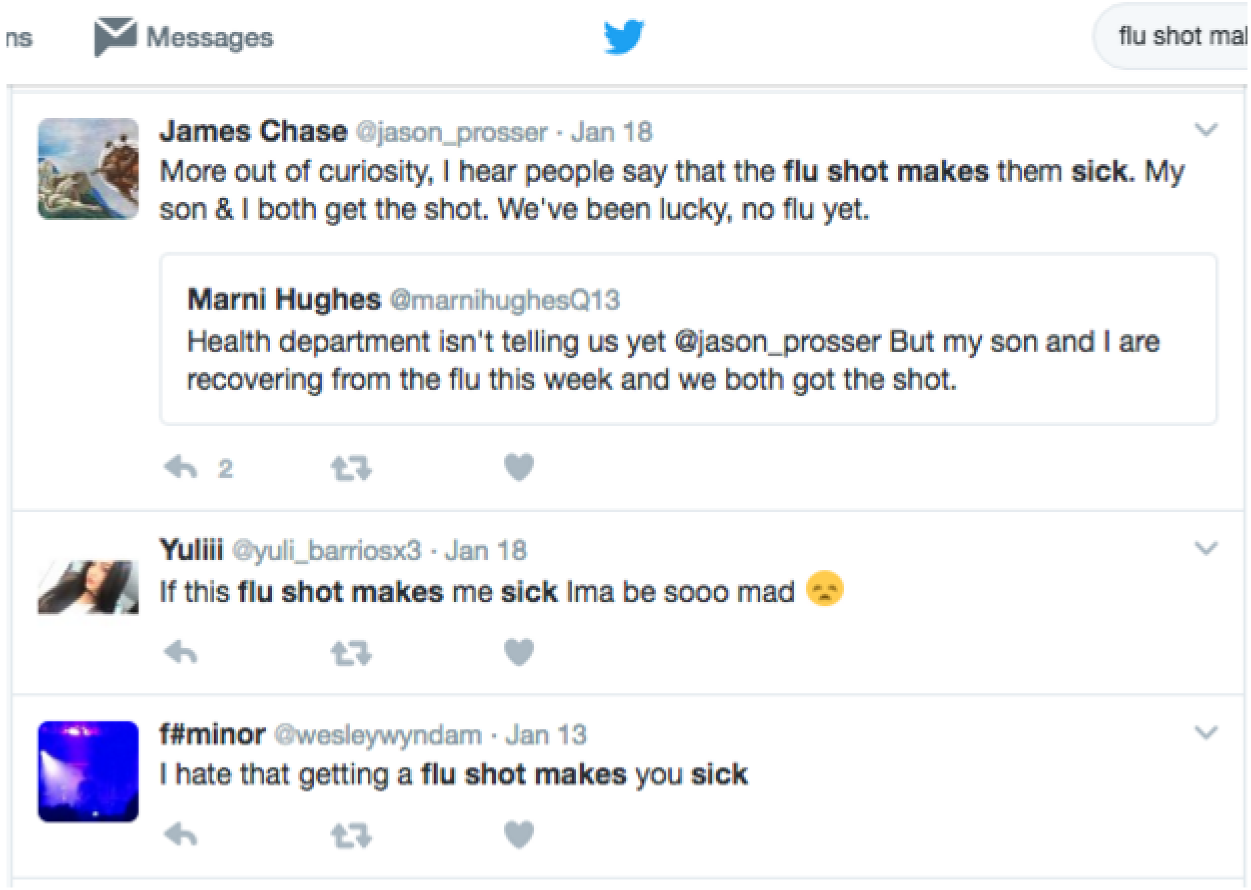
Twitter Flu Shot Misconceptions Field Experiment

Marcus Demaster, Shih Yu Chang and Allan Lo (Dedicated)

May 2, 2017

Introduction

This study attempts to measure the effectiveness of posting educational feedback as a reply to a post on Twitter stating a falsehood. Specifically, this field experiment tests whether linking to the CDC's educational resource page on flu shots slows the spread of misconceptions that flu shots are unsafe or ineffective on Twitter. Public health officials could find the results of this experiment useful. There is an ongoing struggle to correct widely debunked claims of health hazards ranging from child vaccinations to fluoridation of municipal water.



James Chase @jason_prosser · Jan 18
More out of curiosity, I hear people say that the **flu shot makes** them **sick**. My son & I both get the shot. We've been lucky, no flu yet.

Marni Hughes @marnihughesQ13
Health department isn't telling us yet @jason_prosser But my son and I are recovering from the flu this week and we both got the shot.

Yuliii @yuli_barriosx3 · Jan 18
If this **flu shot makes** me **sick** Ima be sooo mad 😞

f#minor @wesleywyndam · Jan 13
I hate that getting a **flu shot makes** you **sick**

Figure 1: Figure 1. Example tweets claiming flu shots are unsafe

Even broader implications can be drawn from this experiment given the current debate around the spread of misinformation and "fake news" on social media. Facebook is testing algorithms to flag shared "fake news" articles as "disputed" in Germany prior to their national elections [1]. With its open social media platform and large volume of discussed topics, we believe Twitter offers a promising environment for testing the effectiveness of any range of educational outreach methods through the objective outcome measurement of a tweet's retweet and favorite counts following an outreach reply.

Initial Experimental Design and Challenges

The design of the experiment underwent a number of modifications over the course of the project. Initially, the goal was to automate the entire process of data collection and analysis. This would entail the use of the Twitter Rest API to search tweets that contain the key phrase, “flu shot,” but also the filtering of these tweets for those that indicate negative sentiment toward the flu shot’s safety and effectiveness at flu prevention. We tested the Python NLTK Vader package for scoring the sentiment of the tweets. However, this scoring filter lacked the specificity we needed to identify negative tweets directed towards the flu shot’s safety and efficacy.

Furthermore, our goal was to automate our treatments of the collected subjects. Using Twitter’s write API, it is possible to develop a script for sending replies to specific tweets. However, Twitter has developed safeguards to detect and block automated writes that are suspected of being spam. From our reading about other projects, it appears possible to develop around these safeguards. The team decided this would risk consumption of too much time and resources, so we opted to go the manual route of collecting and sending replies to the flu shot tweets of interest.

One last challenge we encountered was the effect of flu seasonality on the volume of tweet subjects we could collect. With the original goal of using only US tweets, we were finding only 1-5 tweets of interest per day in late March. However, the flu season is starting up in the southern hemisphere at this time of year. Expanding our REST API parameters to include worldwide English tweets allowed us to collect up to 20 negative flu shot tweets per day, making our goal of collecting over 100 subjects much more feasible within our time scale.

Final Experimental Design

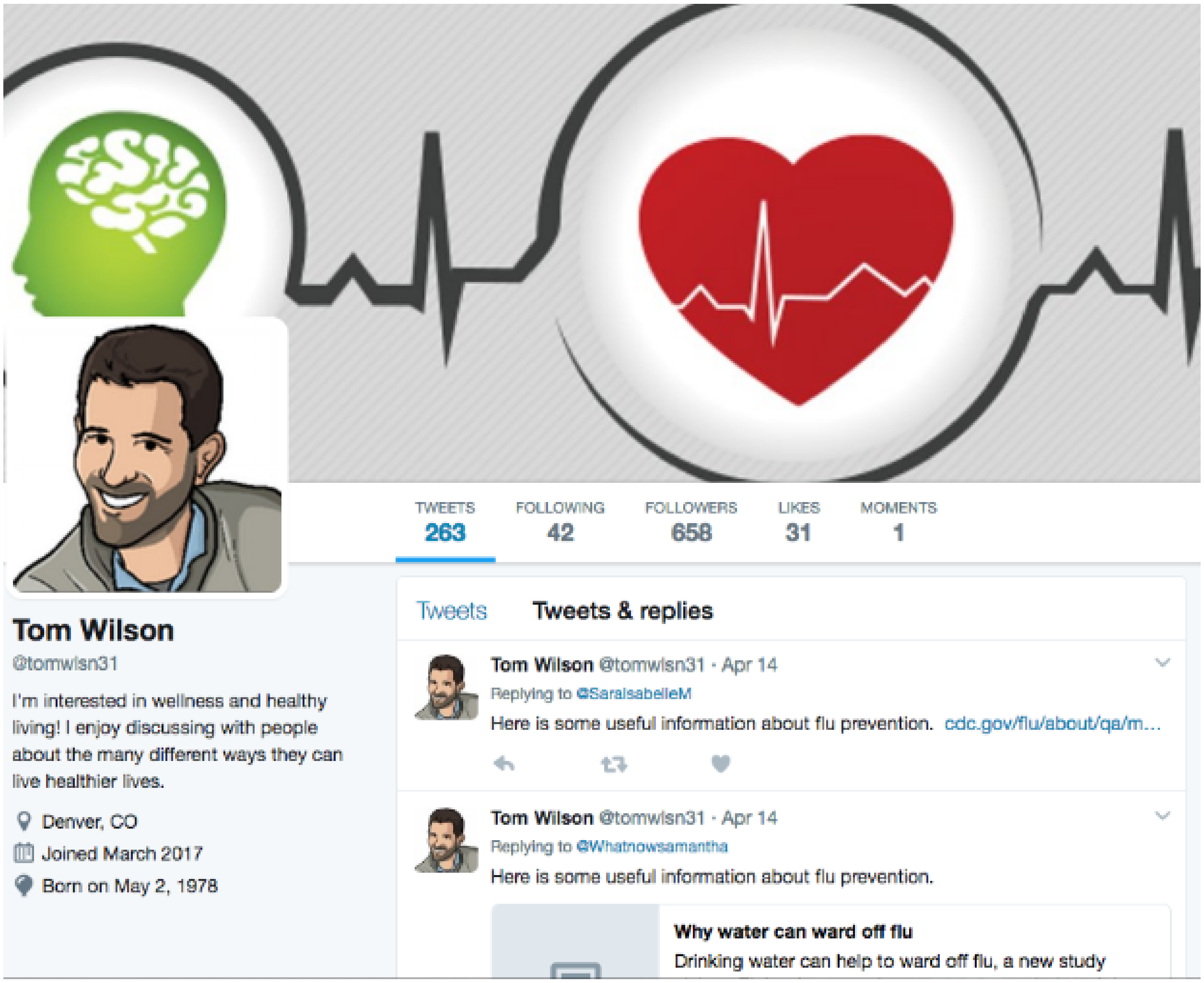
Twitter Profile

Before we could administer replies to our tweets of interest, we put a large amount of consideration into how the Twitter profile appeared to the tweet poster’s followers. As in any social interaction, the degree to which someone accepts new information from another greatly depends on that information giver’s appearance of trustworthiness. In order to optimize the receptiveness of our profile to these twitter followers, we used an avatar of a smiling well-groomed male named Tom Wilson who describes himself in the following way. “I’m interested in wellness and healthy living! I enjoy discussing with people about the many different ways they can live healthier lives.” The profile was also set up to follow and like many health and wellness related channels and was used to periodically tweet out links to an array of health and wellness articles. To further give the profile a greater appearance of legitimacy, we purchased 500 Twitter followers using a social media marketing service, Devumi.com.

Experiment Methods

After we decided on a manual tweet search-and-reply method, we created a schedule to repeat this process every 24 hours during the morning on PST hours. The REST API parameters were set to search only “flu shot” tweets from the previous 24 hours and export key covariates to a spreadsheet. The text of the tweets were then manually filtered based on if the user was tweeting their opinion that the flu shot was unsafe or ineffective. Finally, to increase the statistical power of our t-test, we performed blocking of the tweets by ranking them by the number of followers that belong to the user that sent each tweet. We do this because we believe the number of followers will be directly related to the number of retweets and favorites each tweet receives.

We used the ROXO method (Randomize-Observe-Experiment-Observe). After blocking each set of searched and filtered tweets from the previous 24 hours, we randomly assign one of the first two tweets in our ranked list to the treatment group and the other to the control group. We continue this random assignment for each subsequent pair of tweets in our list.



