PSET 4

Sarah Van Alsten

2/11/2020

Question 1

a. The model would produce misleading results if there was interference between units (such as the intended target sharing the funny video with other friends or their social network). If this was the case, then the observed outcomes for the treated group (and likely some of the untreated group) are not independent of potential assignment for the other units, and you have a violation of SUTVA. Second, because invites were also sent to the friends of initially randomized units, not all treatment assignments were random. On average, people are more similiar to their friends and close social networks than to other people, so the units in treatment may also have more similar outcomes than if the experiment was fully randomized (clustering effects). The model would produce especially misleading results among those with high degress of social connectivity (especially for highly interconnected units between members of the treatment group and linking nodes between the treatment/control groups). The non-supporters who get treatment are more likely to be friends with supporters, are more likely to be similar to those supporters, and, thus, more likely to donate than would non-supporters without a supporter friend.

The regression isn't sufficient because it doesn't include the clustering effects and interconnectivity of the observations.

b.

```
#Degree of individuals in network
#rowsum = # of friends each individual has
dat$friends <- rowSums(G)
dat$id <- as.numeric(as.character(dat$id))

#plot distributions of friends by treat in whole network
dat %>%
  ggplot(aes(x = friends)) +
  geom_histogram() +
  facet_grid(.~D) +
  theme_bw() +
  labs(x = "Number of Friends",
       y = "histogram")+
  ggtitle("Distribtutions of Friends Among Full Sample")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
#summarise mean and SD on # friends
dat %>%
  group_by(D) %>%
  summarise(mean = mean(friends, na.rm = T),
            sd = sd(friends, na.rm = T))
```

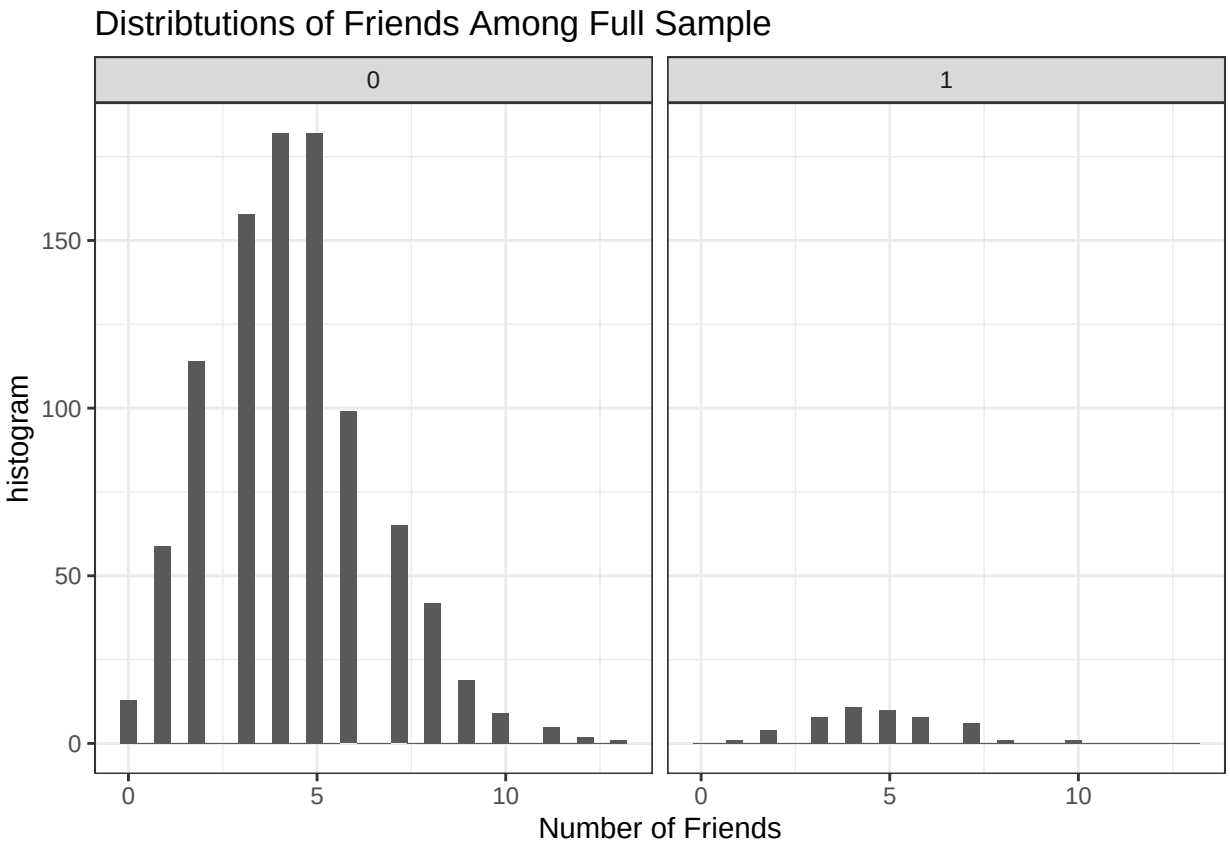


Figure 1: distributions of number of friends

```
## # A tibble: 2 x 3
##       D  mean    sd
##   <dbl> <dbl> <dbl>
## 1     0  4.35  2.13
## 2     1  4.7   1.79
```

```
#assess significance of difference
t.test(dat$friends ~ dat$D)
```

```
##
##  Welch Two Sample t-test
##
## data:  dat$friends by dat$D
## t = -1.3175, df = 56.613, p-value = 0.193
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.8700896  0.1795633
## sample estimates:
## mean in group 0 mean in group 1
##        4.354737        4.700000
```

```
#plot distriubtions of friends by treat in supporters
dat %>%
  filter(supporter == 1) %>%
  ggplot(aes(x = friends)) +
  geom_histogram() +
  facet_grid(.~D) +
  theme_bw() +
  labs(x = "Number of Friends",
       y = "histogram") +
  ggtitle("Distribtutions of Friends Among Supporters")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
#supporter only data
supporter <- dat %>% filter(supporter == 1)

#summarise mean and sd of friends
supporter %>%
  group_by(D) %>%
  summarise(mean = mean(friends, na.rm = T),
            sd = sd(friends, na.rm = T))
```

```
## # A tibble: 2 x 3
##       D  mean    sd
##   <dbl> <dbl> <dbl>
## 1     0  4.4   2.42
## 2     1  4.24  1.69
```

