y_i(\text{control})$ for all i . We find a strong evidence against null hypothesis.

Problem 4

Part a:

To test if ATE is identified means that it can be expressed in terms of observable quantities.

ATE:

$$\tau = E[Y_i(1) - Y_i(0)]$$

For Observed treated units:

$$E[Y_i|D_i = 1]$$

For Observed Control units:

$$E[Y_i|D_i = 0]$$

1. In the setting of block stratification, we will have the following according to consistency, which states that the potential outcomes of an individual is equal to the actual observed outcome.

$$E[Y_i(1)|G_i = g] = E[Y_i|D_i = 1, G_i = g]$$

$$E[Y_i(0)|G_i = g] = E[Y_i|D_i = 0, G_i = g]$$

2. For ATE in a specific strata $G_i = g$ in the experiment we observed:

$$\tau_g = E[Y_i|D_i = 1, G_i = g] - E[Y_i|D_i = 0, G_i = g]$$

3. If we sum up all strata given their weight respectively:

$$\tau = \sum_{g=1}^G \frac{N_g}{N} \tau_g$$

Substituting the expression for τ_g :

$$\tau = \sum_{g=1}^G \frac{N_g}{N} E[Y_i|D_i = 1, G_i = g] - E[Y_i|D_i = 0, G_i = g]$$

Now we have successfully express ATE in terms of observable quantities. Under assumptions of ignorability, consistency, and positivity, this would be identified in the experiment setting in this case.

Part b:

Assume that $E[\hat{\tau}(g)|G_i = g, N_g = n_g] = \tau(g)$ and that $E[\frac{N_g}{N}] = Pr(G_i = g)$

Initially, we take the expected value of the estimand:

$$E[\hat{\tau}] = E \left[\sum_{g=1}^G \hat{\tau}(g) \frac{N_g}{N} \right]$$

According to the linearity of expectation, we will have

$$E \left[\sum_{g=1}^G \hat{\tau}(g) \frac{N_g}{N} \right] = \sum_{g=1}^G E \left[\hat{\tau}(g) \frac{N_g}{N} \right] = \sum_{g=1}^G E[\hat{\tau}(g)] E \left[\frac{N_g}{N} \right]$$

From the assumption we know that since $E[\hat{\tau}(g)|G_i = g, N_g = n_g] = \tau(g)$, we will have:

$$\sum_{g=1}^G E[\hat{\tau}(g)] E \left[\frac{N_g}{N} \right] = \sum_{g=1}^G \tau(g) Pr(G_i = g)$$

By the definition of ATE, we know that from this experimental setting the ATE is equal to the sum of all ATE from each strata. Therefore, we have:

$$E[\hat{\tau}(g)] = \tau$$

Part c: Assume that $E[\frac{N_g}{N}] = Pr(G_i = g)$

$$E[\hat{\tau}_w] = E \left[\frac{1}{N} \sum_{i=1}^N \left(\frac{D_i Y_i}{w(G_i)} - \frac{(1 - D_i) Y_i}{1 - w(G_i)} \right) \right]$$

According to linearity of expectation, we will have

$$E \left[\frac{1}{N} \sum_{i=1}^N \left(\frac{D_i Y_i}{w(G_i)} - \frac{(1 - D_i) Y_i}{1 - w(G_i)} \right) \right] = \frac{1}{N} \sum_{i=1}^N E \left[\left(\frac{D_i Y_i}{w(G_i)} \right) \right] - E \left[\left(\frac{(1 - D_i) Y_i}{1 - w(G_i)} \right) \right]$$

Next, according to law of total expectation, we will have:

$$\frac{1}{N} \sum_{i=1}^N E \left[\left(\frac{D_i Y_i}{w(G_i)} \right) \right] - E \left[\left(\frac{(1 - D_i) Y_i}{1 - w(G_i)} \right) \right] = \frac{1}{N} \sum_{i=1}^N E \left[E \left[\left(\frac{D_i Y_i}{w(G_i)} \right) | G_i \right] \right] - E \left[E \left[\left(\frac{(1 - D_i) Y_i}{1 - w(G_i)} \right) | G_i \right] \right]$$

Therefore, we will have:

$$\frac{1}{N} \sum_{i=1}^N E \left[\left(\frac{1 * Y_i(1)}{Pr(D_i = 1 | G_i)} \right) Pr(D_i = 1 | G_i) \right] - E \left[\left(\frac{1 * Y_i(0)}{Pr(D_i = 0 | G_i)} \right) Pr(D_i = 0 | G_i) \right]$$

Eventually, we will have after cancellation:

$$\frac{1}{N} \sum_{i=1}^N E[Y_i(1)] - E[Y_i(0)]$$