Tom Wilson

@tomwlsn31

I'm interested in wellness and healthy living! I enjoy discussing with people about the many different ways they can live healthier lives.

Denver, CO

Joined March 2017

Born on May 2, 1978

Tom Wilson @tomwlsn31 · Apr 14

Replying to @SaraIsabelleM

Here is some useful information about flu prevention. cdc.gov/flu/about/qa/m...

Tom Wilson @tomwlsn31 · Apr 14

Replying to @Whatnowsamantha

Here is some useful information about flu prevention.

Why water can ward off flu

Drinking water can help to ward off flu, a new study

Figure 2: Figure 2. Experiment Administrator Twitter Profile

Both the treatment and control groups of tweets received a reply with a link to a webpage and the lead-in phrase, "Here is some useful information about flu prevention." We wanted the control group to have a placebo lead-in phrase and link as well so that the only difference measured in outcomes should be attributable to the content of the linked webpage itself.

The treatment link was to the CDC's Q&A page on flu prevention, specifically the subpage regarding flu vaccine misconceptions [2]. The intent is for this page to address the misconceptions espoused by the tweet and inhibit the tweeter's followers from retweeting or liking the tweet. The placebo link, on the other hand, is a Daily Mail article [3] that suggests drinking more water each day will help to prevent the flu. This article does not mention the flu vaccine. However, since it is about the broader topic of flu prevention, it should be received by followers as relevant and not spammy.

As soon as the replies were sent, the baseline measures of retweet and favorite counts were observed and noted. We allowed an 8 day experiment period to pass after each 24-hour administration set. We then observed the retweet and favorite counts for each tweet as the outcome measures. In all, we performed 10 daily administrations and collected 148 tweets/subjects.

Analysis

In this analysis part, we begin by exploratory data analysis (EDA) for the collected Tweets data. We perform three analysis tasks here. First, we analyze generalt ATE by studing causal effect of treatment and control groups. Second, localtion effect, i.e., US V.S. Non-US, is considered. Finally, we consider the effect of using different output measures in our causal analysis. At appendix, we consider the causal effect of non-compliance and analyze causal effect by using predicted missing Tweets about flu shot.

```r
# load packages
library(foreign)
library(astsa)
library(forecast)
```

```
## Warning: package 'forecast' was built under R version 3.3.2

## Loading required package: zoo

##
## Attaching package: 'zoo'

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

## Loading required package: timeDate

## This is forecast 7.3

##
## Attaching package: 'forecast'

## The following object is masked from 'package:astsa':
##
##     gas
```

```
library(car)
library(ggplot2)
library(lattice)
library(car)
library(lmtest)
library(sandwich)
```

0, EDA for Collected Data

```
# Retweet_Outcome, Favorite_Outcome
Flu_Data <- read.csv("W241_Final_Project_Data_v1.csv", header=TRUE)
str(Flu_Data)
```

```
## 'data.frame':    94 obs. of  19 variables:
##  $ User_ID             : num  7.39e+17 2.59e+09 7.66e+17 7.61e+08 4.22e+08 ...
##  $ User_Screen_Name    : Factor w/ 91 levels "_stacey17_","_xoViviana",..: 65 52 16 13 18 31 75 15 8
##  $ User_Statuses_Count : num  108 7982 23700 2337 34285 ...
##  $ User_Followers_Count: num  3621 1715 1575 838 779 ...
##  $ User_Friends_Count  : num  3705 637 1502 1677 323 ...
##  $ User_Listed_Count   : num  34 19 17 11 3 7 21 7 1 4 ...
##  $ User_Timezone       : Factor w/ 21 levels "","Alaska","America/Chicago",..: 1 1 15 17 6 20 1 17 7
##  $ Tweet_ID            : num  8.49e+17 8.49e+17 8.49e+17 8.49e+17 8.49e+17 ...
##  $ Tweet_Text          : Factor w/ 93 levels "\"Safe vaccine\" against swine flu caused narcolepsy i
##  $ Tweet_Retweet_Count : num  0 0 1 0 0 0 1 1 0 0 ...
##  $ Tweet_Favorite_Count: num  1 3 0 0 0 0 0 0 2 0 ...
##  $ Tweet_Created_At    : Factor w/ 94 levels "4/2/17 12:44",..: 14 13 30 17 4 10 1 16 26 11 ...
##  $ Assign_Ind          : num  1 0 1 0 0 1 1 0 1 0 ...
##  $ Assign_Date         : Factor w/ 6 levels "4/3/17","4/4/17",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ Reply               : Factor w/ 3 levels "\nHere is some useful information about flu prevention.
##  $ Retweet_Outcome     : num  0 0 0 0 0 0 1 1 0 0 ...
##  $ Favorite_Outcome    : num  2 3 0 0 0 0 0 1 3 1 ...
##  $ Sex                 : int  0 1 1 0 0 1 1 1 1 0 ...
##  $ Tweets_pos_rate     : num  0.3776 0.1339 0.0755 0.1516 0.0328 ...
```

User_Statuses_Count

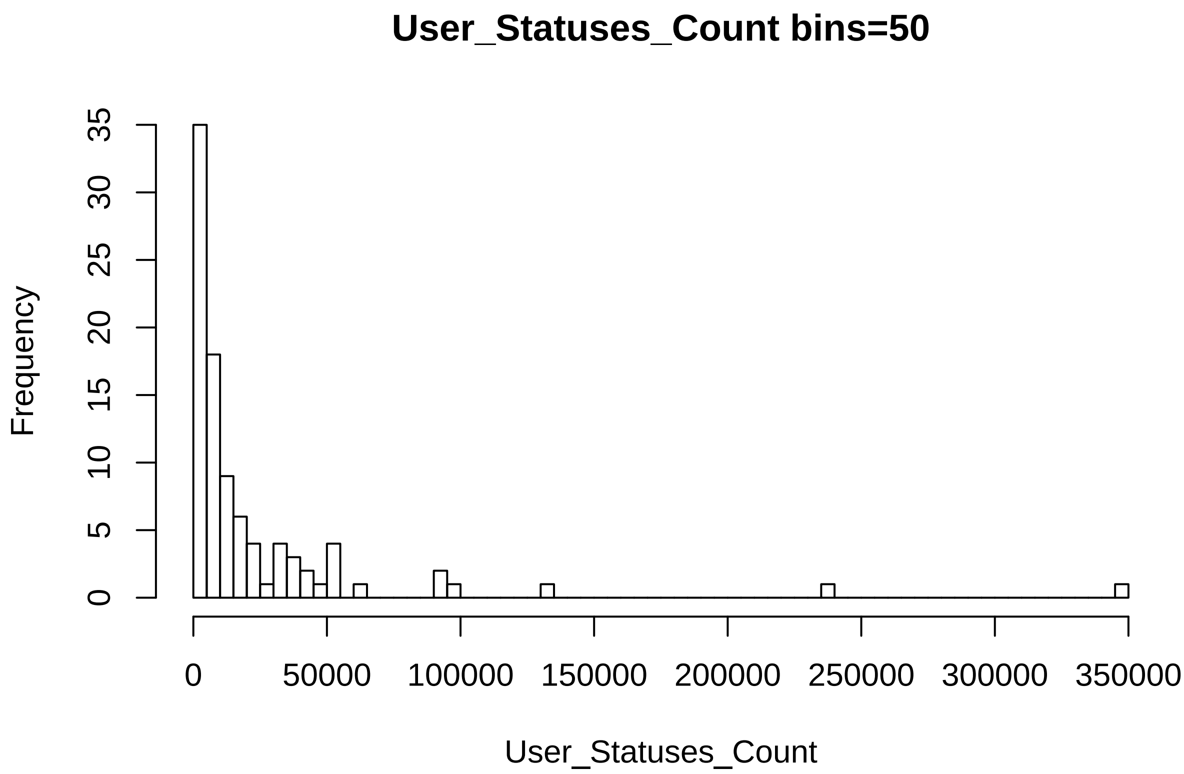
```
summary(Flu_Data$User_Statuses_Count)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##       3    2343    8580   23130   22840  348600
```

```
quantile(Flu_Data$User_Statuses_Count, probs=c(1,5,10,25,50,75,90,99, NA)/100)
```

```
##        1%        5%       10%       25%       50%       75%       90% 
##     87.63    348.95    942.90   2343.00   8580.00  22840.25  51922.10 
##       99%           
## 244856.52        NA
```

```r
hist(Flu_Data$User_Statuses_Count, main='User_Statuses_Count bins=50', xlab='User_Statuses_Count',
breaks=50)
```



User__Followers__Count

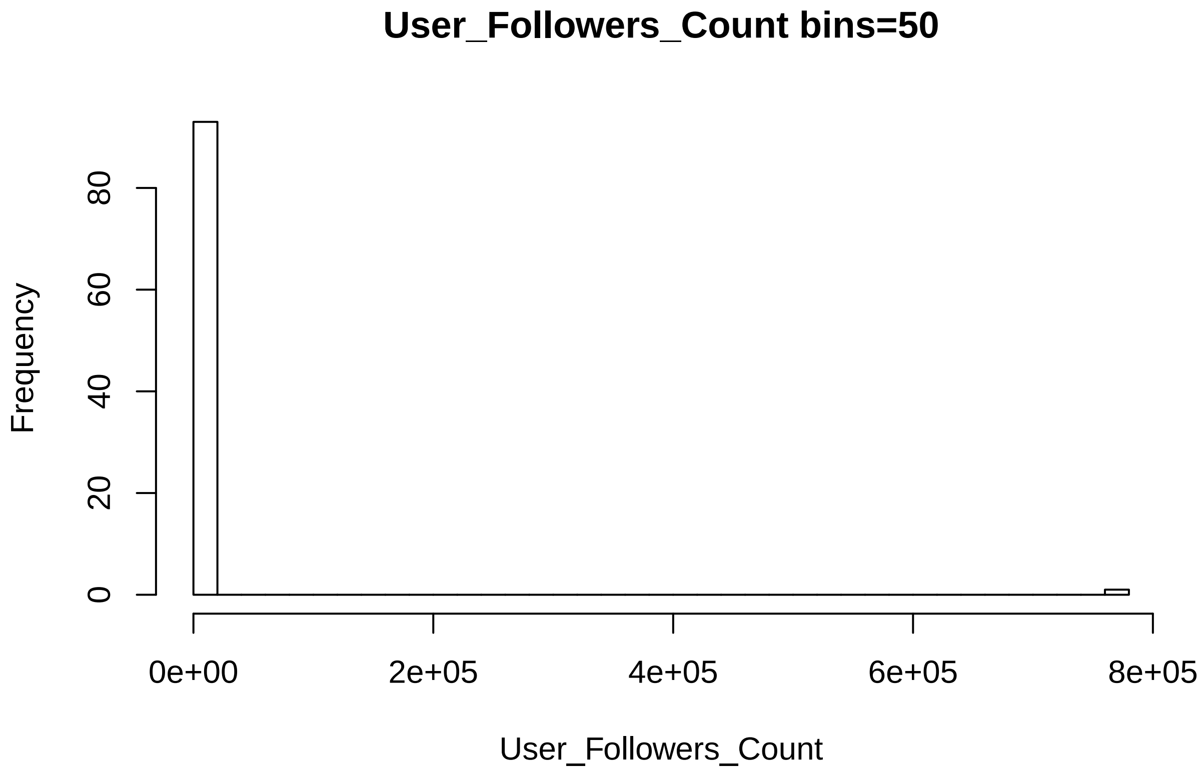
```r
summary(Flu_Data$User_Followers_Count)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##      1.0    188.2    393.0   9620.0    879.2 778000.0
```

```r
quantile(Flu_Data$User_Followers_Count, probs=c(1,5,10,25,50,75,90,99, NA)/100)
```

```
##       1%       5%      10%      25%      50%      75%      90%      99%
##     3.79    69.20   103.60   188.25   393.00   879.25  3112.80 72281.41
##
##       NA
```

```r
hist(Flu_Data$User_Followers_Count, main='User_Followers_Count bins=50', xlab='User_Followers_Count',
breaks=50)
```