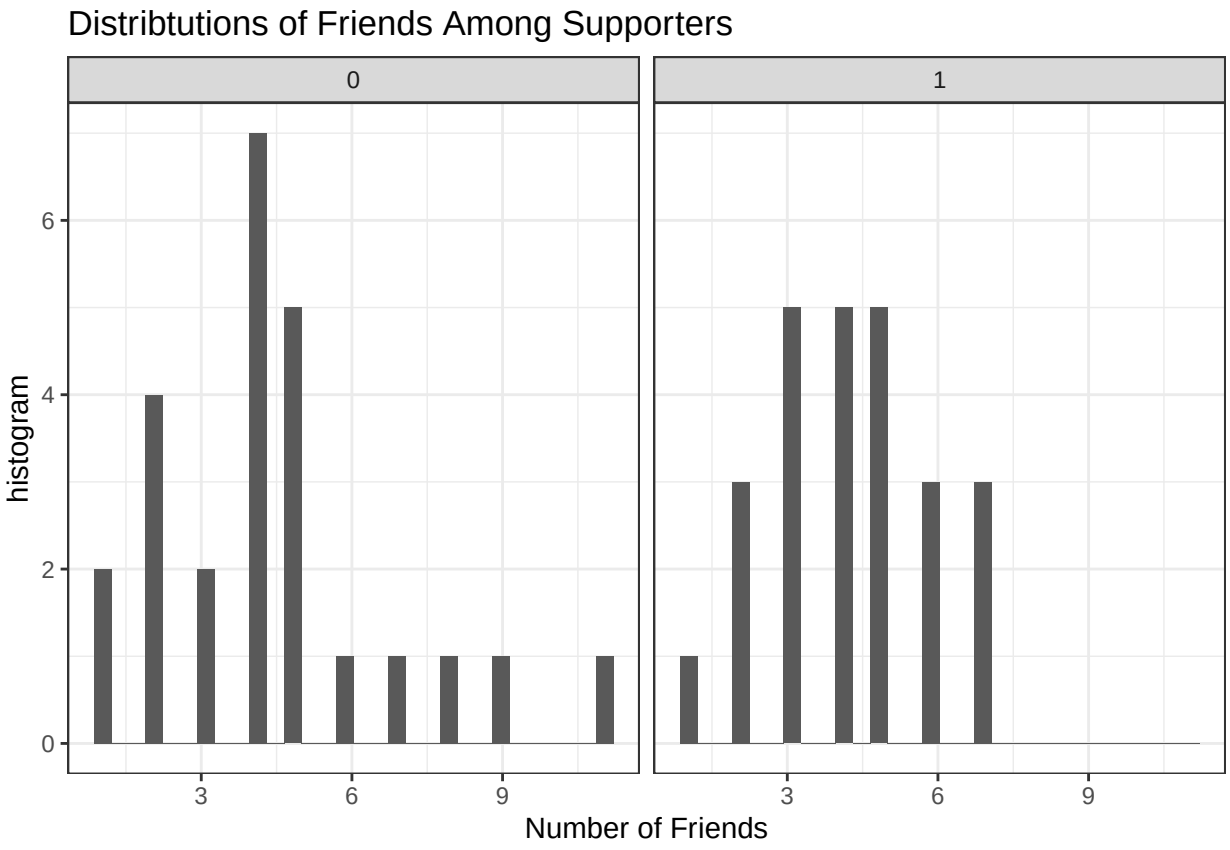


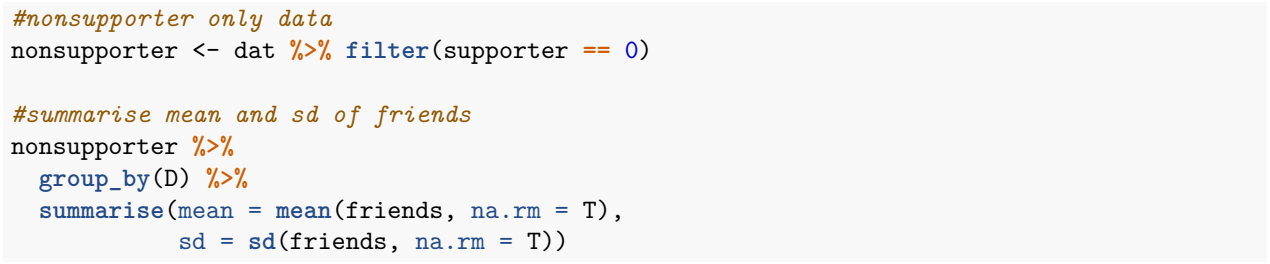
Figure 2: distributions of number of friends

```r
#assess significance of difference
t.test(supporter$friends ~ supporter$D)
```

```
##
##  Welch Two Sample t-test
##
## data:  supporter$friends by supporter$D
## t = 0.27138, df = 42.959, p-value = 0.7874
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -1.029026  1.349026
## sample estimates:
## mean in group 0 mean in group 1
##            4.40            4.24
```

```r
#plot distriubtions of friends by treat in non-supporters
dat %>%
  filter(supporter == 0) %>%
  ggplot(aes(x = friends)) +
  geom_histogram() +
  facet_grid(.~D) +
  theme_bw() +
  labs(x = "Number of Friends",
       y = "histogram")+
  ggtitle("Distribtutions of Friends Among Non-Supporters")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



```r
#nonsupporter only data
nonsupporter <- dat %>% filter(supporter == 0)

#summarise mean and sd of friends
nonsupporter %>%
  group_by(D) %>%
  summarise(mean = mean(friends, na.rm = T),
            sd = sd(friends, na.rm = T))
```

```
## # A tibble: 2 x 3
##       D  mean    sd
##   <dbl> <dbl> <dbl>
## 1     0  4.35  2.13
## 2     1  5.16  1.80
```

```r
#assess signficance of difference
t.test(nonsupporter$friends ~ nonsupporter$D)
```

```
##
##  Welch Two Sample t-test
##
## data:  nonsupporter$friends by nonsupporter$D
```

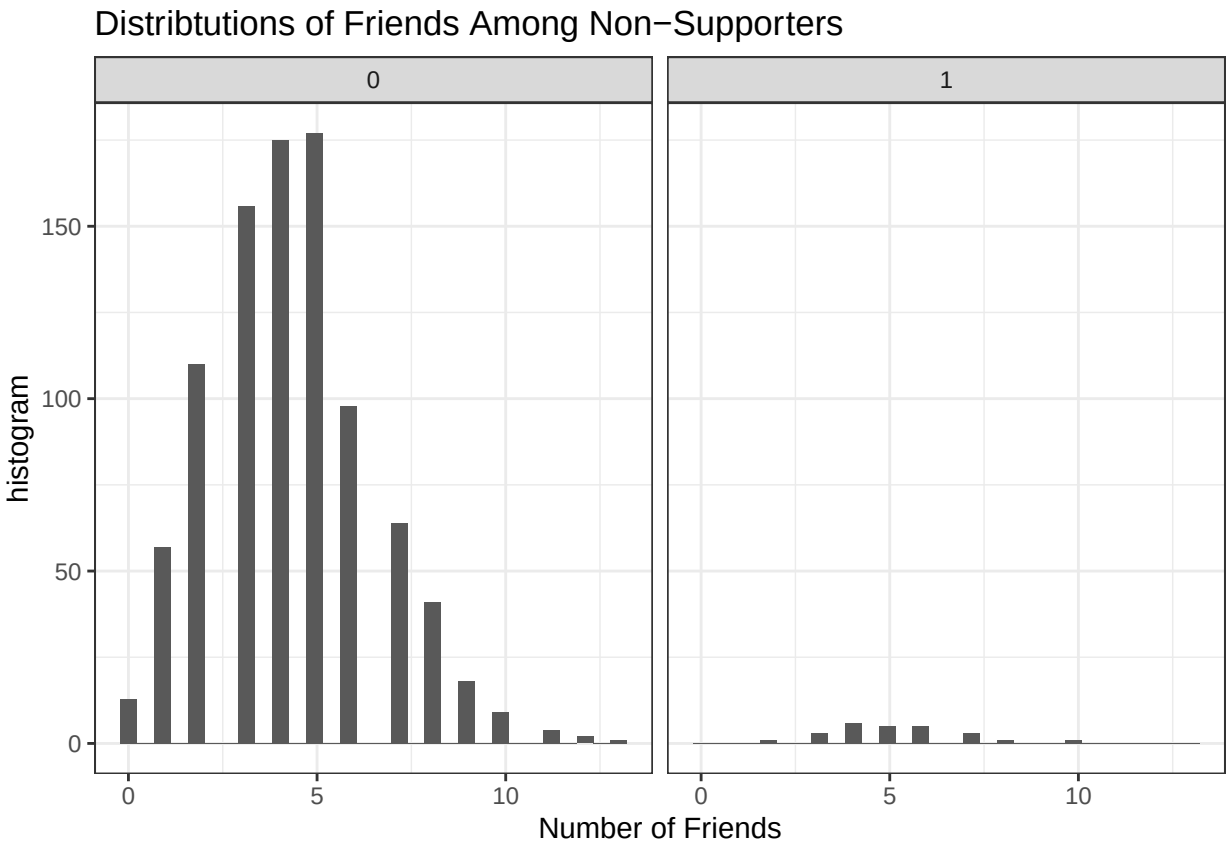


Figure 3: distributions of number of friends