User_Friends_Count

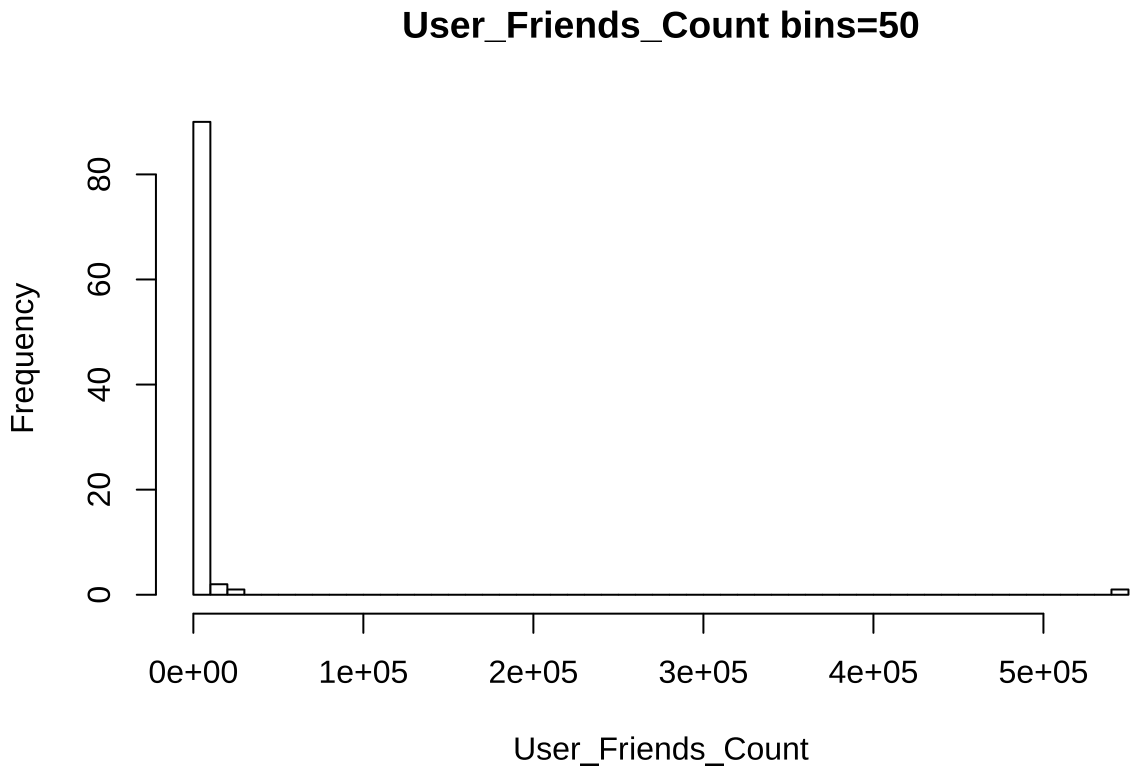
```
summary(Flu_Data$User_Friends_Count)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##     22.0    252.5    405.5   7008.0    821.5 541200.0
```

```
quantile(Flu_Data$User_Friends_Count, probs=c(1,5,10,25,50,75,90,99, NA)/100)
```

```
##       1%       5%      10%      25%      50%      75%      90%      99%
##    30.37    89.30   146.60   252.50   405.50   821.50  2231.30 57443.67
##
##       NA
```

```
hist(Flu_Data$User_Friends_Count, main='User_Friends_Count bins=50', xlab='User_Friends_Count',
breaks=50)
```



User__Listed__Count

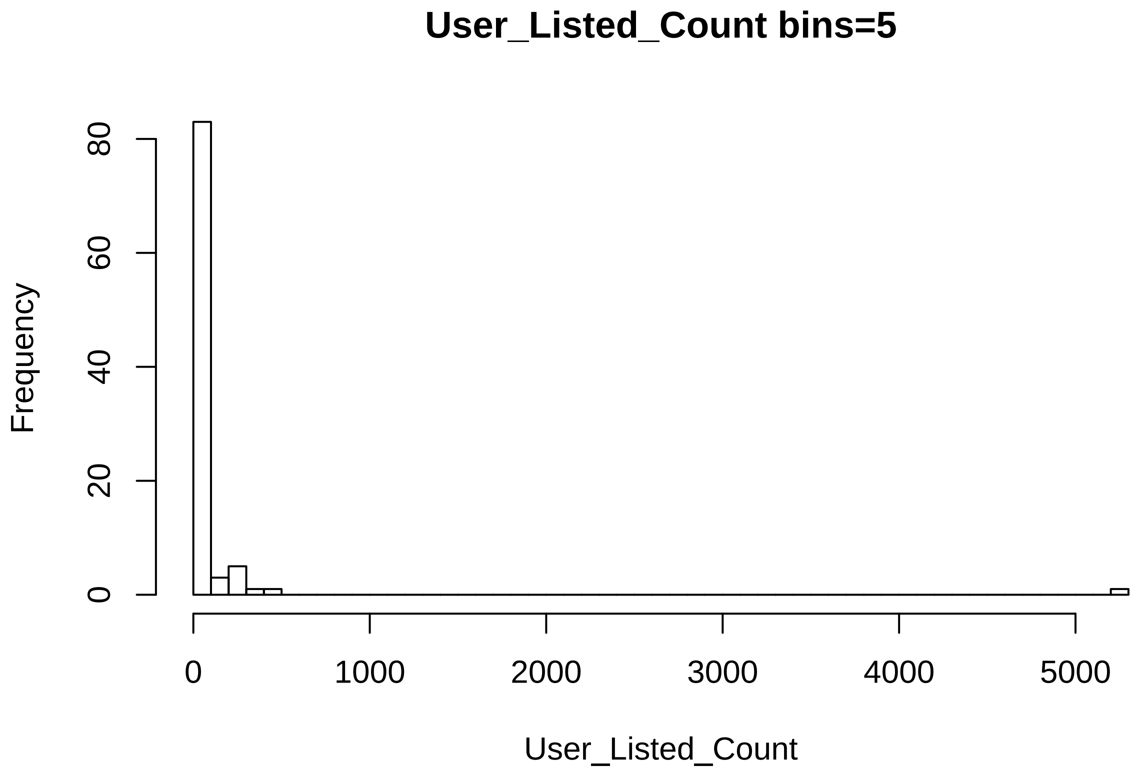
```
summary(Flu_Data$User_Listed_Count)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     0.0     2.0     6.5    92.0    23.5  5208.0
```

```
quantile(Flu_Data$User_Listed_Count, probs=c(1,5,10,25,50,75,90,99, NA)/100)
```

```
##     1%      5%     10%     25%     50%     75%     90%     99%
##   0.00    0.00    0.00    2.00    6.50   23.50  145.40  789.57      NA
```

```
hist(Flu_Data$User_Listed_Count, main='User_Listed_Count bins=5', xlab='User_Listed_Count',
breaks=50)
```



Tweet_Retweet_Count

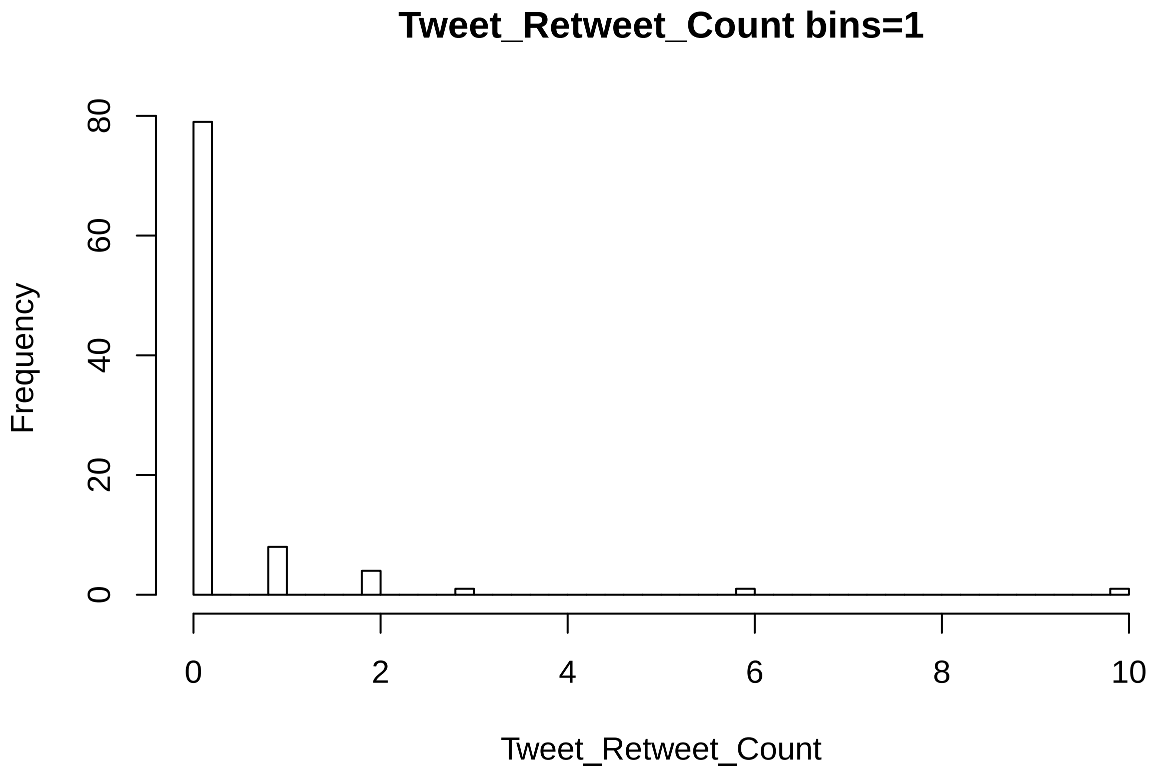
```
summary(Flu_Data$Tweet_Retweet_Count)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0000  0.0000  0.0000  0.3723  0.0000 10.0000
```

```
quantile(Flu_Data$Tweet_Retweet_Count, probs=c(1,5,10,25,50,75,90,99, NA)/100)
```

```
##   1%   5%  10%  25%  50%  75%  90%  99%
## 0.00 0.00 0.00 0.00 0.00 0.00 1.00 6.28   NA
```

```
hist(Flu_Data$Tweet_Retweet_Count, main='Tweet_Retweet_Count bins=1', xlab='Tweet_Retweet_Count',
breaks=50)
```



Tweet_Favorite_Count

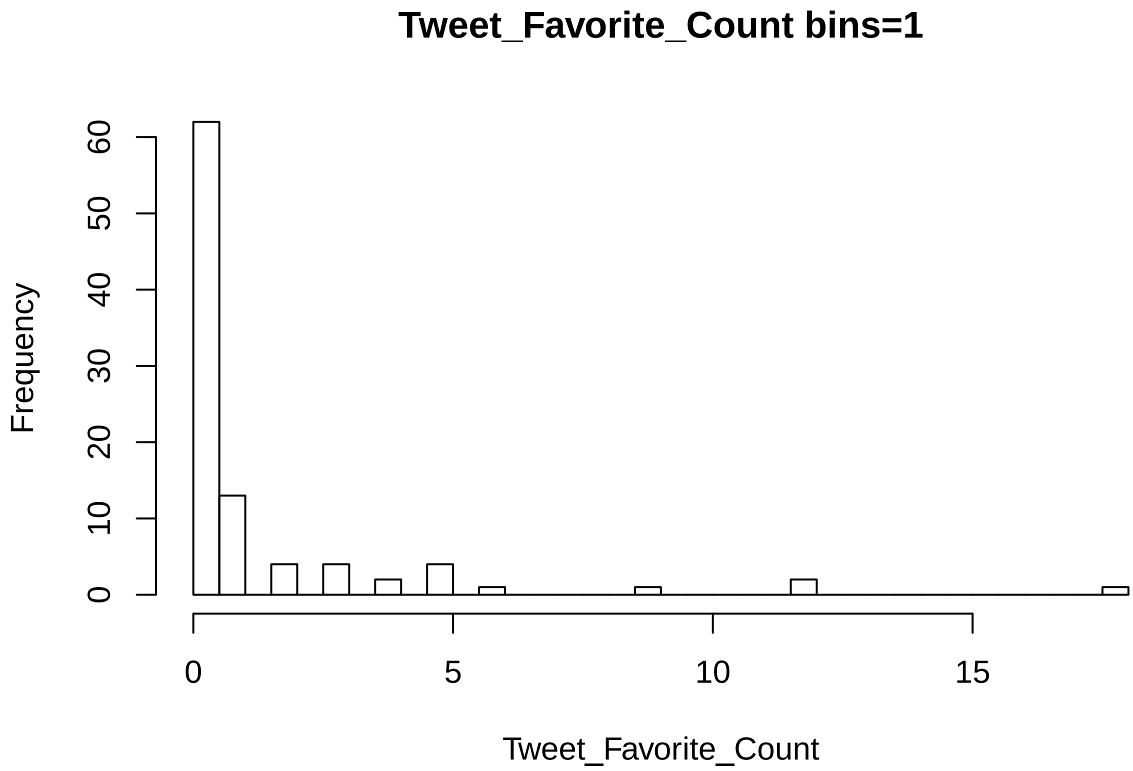
```
summary(Flu_Data$Tweet_Favorite_Count)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   0.000   0.000   1.255   1.000  18.000
```

```
quantile(Flu_Data$Tweet_Favorite_Count, probs=c(1,5,10,25,50,75,90,99, NA)/100)
```

```
##    1%    5%   10%   25%   50%   75%   90%   99%
##  0.00  0.00  0.00  0.00  0.00  1.00  4.00 12.42    NA
```

```
hist(Flu_Data$Tweet_Favorite_Count, main='Tweet_Favorite_Count bins=1', xlab='Tweet_Favorite_Count',
breaks=50)
```



Retweet_Outcome

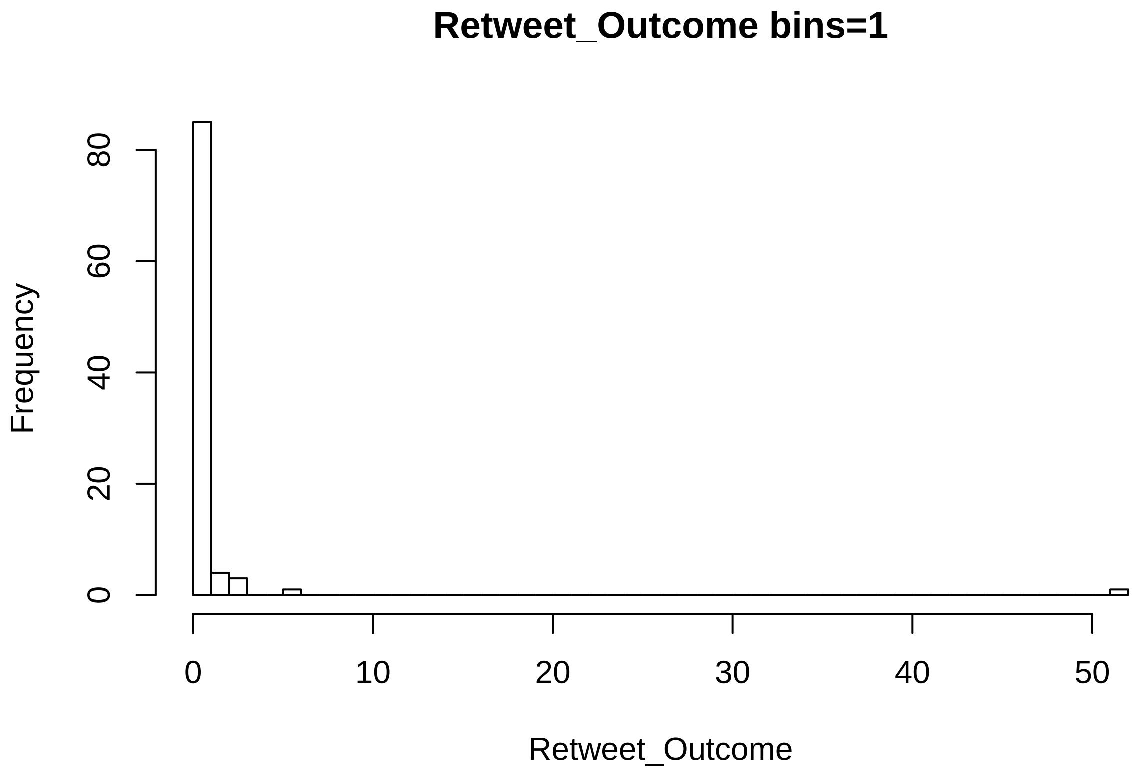
```
summary(Flu_Data$Retweet_Outcome)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##  0.0000  0.0000  0.0000  0.9043  0.0000 52.0000
```

```
quantile(Flu_Data$Retweet_Outcome, probs=c(1,5,10,25,50,75,90,99, NA)/100)
```

```
##   1%   5%  10%  25%  50%  75%  90%  99%      
## 0.00 0.00 0.00 0.00 0.00 0.00 1.00 9.22   NA
```

```
hist(Flu_Data$Retweet_Outcome, main='Retweet_Outcome bins=1', xlab='Retweet_Outcome',
breaks=50)
```



Favorite_Outcome

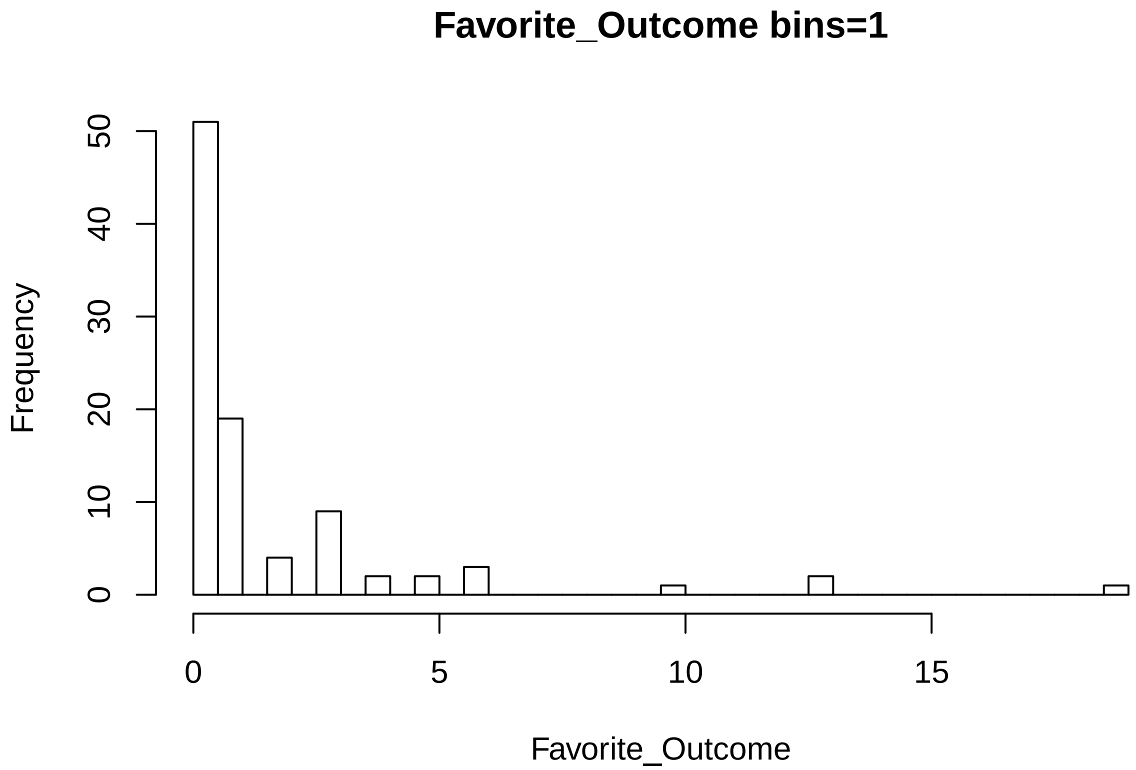
```
summary(Flu_Data$Favorite_Outcome)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   0.000   0.000   0.000   1.543   1.750  19.000
```

```
quantile(Flu_Data$Favorite_Outcome, probs=c(1,5,10,25,50,75,90,99, NA)/100)
```

```
##    1%    5%   10%   25%   50%   75%   90%   99%       
##  0.00  0.00  0.00  0.00  0.00  1.75  4.00 13.42    NA
```

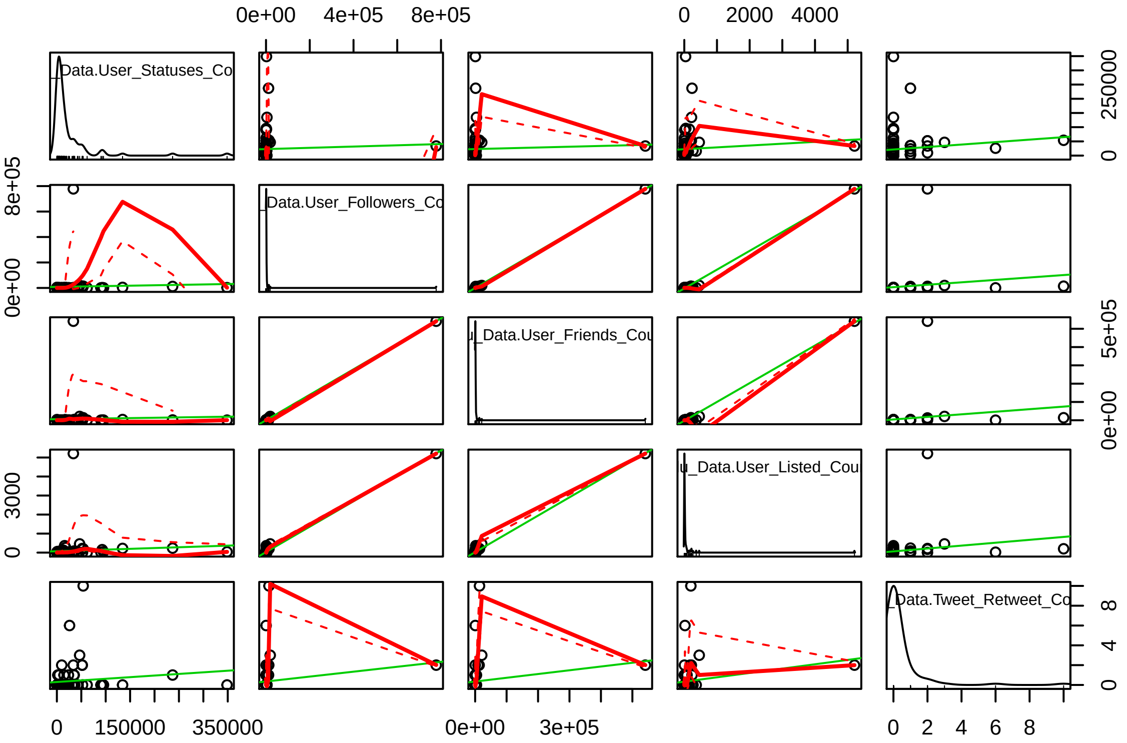
```
hist(Flu_Data$Favorite_Outcome, main='Favorite_Outcome bins=1', xlab='Favorite_Outcome',
breaks=50)
```