```
## t = -2.2046, df = 25.855, p-value = 0.03659
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -1.5586494 -0.0543236
## sample estimates:
## mean in group 0 mean in group 1
##        4.353514        5.160000
```

Although the mean number of friends did not significantly differ between treatment and control groups in the network as a whole or among supporters (both $p > 0.05$), non-supporters who recieved the treatment had more friends ($M = 5.16$, $SD = 1.80$) in the network than did non-supporters ($M = 4.35$, $SD = 2.13$) who recieved control ($t(28.86) = -2.20$, $p < 0.05$). Based on the distributions, this difference was driven largely by a few observations in the non-supporter treatment group with a high number of friends (~10). Altogether, this suggests that among non-supporters, selection into treatment was not fully random and varied according to the number of friends an individual had; presumably due to connections with a supporter also receiving treatment.

c.

```
#function to rerandomize treatment:
#1/2 supporters get treatmet + 1 non-supporter friend of each supporter
reRandomize <- function(){
  Gcopy <- G
  sampleVec <- as.numeric(dat[dat$supporter == 1, "id"])

  #randomly assign treatment to 50% of supporters
  support.treat <- sample(x = sampleVec, size = 25, replace = F)


  #now randomly pick a nonsupporting friend from those who were treated
  #picked IDs of treated group:
  treat.id <- support.treat

  #empty vector for friend ids
  friend.id <- c(rep(99999, 25))

  #loop over matrix, selecting treated rows and sampling their friends
  for (i in 1:25){
    #select row for treated supporter : make it a matrix rowwise
    friend.row <- t(as.matrix(Gcopy[treat.id[i],]))

    #if they only have one friend, that friend gets treatment
    #else, sample from all possible friends
    if (rowSums(friend.row) == 1){
      friend.id[i] <- which(friend.row == 1)

    } else {
      friend.id[i] <- sample(x = unique(which(friend.row == 1)), size = 1, replace = F)

    }
  }

  #concatenate all the treated ids
```

```
  all.treat <- append(treat.id, friend.id)


  #new vector for if the selected individual got treatment
  dstar <- ifelse(dat$id %in% all.treat, 1, 0)

  #return treatment assignment
  return(dstar)
}
```

```
#setseed
set.seed(02139)

#make a new treatment assignment
dstar <- reRandomize()
```

i.

```
#Assess balance on number of friends between simulated treatment group + actual in...

#i. the whole sample
dat <- dat %>%
  arrange(desc(supporter)) %>%
  mutate(rowid = row_number()) %>%
  mutate(simTreat = dstar)


#mean and sd in simulated treat
dat %>%
  filter(simTreat == 1) %>%
  summarise(mean_simulate = mean(friends, na.rm = T),
            sd_simulate = sd(friends, na.rm = T))
```

```
##   mean_simulate sd_simulate
## 1          4.34    2.291154
```

```
#mean and sd in real
dat %>%
  filter(D == 1) %>%
  summarise(mean_actual = mean(friends, na.rm = T),
            sd_actual = sd(friends, na.rm = T))
```

```
##   mean_actual sd_actual
## 1         4.7  1.787142
```

```
#plots
p1 <-
  dat %>%
  filter(simTreat == 1) %>%
  ggplot(aes(x = friends)) +
  geom_histogram() +
```

```
  labs(x = "Friends (Simulated)")+
  theme_bw()

p2 <-
  dat %>%
  filter(D == 1) %>%
  ggplot(aes(x = friends)) +
  geom_histogram() +
  labs(x = "Friends (Actual)") +
  theme_bw()

#put them together
ggpubr::ggarrange(p1, p2)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