Finding relationship among variables

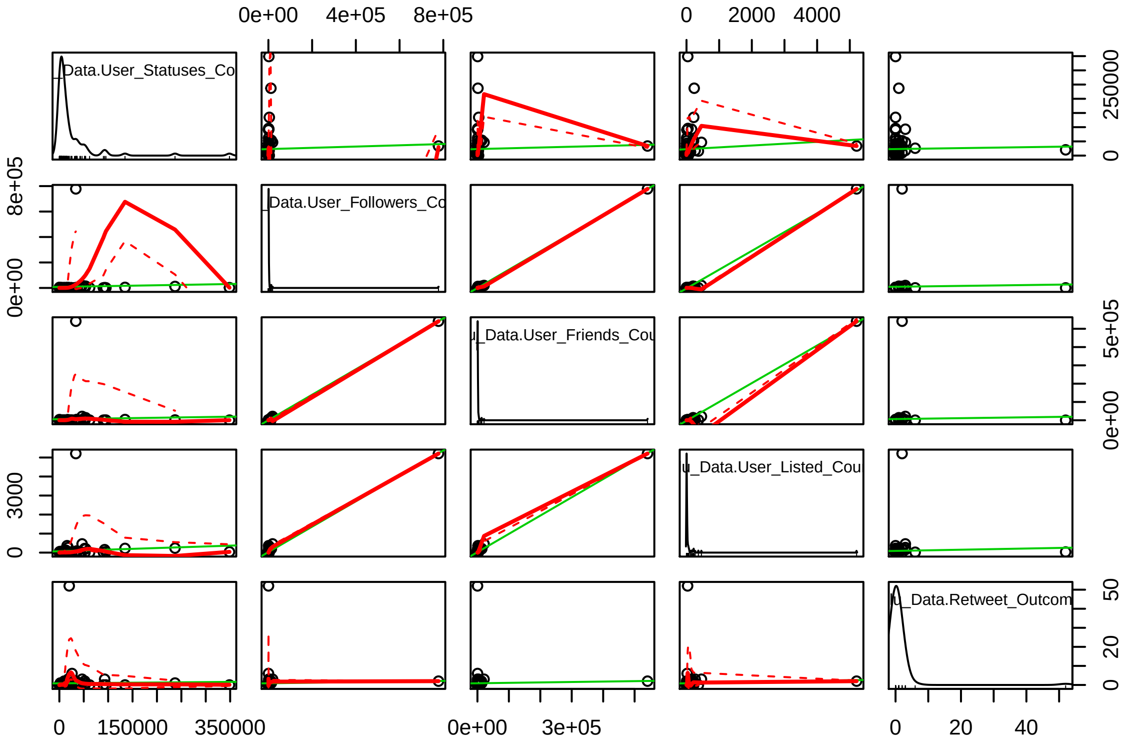
a. Retweet Measure before experiment

```
scatterplotMatrix(~ Flu_Data$User_Statuses_Count + Flu_Data$User_Followers_Count +
                    Flu_Data$User_Friends_Count + Flu_Data$User_Listed_Count +
                    Flu_Data$Tweet_Retweet_Count, data = Flu_Data)
```



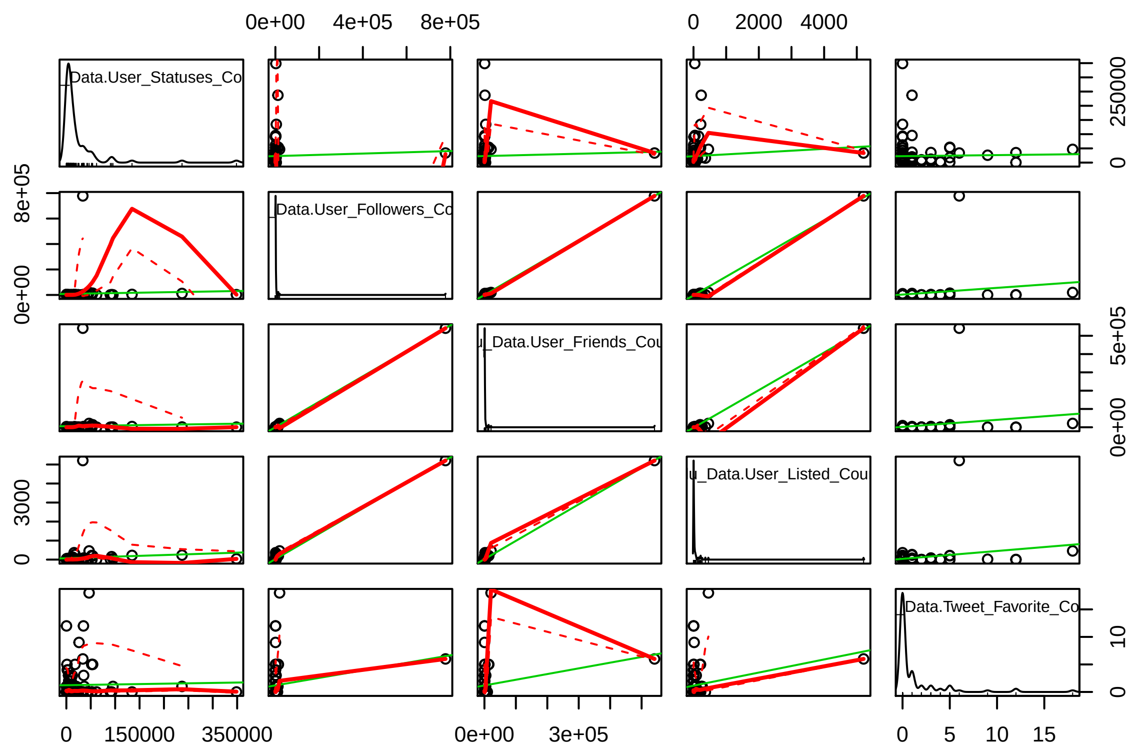
b. Retweet Measure after experiment

```
scatterplotMatrix(~ Flu_Data$User_Statuses_Count + Flu_Data$User_Followers_Count +
                    Flu_Data$User_Friends_Count + Flu_Data$User_Listed_Count +
                    Flu_Data$Retweet_Outcome, data = Flu_Data)
```



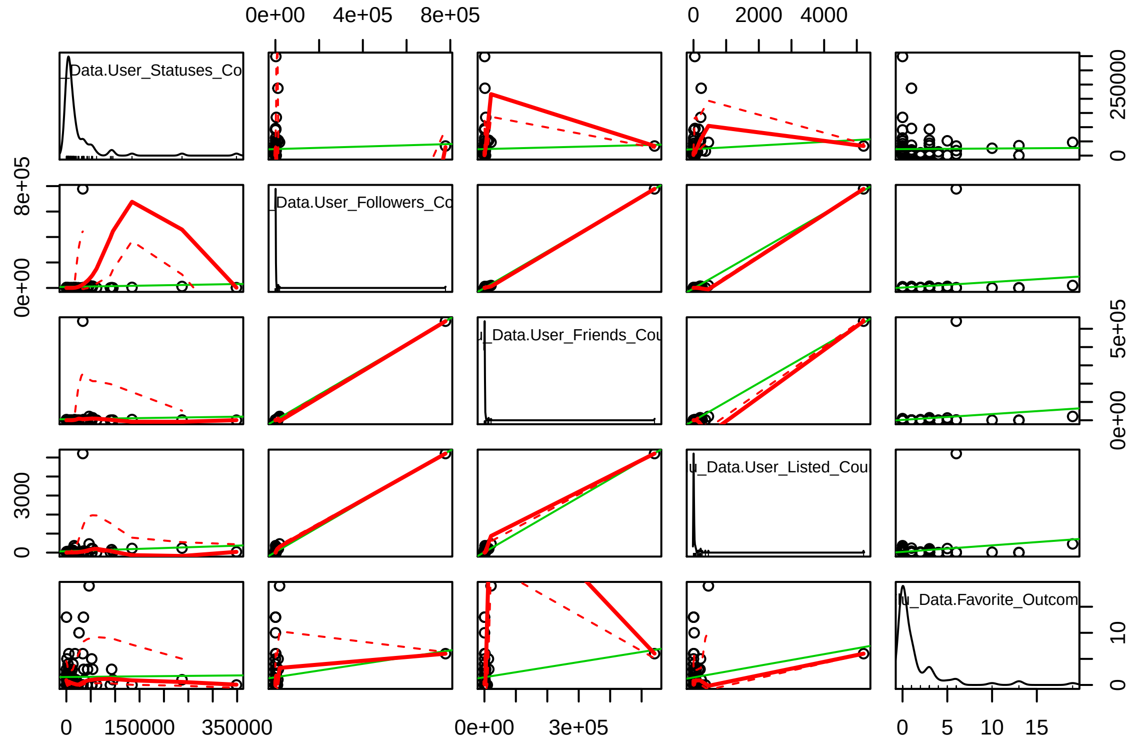
c. Favorite Measure before experiment

```r
scatterplotMatrix(~ Flu_Data$User_Statuses_Count + Flu_Data$User_Followers_Count +
                    Flu_Data$User_Friends_Count + Flu_Data$User_Listed_Count +
                    Flu_Data$Tweet_Favorite_Count, data = Flu_Data)
```



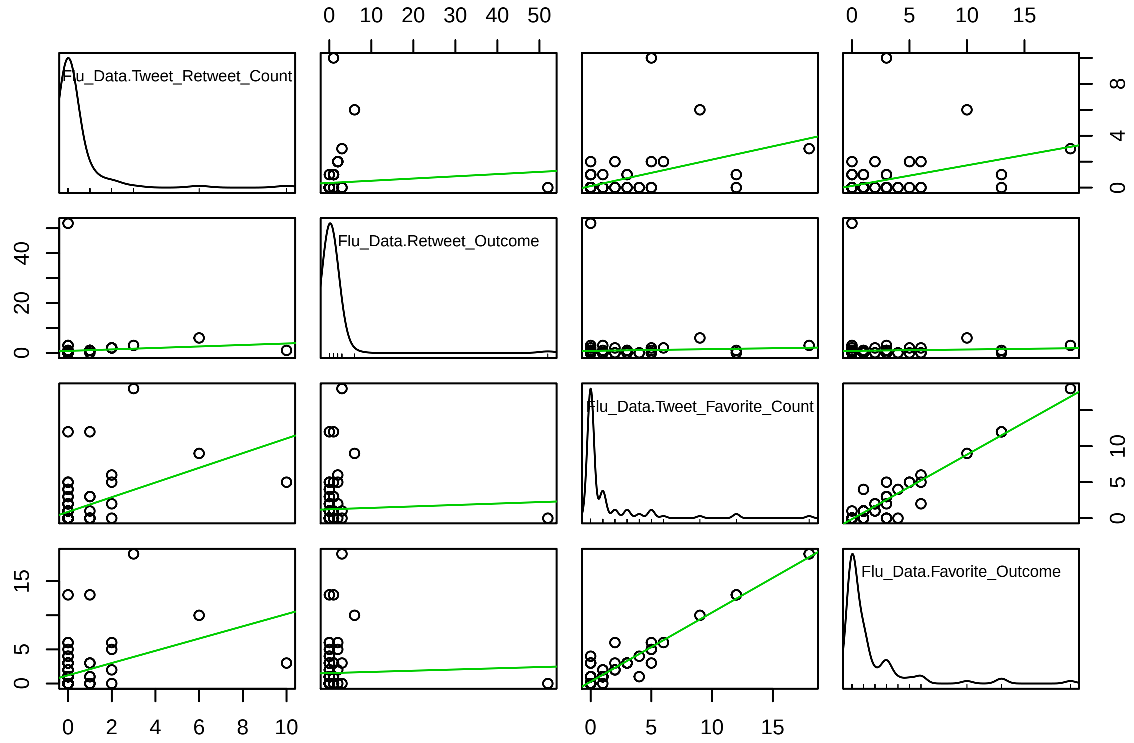
d. Favorite Measure after experiment

```
scatterplotMatrix(~ Flu_Data$User_Statuses_Count + Flu_Data$User_Followers_Count +
                    Flu_Data$User_Friends_Count + Flu_Data$User_Listed_Count +
                    Flu_Data$Favorite_Outcome, data = Flu_Data)
```



e. Output measures relationship before and after experiment

```
scatterplotMatrix(~ Flu_Data$Tweet_Retweet_Count + Flu_Data$Retweet_Outcome +
                    Flu_Data$Tweet_Favorite_Count + Flu_Data$Favorite_Outcome, data = Flu_Data)
```