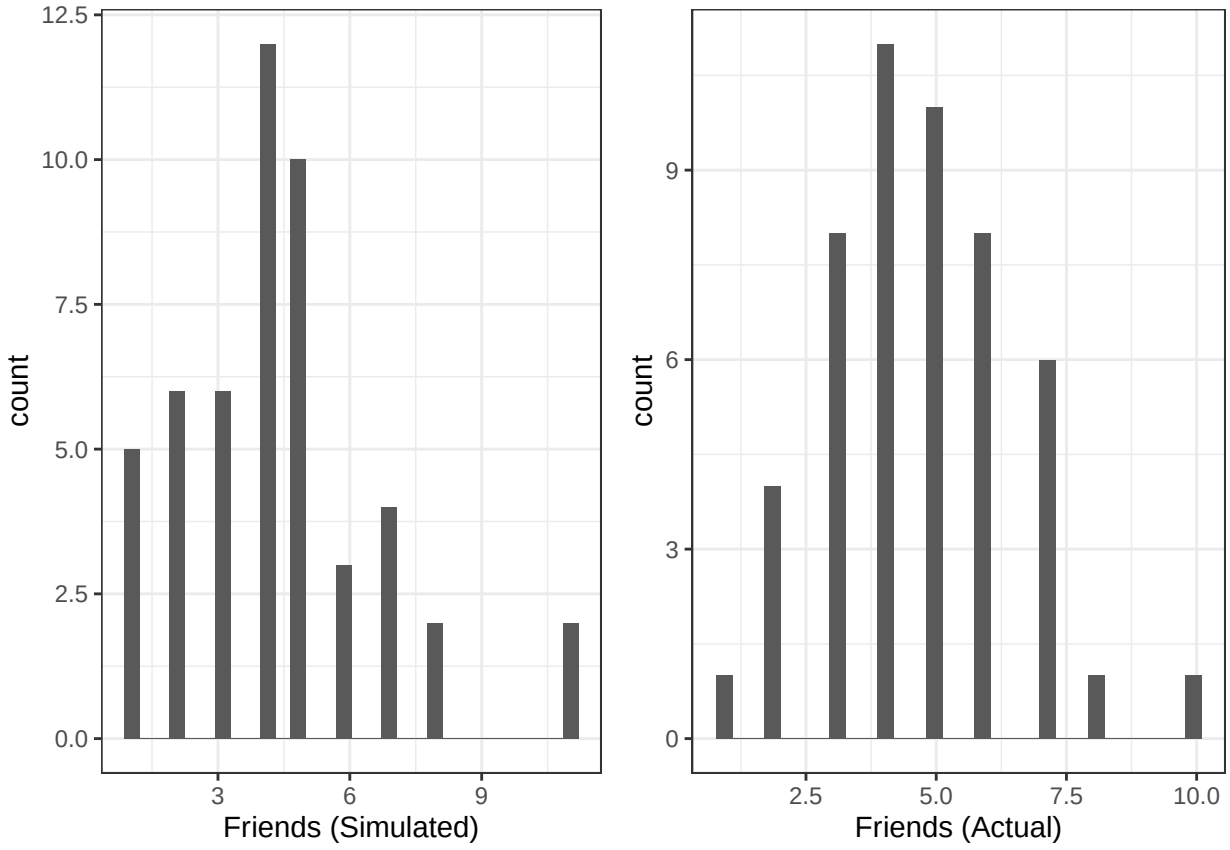


Figure 4: Friends in simulated vs actual, full sample

```
#test difference
t.test(dat[dat$simTreat == 1, "friends"], dat[dat$D == 1, "friends"])
```

```
##
```

```
##  Welch Two Sample t-test
##
## data:  dat[dat$simTreat == 1, "friends"] and dat[dat$D == 1, "friends"]
## t = -0.87606, df = 92.517, p-value = 0.3833
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -1.1760862  0.4560862
## sample estimates:
## mean of x mean of y
##      4.34      4.70
```

ii.

```
#ii. balance among supporters

#mean and sd in simulated treat
dat %>%
  filter(supporter == 1) %>%
  filter(simTreat == 1) %>%
  summarise(mean_simulate = mean(friends, na.rm = T),
            sd_simulate = sd(friends, na.rm = T))
```

```
##   mean_simulate sd_simulate
## 1          6.25    3.304038
```

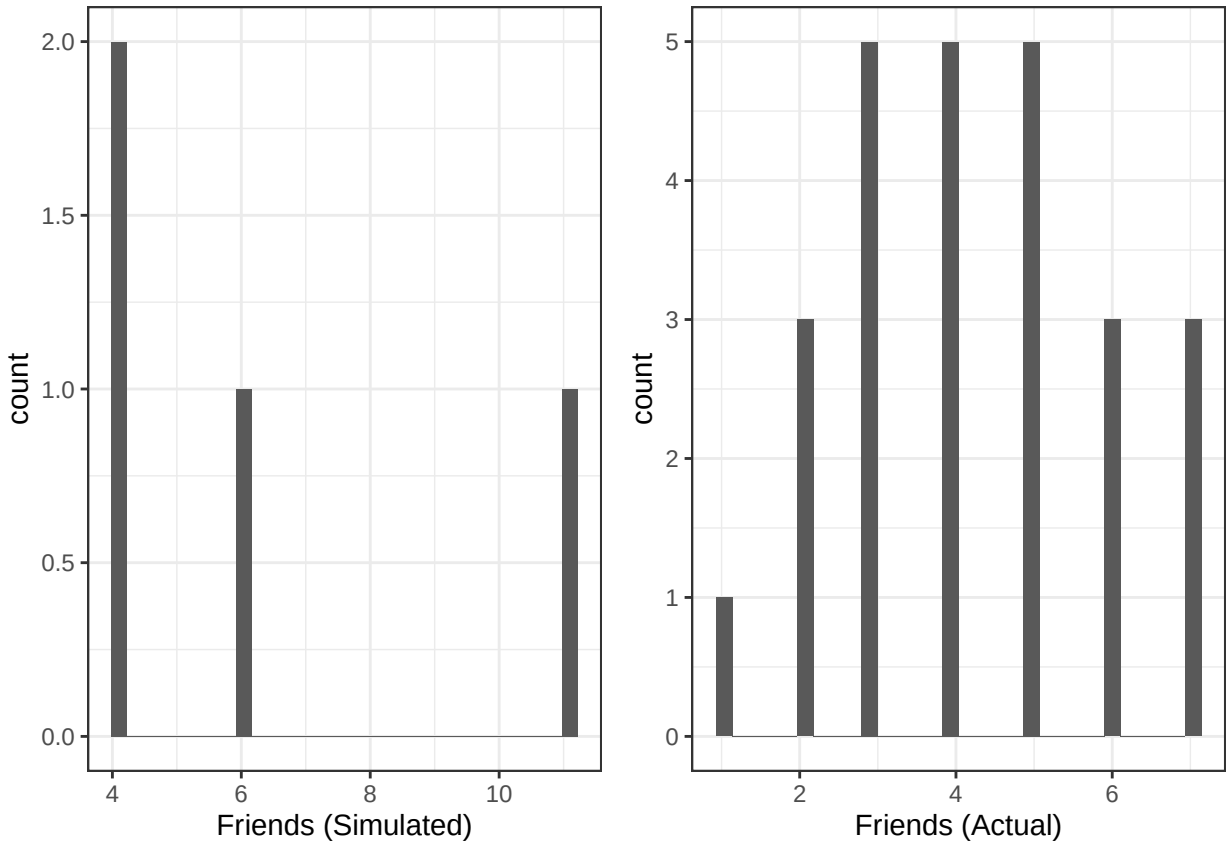
```
#mean and sd in real
dat %>%
  filter(supporter == 1) %>%
  filter(D == 1) %>%
  summarise(mean_actual = mean(friends, na.rm = T),
            sd_actual = sd(friends, na.rm = T))
```

```
##   mean_actual sd_actual
## 1        4.24  1.690168
```

```
#plots
p1 <-
  dat %>%
  filter(supporter == 1) %>%
  filter(simTreat == 1) %>%
  ggplot(aes(x = friends)) +
  geom_histogram() +
  labs(x = "Friends (Simulated)")+
  theme_bw()

p2 <-
  dat %>%
  filter(supporter == 1) %>%
  filter(D == 1) %>%
  ggplot(aes(x = friends)) +
  geom_histogram() +
  labs(x = "Friends (Actual)") +
```

```
  theme_bw()

#put them together
ggpubr::ggarrange(p1, p2)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
#test difference
t.test(dat[dat$simTreat == 1 & dat$supporter == 1, "friends"],
       dat[dat$D == 1 & dat$supporter == 1, "friends"])
```

```
##
##  Welch Two Sample t-test
##
## data:  dat[dat$simTreat == 1 & dat$supporter == 1, "friends"] and dat[dat$D == 1 & dat$supporter ==
## t = 1.192, df = 3.2558, p-value = 0.3128
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -3.125616  7.145616
## sample estimates:
## mean of x mean of y
##      6.25      4.24
```

iii.

```
#ii. balance among non-supporters

#mean and sd in simulated treat
dat %>%
  filter(supporter == 0) %>%
  filter(simTreat == 1) %>%
  summarise(mean_simulate = mean(friends, na.rm = T),
            sd_simulate = sd(friends, na.rm = T))
```

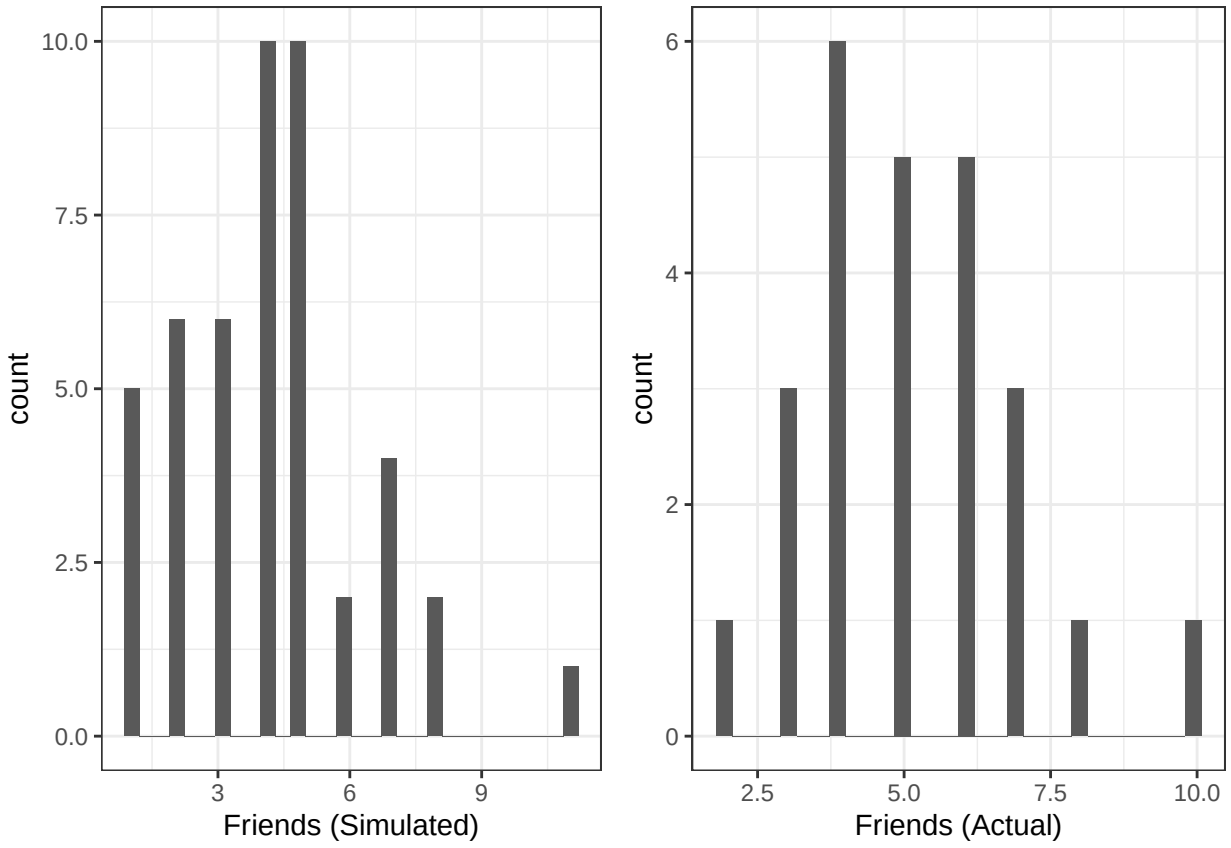
```
##   mean_simulate sd_simulate
## 1      4.173913    2.153079
```

```
#mean and sd in real
dat %>%
  filter(supporter == 0) %>%
  filter(D == 1) %>%
  summarise(mean_actual = mean(friends, na.rm = T),
            sd_actual = sd(friends, na.rm = T))
```

```
##   mean_actual sd_actual
## 1        5.16  1.795364
```

```
#plots
p1 <-
  dat %>%
  filter(supporter == 0) %>%
  filter(simTreat == 1) %>%
  ggplot(aes(x = friends)) +
  geom_histogram() +
  labs(x = "Friends (Simulated)")+
  theme_bw()

p2 <-
  dat %>%
  filter(supporter == 0) %>%
  filter(D == 1) %>%
  ggplot(aes(x = friends)) +
  geom_histogram() +
  labs(x = "Friends (Actual)") +
  theme_bw()

#put them together
ggpubr::ggarrange(p1, p2)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
#test difference
t.test(dat[dat$simTreat == 1 & dat$supporter == 0, "friends"],
       dat[dat$D == 1 & dat$supporter == 0, "friends"])
```

```
##
##  Welch Two Sample t-test
##
## data:  dat[dat$simTreat == 1 & dat$supporter == 0, "friends"] and dat[dat$D == 1 & dat$supporter == 0
## t = -2.0574, df = 57.458, p-value = 0.04419
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -1.94566508 -0.02650883
## sample estimates:
## mean of x mean of y
##  4.173913  5.160000
```

In the network as a whole, the mean number of friends among individuals assigned to treatment via simulation versus individuals assigned to treatment in reality were not significantly different. Similarly, there was not a signficant difference between the mean degree of a) supporters and b) non-supporters randomized to treatment via simulation versus reality. In the whole sample, the distribution of friends was similar to that in the real data, while in supporters and non-supporters, the distributions of friends were left-skewed in comparison to more normal distributions in the real data

d.