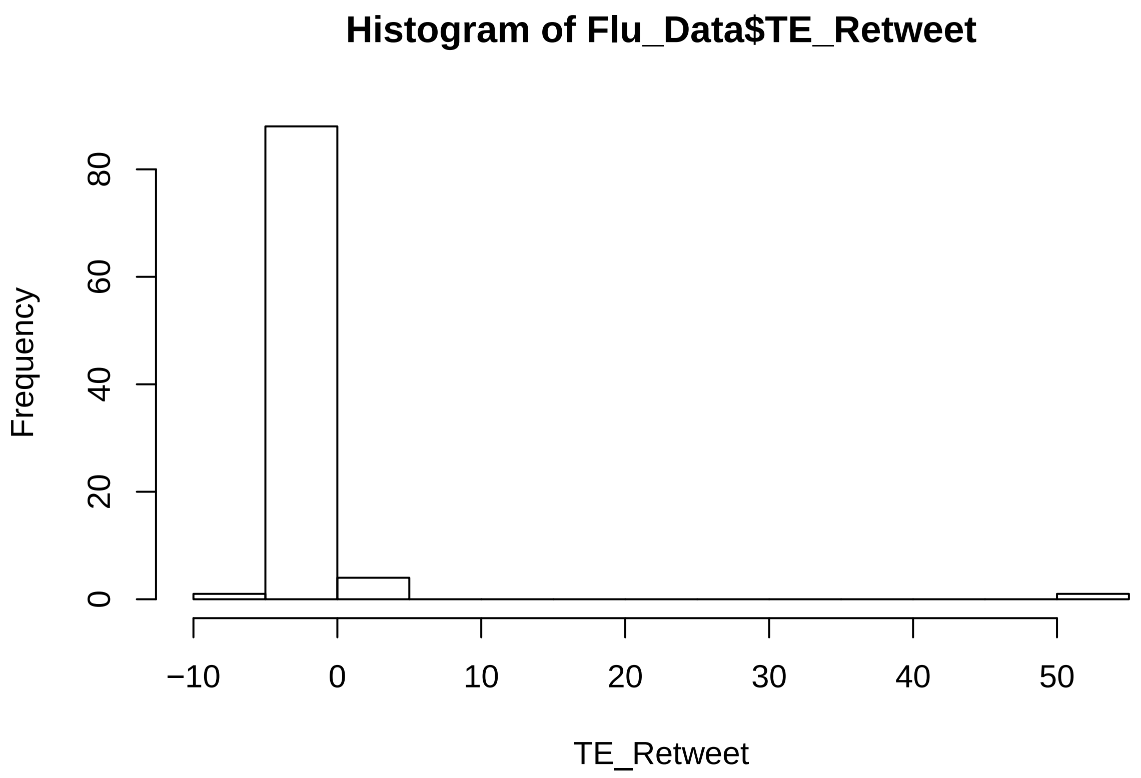
EDA Conclusions

Collected variables with numerical values have two characteristics: first, their distribution is right-skewed; second, the outliers are far away from majority of data. For such abnormal outliers, we suspect these data are contributed by Tweet bots. Among variables User-Statuses-Count, User-Followers-Count, User-Friends-Count, User-Listed-Count, User-Followers-Count has most significant linear relationship with respect to Favorite-Outcome variable. But this linear trend is made by outliers, hence, this may not be a good strategy to utilize this variable (User-Followers-Count) as our experiment measures although this number is easier to read from each Tweet users webpages. By comparing data before and after experiments for our two output measures, Favorite-Outcome variable has much better trend with Tweet-Favorite-Count variable compared to Retweet-outcome and Tweet-Retweet-Count variable.

1. General ATE

We are interested in how is treatment effect for retweet number and favoriate number. At first, we observe the individual treatment effect for each subject by retweet number and favoriate number.

```
Flu_Data <- read.csv("W241_Final_Project_Data_v1.csv", header=TRUE)
#Flu_Data_CDC <- Flu_Data[Flu_Data$Assign_Ind == 1, ]
#Flu_Data_CDC$TE_Retweet <- Flu_Data_CDC$Retweet_Outcome - Flu_Data_CDC$Tweet_Retweet_Count
#hist(Flu_Data_CDC$TE_Retweet, freq = TRUE, breaks = 10, xlab = "TE_Retweet")
#summary(Flu_Data_CDC$TE_Retweet)
Flu_Data$TE_Retweet <- Flu_Data$Retweet_Outcome - Flu_Data$Tweet_Retweet_Count
hist(Flu_Data$TE_Retweet, freq = TRUE, breaks = 10, xlab = "TE_Retweet")
```

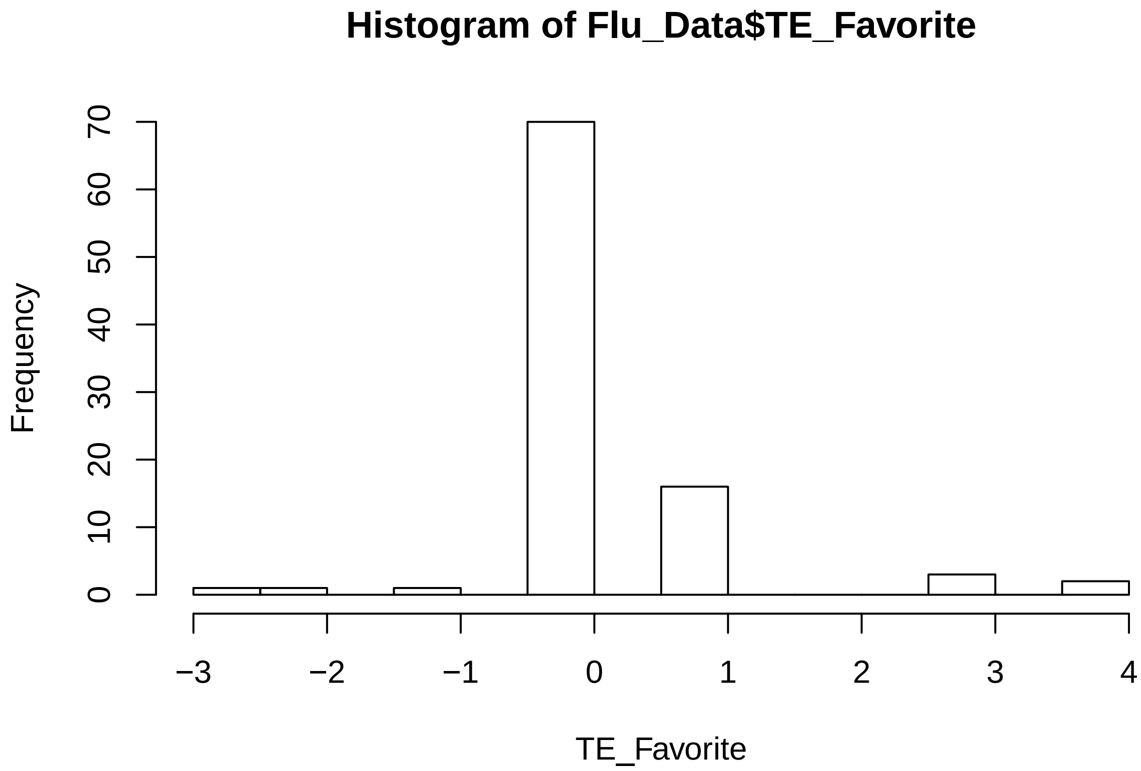


```
summary(Flu_Data$TE_Retweet)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -9.0000  0.0000  0.0000  0.5319  0.0000 52.0000
```

The treatment effect for these subjects measured by favoriate retweet number is **0, 0, -1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, -9, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 3, 0, 52, 0, 1, 0**.

```
#Flu_Data_CDC$TE_Favorite <- Flu_Data_CDC$Favorite_Outcome - Flu_Data_CDC$Tweet_Favorite_Count
#hist(Flu_Data_CDC$TE_Favorite, freq = TRUE, breaks = 10, xlab = "TE_Favorite")
#summary(Flu_Data_CDC$TE_Favorite)
Flu_Data$TE_Favorite <- Flu_Data$Favorite_Outcome - Flu_Data$Tweet_Favorite_Count
hist(Flu_Data$TE_Favorite, freq = TRUE, breaks = 10, xlab = "TE_Favorite")
```



```
summary(Flu_Data$TE_Favorite)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -3.0000  0.0000  0.0000  0.2872  0.0000  4.0000
```

So treatment effect for these subjects measured by favorite number is **1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 3, 0, 0, 0, 0, 1, 4, 0, 0, 0, 0, 0, 0, 0, 0, 1, 3, 1, -2, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, -1, 0, 3, -3, 0, 4, 0, 0**.

Accoding to primary histogram shown, for retweet measure, about 90 % of the subject have a treatment effect equal to zero, i.e., their treated outcome makes no difference compared to their untreated outcome, five of them have a treated outcome larger than her/his untreated outcome, and another two have a treated outcome smaller than her/his untreated outcome. We have to note that there is an outlier for retweet number. This is quite possible generated by tweet bot and we may remove this for later consideration. For favoriate measure of effect, about 70 % of the subject have a treatment effect equal to zero, i.e., their treated outcome makes no difference compared to their untreated outcome, however, there are about 25 % of subjects to have positive favoriate response after treatment of flu related links.

For the population being treated with links, the true average treatment effect (ATE) of them can be obtained as

```
ATE_Retweet <- mean(Flu_Data$TE_Retweet)

ATE_Favorite <- mean(Flu_Data$TE_Favorite)
```

The average treatment effect measured by Retweet is **0.5319149**, and the average treatment effect measured by Favorite is **0.287234** .

Since we can control exactly what websites to reply about correcting flu misunderstanding, we happen to randomly assign the subject to treatment website (CDC link) and control website for each pair of subjects. The ATE for such assignment can be evluated as

```
Flu_Data_CDC <- Flu_Data[Flu_Data$Assign_Ind == 1, ]
est_ATE_Retweet <- mean(Flu_Data_CDC$Retweet_Outcome, na.rm = TRUE) - mean(Flu_Data_CDC$Tweet_Retweet_C

est_ATE_Favorite <- mean(Flu_Data_CDC$Favorite_Outcome, na.rm = TRUE) - mean(Flu_Data_CDC$Tweet_Favorit
```

The average treatment effect measured by Retweet for block assignment is **0**, and the average treatment effect measured by Favorite is **0.4651163**. For Retweet measure, the difference for Retweet is -0.5319149, and for Favorite measure, the difference is 0.1778822. The reason for such difference is that we have not used D_i but d_i, i.e., the single realization of the experiment. Had we tried all possible combinations of treatment allocation, we would have obtained the real value of the ATE. The difference in means of the previous observational study shows that initial Retweet and Favorite are different from these two group of subjects. If we apply t-test by R, such difference is not statistically significant at the 0.05 level for Retweet measure, same for Favorite measure.

```
t.test(Flu_Data_CDC$Tweet_Retweet_Count, Flu_Data_CDC$Retweet_Outcome, alternative = "two.sided")
```

```
##
##  Welch Two Sample t-test
##
## data:  Flu_Data_CDC$Tweet_Retweet_Count and Flu_Data_CDC$Retweet_Outcome
## t = 0, df = 84, p-value = 1
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.4692446  0.4692446
## sample estimates:
## mean of x mean of y
## 0.3953488 0.3953488
```

```
t.test(Flu_Data_CDC$Tweet_Favorite_Count, Flu_Data_CDC$Favorite_Outcome, alternative = "two.sided")
```

```
##
##  Welch Two Sample t-test
##
## data:  Flu_Data_CDC$Tweet_Favorite_Count and Flu_Data_CDC$Favorite_Outcome
## t = -0.58794, df = 83.612, p-value = 0.5582
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -2.038389  1.108157
## sample estimates:
## mean of x mean of y
##  1.441860  1.906977
```