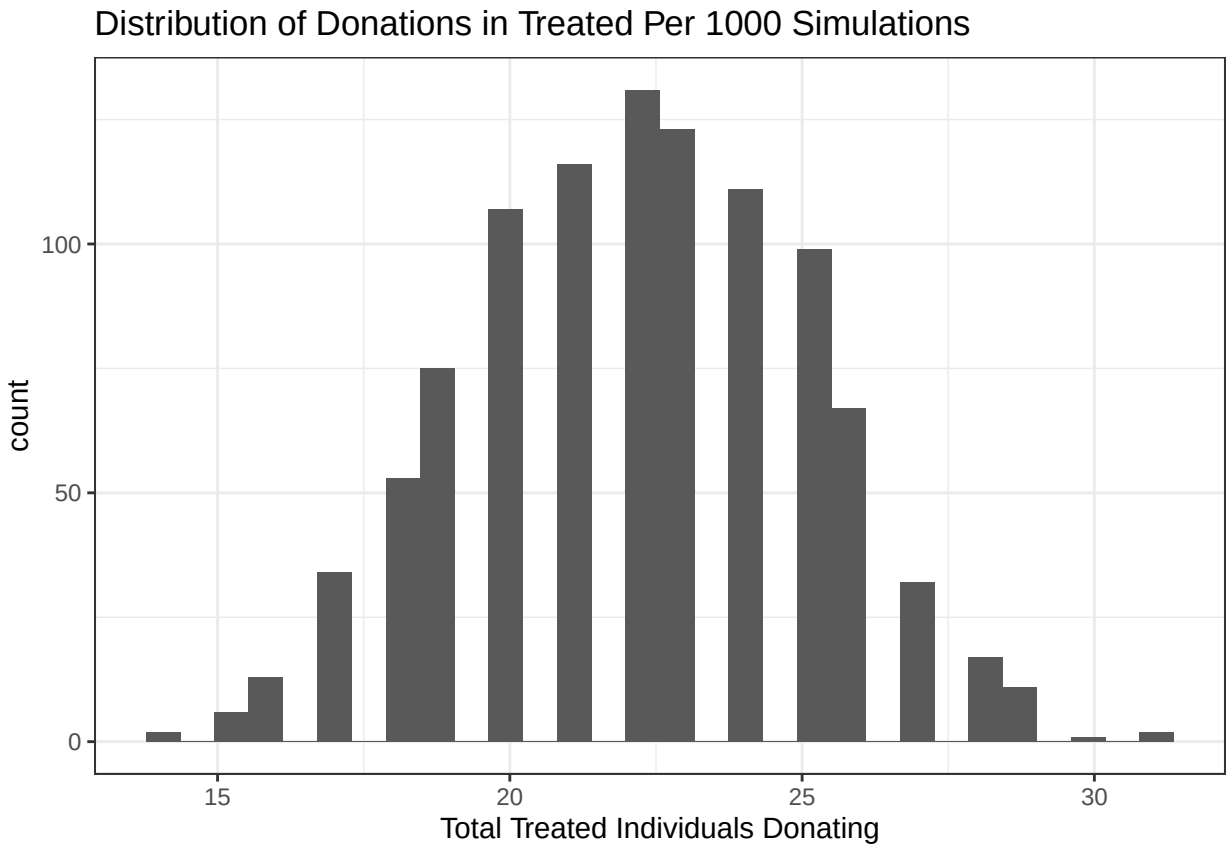
```r
#modify to compute donations
#function to rerandomize treatment:
#1/2 supporters get treatmet + 1 non-supporter friend of each supporter
reRandomize <- function(){
  Gcopy <- G
  sampleVec <- as.numeric(dat[dat$supporter == 1, "id"])

  #randomly assign treatment to 50% of supporters
  support.treat <- sample(x = sampleVec, size = 25, replace = F)


  #now randomly pick a nonsupporting friend from those who were treated
  #picked IDs of treated group:
  treat.id <- support.treat

  #empty vector for friend ids
  friend.id <- c(rep(99999, 25))

  #loop over matrix, selecting treated rows and sampling their friends
  for (i in 1:25){
    #select row for treated supporter : make it a matrix rowwise
    friend.row <- t(as.matrix(Gcopy[treat.id[i],]))

    #if they only have one friend, that friend gets treatment
    #else, sample from all possible friends
    if (rowSums(friend.row) == 1){
      friend.id[i] <- which(friend.row == 1)

    } else {
      friend.id[i] <- sample(x = unique(which(friend.row == 1)), size = 1, replace = F)

    }
  }

  #concatenate all the treated ids
  all.treat <- append(treat.id, friend.id)


  #new vector for if the selected individual got treatment
  dstar <- ifelse(dat$id %in% all.treat, 1, 0)
  return(sum(dstar*dat$donate))

}
```

i.

```r
#run 10000 times
tsum <- replicate(1000, reRandomize(), simplify = "vector")

#make a df for plotting
tsum <- as.data.frame(tsum)

#plot it
```

```r
tsum %>%
  ggplot(aes(x = tsum)) +
  geom_histogram() +
  labs(x = "Total Treated Individuals Donating") +
  theme_bw() +
  ggtitle("Distribution of Donations in Treated Per 1000 Simulations")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



This distribution represents the conditional expectation function of the total number of donations received from individuals (conditional on being in the treatment group). The null hypothesis we are using this distribution to evaluate is that the observed treatment effect in the treated in the real experiment is a good approximation of (equal to) the expected value of the treatment effect in the treated, thus indicating that the effect of possible confounding by degree in the original experiment is minimal and the experiment was truly random:

$$H_0 : \mathbb{E}[\hat{\tau}(X_i)|D_i = 1] = \mathbb{E}[Y_i|D_i = 1]$$
$$H_1 : \mathbb{E}[\hat{\tau}(X_i)|D_i = 1] \neq \mathbb{E}[Y_i|D_i = 1]$$

ii.

```r
#ii
#calculate actual tsum value
tsum.actual <- sum(dat$D * dat$donate)
```

```r
#add vertical line to plot
tsum %>%
  ggplot(aes(x = tsum)) +
  geom_histogram() +
  geom_vline(xintercept = tsum.actual, color = "red") +
  labs(x = "Total Treated Individuals Donating") +
  theme_bw() +
  ggtitle("Distribution of Donations in Treated Per 1000 Simulations")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

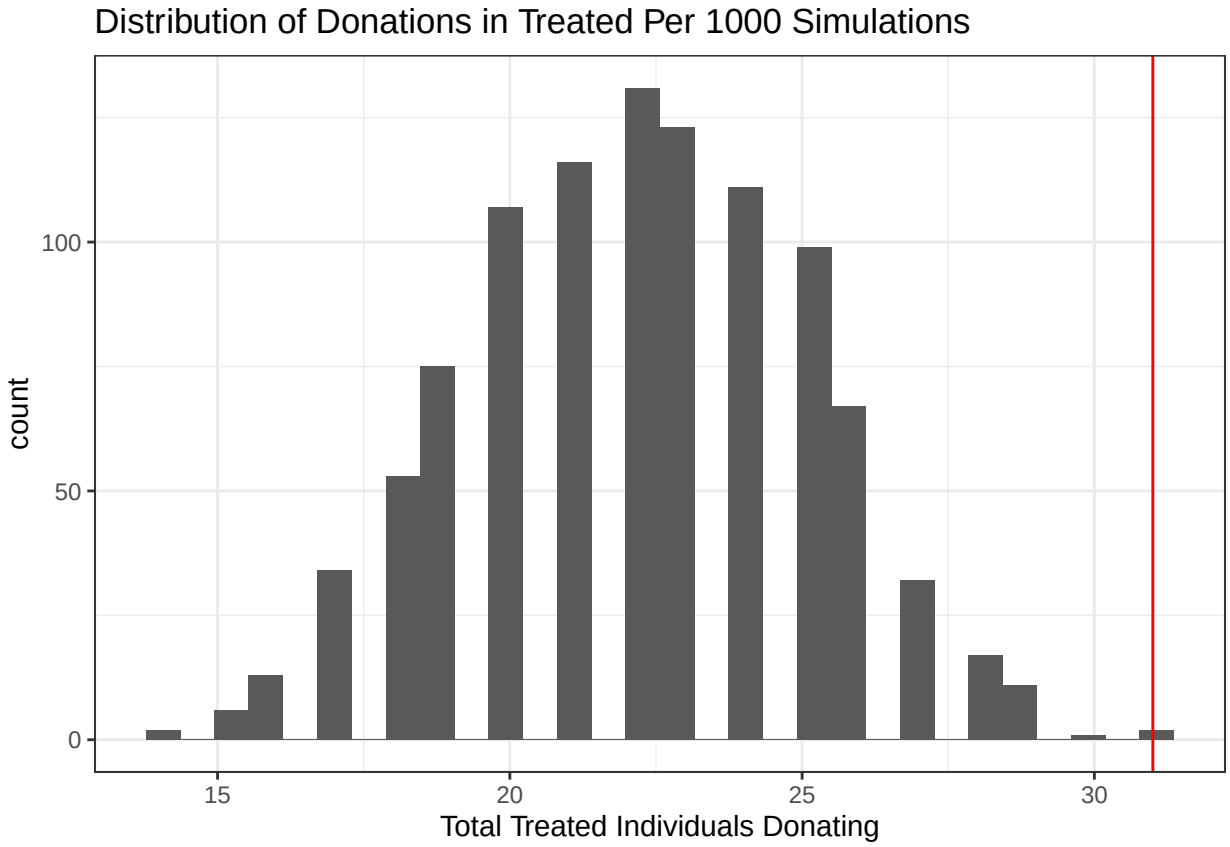


Figure 5: Comparison of Actual Tsum with Simulated Distribution

iii.

```r
#iii

#calculate a test statistic comparing actual Tsum to simulated
#the simulated distribution is approximately normal: one sample t test
t.test(x = tsum$tsum, mu = tsum.actual)
```

```
##
```

```
##  One Sample t-test
##
## data:  tsum$tsum
## t = -95.531, df = 999, p-value < 2.2e-16
## alternative hypothesis: true mean is not equal to 31
## 95 percent confidence interval:
##  22.02128 22.38272
## sample estimates:
## mean of x
##    22.202
```

The realized value of Tsum was significantly different from the mean simulated value of Tsum ($p < 0.001$). This indicates that the value of the treatment effect in the treated in the actual experiment is not a good approximation of the expected treatment effect in the treated.

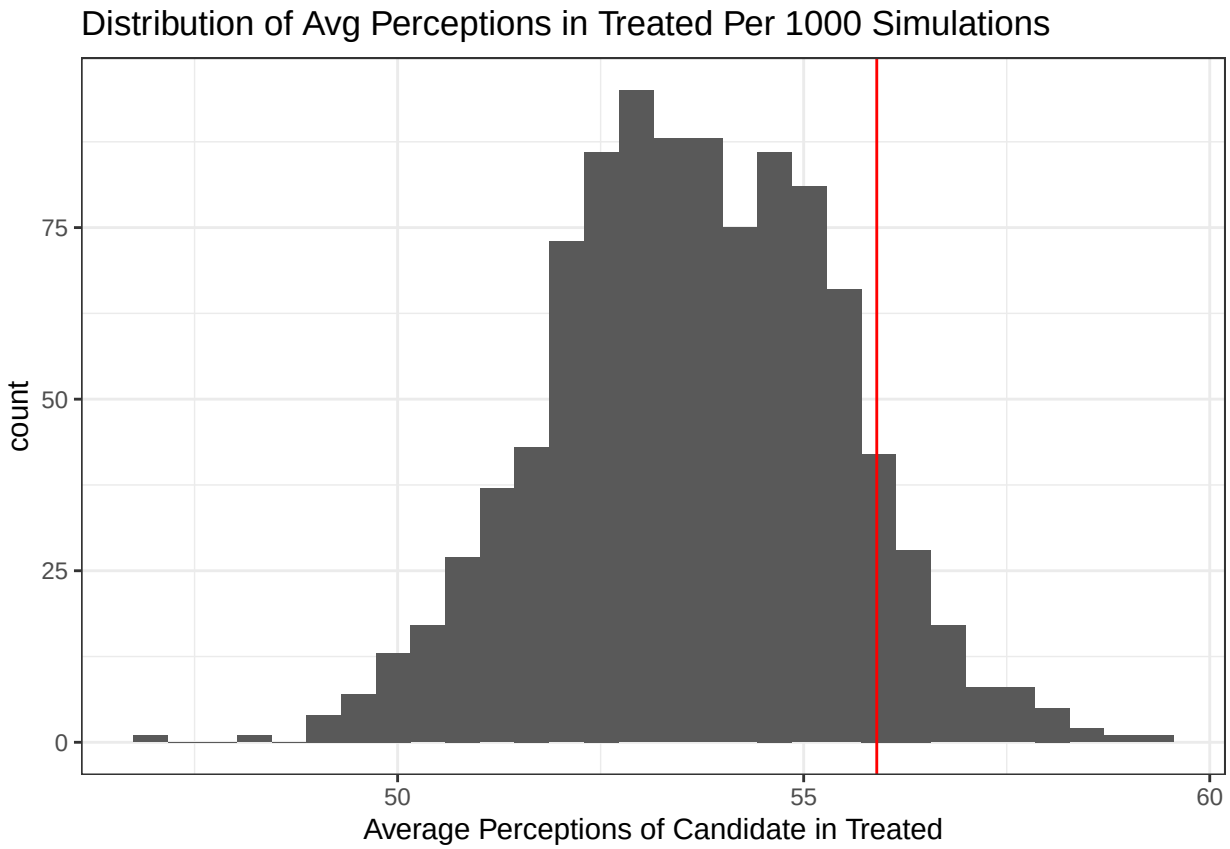
e. In the initial data from the experiment, I found that non-supporters who recieved the campaign message had a greater number of friends than non-supporters who didn't recieve the message. This is potentially a problem for evaluating the effect of the intervention, because it indicates that those who received treatment differ systematically from those who did not. We want to know if the message will work to increase donations in individuals as a whole, and not base that evaluation on individuals with high numbers of friends only, who may have been more likely to donate in the first place. To assess whether the experiment already run was a good approximation of the expected effect of the campaign message, I simulated the experiment 10,000 times by randomly assigning the campaign donation message treatment to half of supporters and one non-supporting friend, as in the original experiment. Based on these new random assignments, I determined the expected number of donations that would have been recieved in each of the random trials. From these numbers, I calculated the average number of donations that we would expect, and compared that value to what you actually found. Essentially, the simulation allowed us to see the expected results of the campaign message even if different individuals were selected for the treatment than actually were in reality. Because the value of what we would generally expect to happen given randomization differed from what we actually found in the potentially biased sample, it seems that the over-connectivity of non-supporters in the current experiment may have overinflated our estimate of how effective the message was.

f.

```r
#f.
#retest randomization on perceptions of candidate
#modify function to compute Di*Yi of each simulated treatment vector
reRandomizeThermo <- function(){
  Gcopy <- G
  sampleVec <- as.numeric(dat[dat$supporter == 1, "id"])

  #randomly assign treatment to 50% of supporters
  support.treat <- sample(x = sampleVec, size = 25, replace = F)


  #now randomly pick a nonsupporting friend from those who were treated
  #picked IDs of treated group:
  treat.id <- support.treat

  #empty vector for friend ids
  friend.id <- c(rep(99999, 25))

  #loop over matrix, selecting treated rows and sampling their friends
```

```r
  for (i in 1:25){
    #select row for treated supporter : make it a matrix rowwise
    friend.row <- t(as.matrix(Gcopy[treat.id[i],]))

    #if they only have one friend, that friend gets treatment
    #else, sample from all possible friends
    if (rowSums(friend.row) == 1){
      friend.id[i] <- which(friend.row == 1)

    } else {
      friend.id[i] <- sample(x = unique(which(friend.row == 1)), size = 1, replace = F)

    }
  }

  #concatenate all the treated ids
  all.treat <- append(treat.id, friend.id)


  #new vector for if the selected individual got treatment
  dstar <- ifelse(dat$id %in% all.treat, 1, 0)

  #multiply feelings about candidate by dstar,
  #for average, divide by # treated units (50)
  return(sum(dstar*dat$thermo)/50)

}

#run 1,000 times
thermo.res <- replicate(n = 1000, reRandomizeThermo(), simplify = "vector")

#get the actual avg perception of candidate
tavg.actual <- sum(dat$D * dat$thermo)/sum(dat$D == 1)

#make a df for plotting
tavg <- as.data.frame((as.matrix(thermo.res)))
tavg$V1 <- as.numeric(tavg$V1)

#plot it
tavg %>%
  ggplot(aes(x = V1)) +
  geom_histogram() +
  geom_vline(xintercept = tavg.actual, color = "red") +
  labs(x = "Average Perceptions of Candidate in Treated") +
  theme_bw() +
  ggtitle("Distribution of Avg Perceptions in Treated Per 1000 Simulations")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
#calculate test statistic for sharp null: % of simulations where perception was > observed
sharp.null <- sum(tavg$V1 > tavg.actual) / length(tavg$V1)

#2 sided p value= this *2
sharp.null*2
```