2. Location Effect

We will adopt the output measure as favorite count here, the same approach can be applied to Retweet measure. We are interested about causal effect of different geographical area. The global region is classified into two regions: tweets from U.S. and tweets from non-U.S. We want to explore the treatment effect at different regions.

```r
library(foreign)
library(data.table)
library(stargazer)
# US = 1, Non-US = 0
# CDC = 1, Non-CDC = 0
# CDC, Favorite_Outcome, At_US_or_not
Flu_US <- read.csv("W241_Final_Project_Data_Flu_US.csv", header=TRUE)
head(Flu_US)
```

```
##   CDC Favorite_Outcome At_US_or_not
## 1   1                2            0
## 2   0                3            0
## 3   1                0            0
## 4   0                0            1
## 5   0                0            1
## 6   1                0            0
```

```r
Flu_US_t <- data.table(Flu_US)
```

We begin by estimating the effect of having a CDC reply link for favorite outcome with respect to US and non US area.

```r
estimation.ate <- function(outcomes, treatment) {
  # estimates ates
  mean(outcomes[treatment==1]) - mean(outcomes[treatment==0])
  }
(ATE_US <- estimation.ate(Flu_US$Favorite_Outcome[Flu_US$At_US_or_not == 1],
                                        Flu_US$CDC[Flu_US$At_US_or_not == 1]))
```

```
## [1] 2.297619
```

```r
(ATE_Non_US <- estimation.ate(Flu_US$Favorite_Outcome[Flu_US$At_US_or_not == 0],
                                        Flu_US$CDC[Flu_US$At_US_or_not == 0]))
```

```
## [1] -0.8484848
```

For US, the effect of treatment is **2.297619**, and for Non-US region, the effect of treatment is **-0.8484848**. The standard errors of the estimated ATE at US and Non-US area are

```r
(SE_US <-
    sqrt(var(Flu_US$Favorite_Outcome[Flu_US$At_US_or_not == 1 & Flu_US$CDC == 0]) /
            length(Flu_US$Favorite_Outcome[Flu_US$At_US_or_not == 1 & Flu_US$CDC == 0]) +
            var(Flu_US$Favorite_Outcome[Flu_US$At_US_or_not == 1 & Flu_US$CDC == 1]) /
            length(Flu_US$Favorite_Outcome[Flu_US$At_US_or_not == 1 & Flu_US$CDC == 1])))
```

```
## [1] 1.122915
```

```
(SE_Non_US <-
    sqrt(var(Flu_US$Favorite_Outcome[Flu_US$At_US_or_not == 0 & Flu_US$CDC == 0]) /
            length(Flu_US$Favorite_Outcome[Flu_US$At_US_or_not == 0 &
                                            Flu_US$CDC == 0]) +
            var(Flu_US$Favorite_Outcome[Flu_US$At_US_or_not == 0 & Flu_US$CDC == 1]) /
            length(Flu_US$Favorite_Outcome[Flu_US$At_US_or_not == 0 &
                                            Flu_US$CDC == 1])))
```

```
## [1] 0.6445853
```

For US area, the standard error is **1.1229147**, and for Non-US area, the standard error is **0.6445853**.

```
(SE_ALL <- sqrt(SE_US^2 *
                            dim(Flu_US[Flu_US$At_US_or_not == 1, ])[1]^2 +
                            SE_Non_US^2 *
                            dim(Flu_US[Flu_US$At_US_or_not == 0, ])[1]^2) /
    dim(Flu_US)[1])
```

```
## [1] 0.6339381
```

The overall Standard Error is **0.6339381**.

From equation (3.10) of FE textbook, the overall ATE for both areas combined is

```
(ATE_ALL <- (dim(Flu_US[Flu_US$At_US_or_not == 1, ])[1] *
                        ATE_US +
                        dim(Flu_US[Flu_US$At_US_or_not == 0, ])[1] *
                        ATE_Non_US) / dim(Flu_US)[1])
```

```
## [1] 0.6576287
```

If we apply pooling method, we have

```
(ATE_pooled <- estimation.ate(Flu_US$Favorite_Outcome, Flu_US$CDC))
```

```
## [1] 0.6716826
```

We have to note that ATE value calculated from pool **0.6716826** is different from ATE value obtained from combined one **0.6576287**. This is a biased estimate because the probability of being assigned to the treatment group varies by block (area): in US this probability is $17/45 = 37.8\%$, while in Non-US the probability of being assigned to the treatment group is $26/49 = 53.1\%$. Besides, the number of Favorites is lower on average in Non-US, so the overall treatment effect calculated this way is larger than it actually is. Therefore, if outcomes were higher in the treatment group, it might reflect differences between US and Non-US rather than a treatment effect.

The treatment effect is not statistically significant in US, Non-US, and global area by using randomization inference to test the sharp null hypothes.

```
distribSNH_US <-
    replicate(10e3, estimation.ate(Flu_US$Favorite_Outcome[Flu_US$At_US_or_not == 1],
                                    sample(Flu_US$CDC[Flu_US$At_US_or_not == 1])))
(pvalue_US <- mean(ATE_US >= distribSNH_US))
```

```
## [1] 0.9888
```

```
distribSNH_Non_US <-
    replicate(10e3, estimation.ate(Flu_US$Favorite_Outcome[Flu_US$At_US_or_not == 0],
                                   sample(Flu_US$CDC[Flu_US$At_US_or_not == 0])))
(pvalue_Non_US <- mean(ATE_Non_US >=
                                               distribSNH_Non_US))
```

```
## [1] 0.1325
```

```
distribSNH_ALL <-
    replicate(10e3, estimation.ate(Flu_US$Favorite_Outcome,
                                   c(sample(Flu_US$CDC[Flu_US$At_US_or_not == 1]),
                                   sample(Flu_US$CDC[Flu_US$At_US_or_not == 0]))))
(pvalue_senator <- mean(ATE_ALL >= distribSNH_ALL))
```

```
## [1] 0.8417
```

```
df_distribSNH_ALL <-
    rbind(data.frame(ATE = distribSNH_US,
                          state = rep("US" , 10e3),
                          ATE_estimate = rep(ATE_US, 10e3)),
            data.frame(ATE = distribSNH_Non_US,
                          state = rep("Non-US", 10e3),
                          ATE_estimate = rep(ATE_Non_US, 10e3)),
            data.frame(ATE = distribSNH_ALL,
                          state = rep("both", 10e3),
                          ATE_estimate = rep(ATE_ALL, 10e3)))
library(ggplot2)
density <- ggplot(df_distribSNH_ALL, aes(ATE))
density + geom_density() + facet_grid( ~ state) +
    geom_vline(aes(xintercept = ATE_estimate), color = "red") +
    theme(legend.position = "none")
```

By ploting histograms for both the treatment and control groups in each area, all these data are distributed with right-skewness.

```
par(mfrow = c(2,2))
Flu_US_t [At_US_or_not==0 & CDC==0, hist(Favorite_Outcome, main = "US Control")]
Flu_US_t [At_US_or_not==0 & CDC==1, hist(Favorite_Outcome, main = "US Treatment")]
Flu_US_t [At_US_or_not==1 & CDC==0, hist(Favorite_Outcome, main = "Non-US Control")]
Flu_US_t [At_US_or_not==1 & CDC==0, hist(Favorite_Outcome, main = "Non-US Treat")]
```

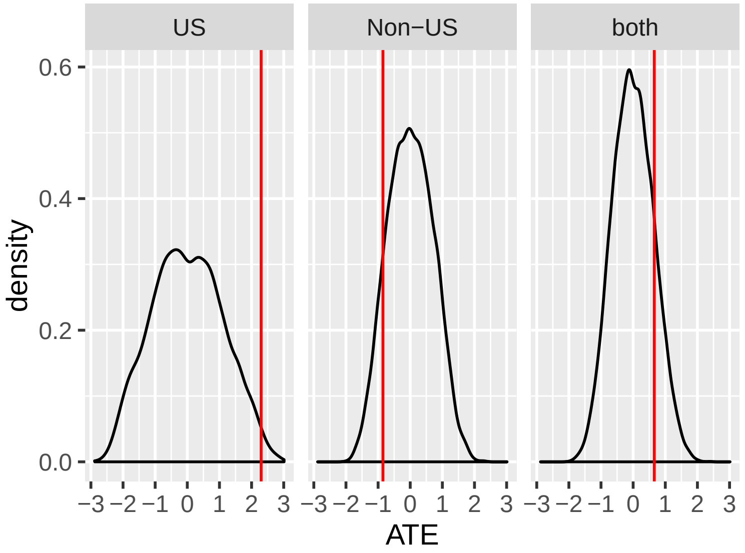
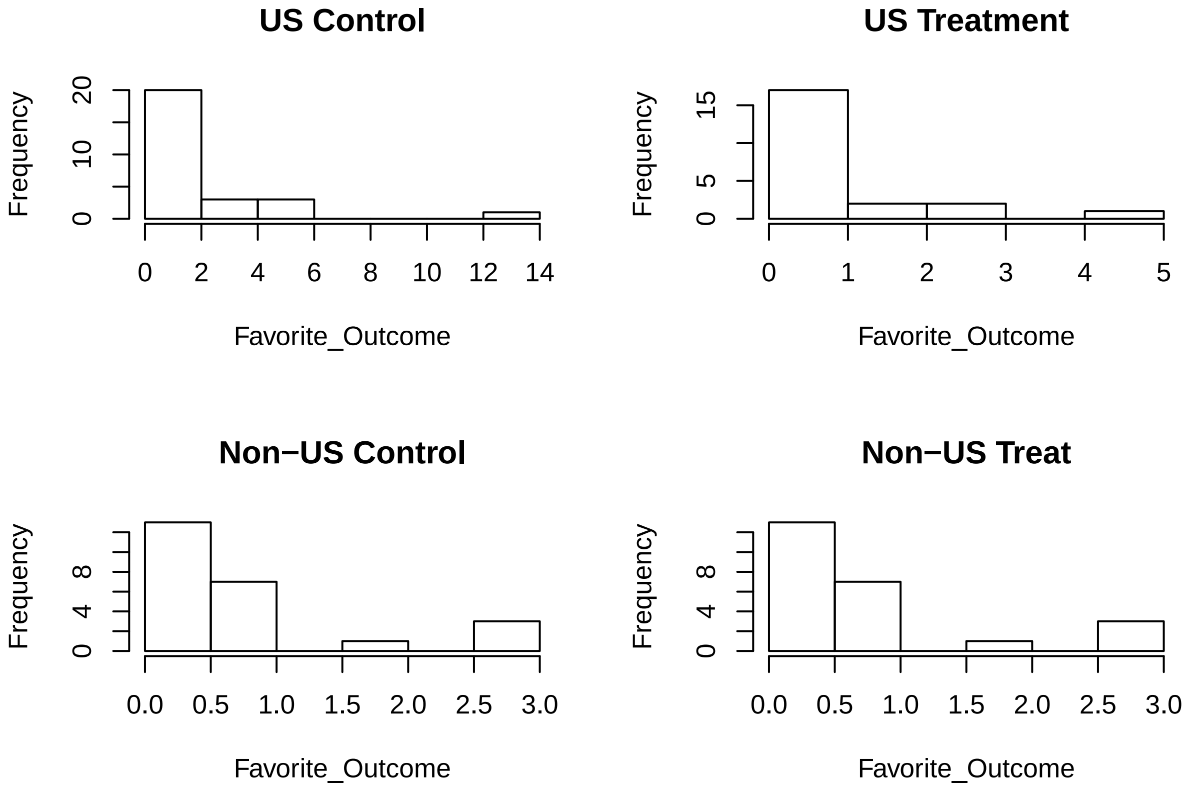