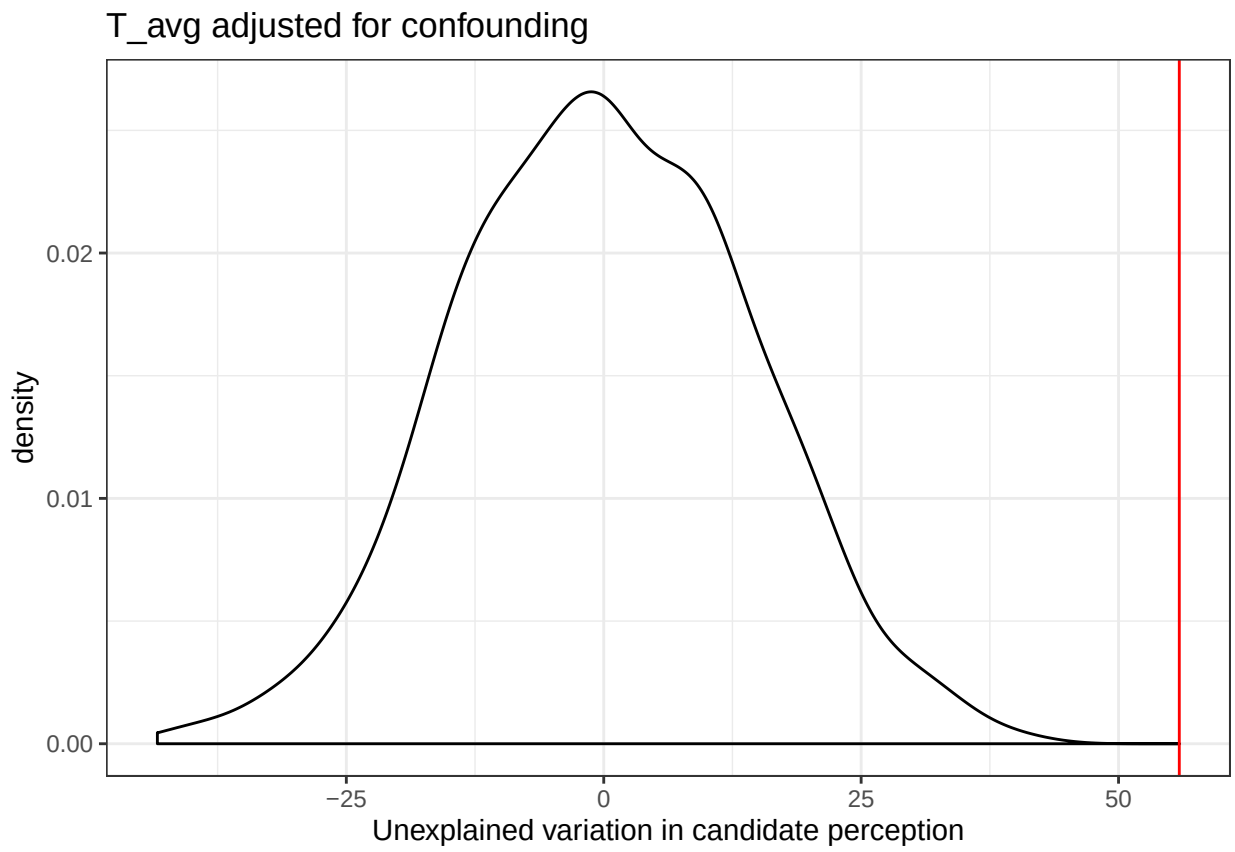
```
## [1] 0.182
```

This tells us that the perceptions of the candidate due to the fundraising campaign are not extreme enough to be different from what would be expected by chance. Essentially, it seems that the campaign was not effective in increasing ratings of the candidate.

g.

```
#g.
#calculate tsumavg using residuals
mod2 <- lm(thermo   supporter* factor(friends), data = dat)

#get the residuals for treated only
dat$resid <- residuals(mod2)

#plot residuals against tavg actual
dat %>%
  ggplot(aes(x = resid))+
```

```
  geom_density()+
  geom_vline(xintercept = tavg.actual, color = "red") +
  labs(x = "Unexplained variation in candidate perception")+
  theme_bw()+
  ggtitle("T_avg adjusted for confounding")
```




```
#calculate test statistics
test2 <- sum(tavg.actual > dat$resid ) / length(tavg$V1)
test2
```

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

These results suggest that perceptions of the candidate adjusted for confounding by number of friends and supporter status are not significantly different from 0. This suggests no effect of the campaign (which is captured by the residuals).

h. No, we cannot answer this question with the given data. We were only testing the sharp null hypothesis of no effect for any observations, but this doesn't give us an estimate of magnitude. Under the sharp null hypothesis, we don't compute means and standard errors- each observation is compared individually to the overall cut-off and the p-value comes from the percentage that exceed that cutoff. Without the mean and standard error, we can't estimate magnitude.

Question 2

a) A DAG is useful for conceptualizing the causal relationships among sets of variables/constructs, and it helps you identify specifically what you need to control for to estimate (unbiased) effects of interest, as well as what you should not control for at risk of induced other causal dependencies. It buys you additional insight into *how* you think X causes Y, and gives you more room to make causal hypotheses/conclusions in observational data. However, it is not necessarily a testable hypothesis in that you cannot generally test if your proposed DAG is 'true': your conceptualization of how variables relate may be wrong, but this is not testable with the DAG.

b) The backdoor criterion is the statement that, in identifying the causal relationship between X and Y, you have blocked all other confounding pathways through which X is otherwise associated with Y, have not opened any pathways by conditioning on colliders, and have not conditioned on variables causally post to X. It is esentially a way of ensuring that the generated estimate of the effect of X on Y actually represents the causal effect of X on Y, and not the effect of X on Y due to other associations.

c) All pathways between X and Y:

$$X -> Y$$
$$X <- Z_1 -> Y$$
$$X -> Z_3 -> Y$$
$$X <- Z_1 <- Z_2 -> Y$$
$$X <- Z_4 -> Z_1 -> Y$$
$$X <- Z_4 -> Z_1 <- Z_2 -> Y$$

d) Z_1 is a collider. If you condition on it, you unblock the path.

e) Z_1 is a confounder. If you condition on it, you block the path.

f) No, conditioning on Z_1 only does not satisfy the back door criterion, because it opens up the pathway $X <- Z_4 -> Z_1 <- Z_2 -> Y$ where Z_1 is a collider. Thus, there remains an unblocked path between X and Y when only Z_1 is conditioned on because we induce association between Z_4 and Z_2, and estimates would not represent the 'true' causal effect.

g) The minimally sufficient sets to estimate the effect of X on Y are:

- both Z_1 and Z_2 or
- both Z_1 and Z_4