Figure 3: Senator: Sampling distribution of the ATE under the SNH in each state



3. Relationship of output measures: Retweet v.s. Favorite and Sex Causal Effect

In this part of investigation, we want to understand the effect of uding different output measures in causal analysis. We begin by understanding the causal effect of CDC link on whether some Tweet is retweet after one week? **

```
library(foreign)
library(pastecs)
# Function to calculate confidence interval
CI <- function(coefficients){
CI.lower = as.numeric(round(coefficients[1] - 1.96 * coefficients[2], 4))
CI.upper = as.numeric(round(coefficients[1] + 1.96 * coefficients[2], 4))
return(list(
ate.point.estimate = round(coefficients[1], 4),
CI = paste0("[",CI.lower,",", CI.upper,"]"),
p.value = round(coefficients[4], 4)))
}

# A function to calculate RSEs
RSEs <- function(model){
    require(sandwich, quietly = TRUE)
    require(lmtest, quietly = TRUE)
    newSE <- vcovHC(model)
    coeftest(model, newSE)
    }
Flu_Data <- read.csv("W241_Final_Project_Data_v1.csv", header=TRUE)
#head(Flu_Data)

model_5a <- lm(Retweet_Outcome ~ Assign_Ind, data = Flu_Data)
summary(model_5a)
```

```
##
## Call:
## lm(formula = Retweet_Outcome ~ Assign_Ind, data = Flu_Data)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -1.333 -1.333 -0.395 -0.395 50.667
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    1.333      0.758   1.759   0.0819 .
## Assign_Ind    -0.938      1.121  -0.837   0.4048
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.413 on 92 degrees of freedom
## Multiple R-squared:  0.007557,   Adjusted R-squared:  -0.00323
## F-statistic: 0.7005 on 1 and 92 DF,  p-value: 0.4048
```

```
coef_model_5a <- summary(model_5a)$coefficients[2, ]
#print(coef_model_5a)
results_5a <- CI(coef_model_5a)
print(results_5a)
```

```
## $ate.point.estimate
## Estimate
##   -0.938
##
## $CI
## [1] "[-3.1345,1.2585]"
##
## $p.value
## Pr(>|t|)
##   0.4048
```

the estimated effect of CDC treatment is -0.938, with onfidence interval [-3.1345,1.2585] and p-value 0.4048. This model is not very good by having such large p-value, hence, we are interested in adding other covariates by having retweet and favorite count on day 0 (before experiment) to the regression.

```
# Function to calculate confidence interval
CI <- function(coefficients){
CI.lower = as.numeric(round(coefficients[1] - 1.96 * coefficients[2], 4))
CI.upper = as.numeric(round(coefficients[1] + 1.96 * coefficients[2], 4))
return(list(
ate.point.estimate = round(coefficients[1], 4),
CI = paste0("[",CI.lower,",", CI.upper,"]"),
p.value = round(coefficients[4], 4)))
}


model_5b <- update(model_5a, . ~ . + Tweet_Retweet_Count + Tweet_Favorite_Count)
summary(model_5b)
```

```
##
## Call:
## lm(formula = Retweet_Outcome ~ Assign_Ind + Tweet_Retweet_Count +
##     Tweet_Favorite_Count, data = Flu_Data)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -3.126 -1.210 -0.416 -0.252 50.790
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)           1.20983    0.79648   1.519    0.132
## Assign_Ind           -0.95755    1.13207  -0.846    0.400
## Tweet_Retweet_Count   0.28049    0.49128   0.571    0.569
## Tweet_Favorite_Count  0.02232    0.22034   0.101    0.920
##
## Residual standard error: 5.458 on 90 degrees of freedom
## Multiple R-squared:  0.01295,    Adjusted R-squared:  -0.01995
## F-statistic: 0.3935 on 3 and 90 DF,  p-value: 0.758
```

```
coef_model_5b <- summary(model_5b)$coefficients[2, ]
results_5b <- CI(coef_model_5b)
print(results_5b)
```

```
## $ate.point.estimate
## Estimate
##  -0.9576
##
## $CI
## [1] "[-3.1764,1.2613]"
```

```
##
## $p.value
## Pr(>|t|)
##   0.3999
```

the estimated effect of CDC treatment is -0.9576, with confidence interval [-3.1764,1.2613] and p-value 0.3999. By comparing these two models (part(a) and part(b)), it is better to adopt part (b)—with higher p-value, so we are more certain about the treatment effect (stronger statistical significance).

If we switch from the outcome of Retweet to the outcome of Favorite, and use the same regression covariates as in part (b), we have

```
model_5f <- lm(Favorite_Outcome ~ Assign_Ind + Tweet_Retweet_Count +
                        Tweet_Favorite_Count, data = Flu_Data)
summary(model_5f)
```

```
##
## Call:
## lm(formula = Favorite_Outcome ~ Assign_Ind + Tweet_Retweet_Count +
##     Tweet_Favorite_Count, data = Flu_Data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.6503 -0.4665 -0.1472 -0.0891  3.5335
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)           0.14716    0.13325   1.104   0.2724
## Assign_Ind            0.31931    0.18939   1.686   0.0953 .
## Tweet_Retweet_Count  -0.17106    0.08219  -2.081   0.0402 *
## Tweet_Favorite_Count  1.04597    0.03686  28.375   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9131 on 90 degrees of freedom
## Multiple R-squared:  0.9149, Adjusted R-squared:  0.912
## F-statistic: 322.4 on 3 and 90 DF,  p-value: < 2.2e-16
```

```
coef_model_5f <- summary(model_5f)$coefficients[2, ]
results_5f <- CI(coef_model_5f)
print(results_5f)
```

```
## $ate.point.estimate
## Estimate
##   0.3193
##
## $CI
## [1] "[-0.0519,0.6905]"
## 
## $p.value
## Pr(>|t|) 
##   0.0953
```

The estimated effect of CDC treatment is 0.3193, with confidence interval [-0.0519,0.6905] and p-value 0.0953. We can see that the causal effect becomes positive, moreover, this meausre has much smaller confidence interval and p-value, i.e., much reliable causal measures. If we consider Sex effect, we have

```
model_5f_sex <- lm(Favorite_Outcome ~ Assign_Ind*Sex + Tweet_Retweet_Count +
                        Tweet_Favorite_Count, data = Flu_Data)
summary(model_5f_sex)
```

```
## 
## Call:
## lm(formula = Favorite_Outcome ~ Assign_Ind * Sex + Tweet_Retweet_Count + 
##     Tweet_Favorite_Count, data = Flu_Data)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -3.6388 -0.4472 -0.2349 -0.0445  3.5528 
## 
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)    
## (Intercept)             0.04451    0.19673   0.226   0.8215    
## Assign_Ind              0.40270    0.26941   1.495   0.1386    
## Sex                     0.19043    0.26295   0.724   0.4708    
## Tweet_Retweet_Count    -0.18248    0.08435  -2.163   0.0332 *  
## Tweet_Favorite_Count    1.04790    0.03771  27.785   <2e-16 ***
## Assign_Ind:Sex         -0.14295    0.38917  -0.367   0.7143    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.9205 on 88 degrees of freedom
## Multiple R-squared:  0.9154, Adjusted R-squared:  0.9106 
## F-statistic: 190.4 on 5 and 88 DF,  p-value: < 2.2e-16
```

```
coef_model_5f_sex <- summary(model_5f_sex)$coefficients[2, ]
results_5f_sex <- CI(coef_model_5f_sex)
print(results_5f_sex)
```

```
## $ate.point.estimate
## Estimate 
##   0.4027
```

```
##
## $CI
## [1] "[-0.1253,0.9307]"
##
## $p.value
## Pr(>|t|)
##   0.1386
```

Now, the estimated effect of CDC treatment is 0.4027, with p-value 0.1386 and confidence interval [-0.1253,0.9307]. We have the results show that the treatment works better for Male. However, we found that the Tweets initial favorite count is a good predictor for output measured by favorite count (statistically significant).

Conclusion

There are some directions for us to improve further. First, we can use Amazon Mechanical Turk (AMT) to help us collect and filter Tweets. AMT is a crowdsourcing Internet marketplace enabling individuals and businesses to coordinate the use of human intelligence to perform tasks that computers are currently unable to do. We are then able to post jobs known as Human Intelligence Tasks (HITs), such as writing flu shot descriptions, or identifying performance on various flu medicines. Second, we can develope reply-bot to circumvent Twitter spam rule restrictions. Third, we may design experiments to test different value engagement strategies such as authority, objectivity, and empathy. Finally, the approaches adopted in this study can also be applied to study other public health outreach topics, e.g., childhood immunizations, antibiotic usage, and epidemics (Zika,Ebola, HIV, Malaria).

Appendix

In this Appendix, we will consider CACE effect for non-complicance treatment and apply Gibbs sampling to perform causal analysis based on predicted un-observable new Tweets.

CACE Effect

Suppose we have following table by considering one-sided noncomplpiance situation. This table is made by assume that 80% of our targets get treated actually. The response means that the number of output measures is increased.

Treat. Ass.	Treated	No.#	Response
Baseline	NO	20	35%
Treatment	Yes	33	52%
Treatment	NO	10	34%
Placebo	Yes	40	33%
Placebo	No	11	36%

```
df_5_11 <- data.frame(Assignment = c(rep("Baseline", 20),
                                     rep("Treatment", 43),
                                     rep("Placebo", 51)),
                                     # 0 for untreated, 1 for treated
```

```
                                Treated = c(rep(0, 20), rep(1, 33), rep(0, 10),
                                                rep(1, 40), rep(0, 11)),
                                            # 0 for not Response, 1 for Response
                                Response = c(rep(1, round(35/100*20)),
                                                rep(0, 20 - round(35/100*20)),
                                                rep(1, round(52/100*33)),
                                                rep(0, 33 - round(52/100*33)),
                                                rep(1, round(34/100*10)),
                                                rep(0, 10 - round(34/100*10)),
                                                rep(1, round(33/100*40)),
                                                rep(0, 40 - round(33/100*40)),
                                                rep(1, round(36/100*11)),
                                                rep(0, 11 - round(36/100*11))))
head(df_5_11)
```

```
##   Assignment Treated Response
## 1   Baseline       0        1
## 2   Baseline       0        1
## 3   Baseline       0        1
## 4   Baseline       0        1
## 5   Baseline       0        1
## 6   Baseline       0        1
```

Given current experimental design, we first want to know whether the proportion of compliers who get CDC link reply at the Treatment group is statistcally different from the proportion of compliers who get non-CDC link reply at the Placebo group.

```
library(data.table)
dt_5_11 <- as.data.table(df_5_11)
#print(dt_5_11, topn = 10)
ComplianceRate_Treatment <- dt_5_11[Assignment== "Treatment", mean(Treated)]
ComplianceRate_Placebo <- dt_5_11[Assignment== "Placebo", mean(Treated)]
```

We then apply t-test to examine the statistically significantly different from each other.

```
t.test(x = dt_5_11[Assignment == "Treatment", Treated], y = dt_5_11[Assignment == "Placebo", Treated])
```

```
##
##  Welch Two Sample t-test
##
## data:  dt_5_11[Assignment == "Treatment", Treated] and dt_5_11[Assignment == "Placebo", Treated]
## t = -0.19312, df = 88.421, p-value = 0.8473
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.1904807  0.1567370
## sample estimates:
## mean of x mean of y
## 0.7674419 0.7843137
```

T-test result shows that they are not different from one another since p-value is 0.8473.

Next, we wish to understand whether Never Takers in the treatment and placebo groups have the same rate of response.

```
NeverTakersRate_Treatment <- dt_5_11[Assignment== "Treatment" & Treated == 0, mean(Response)]
NeverTakersRate_Placebo <- dt_5_11[Assignment== "Placebo" & Treated == 0, mean(Response)]
```

Never Takers rates for the treatment is 0.3 and placebo is 0.3636364. We then apply t-test to examine the statistically significantly different from each other.

```
t.test(x = dt_5_11[Assignment == "Treatment" & Treated == 0 , Response], y = dt_5_11[Assignment == "Pla
```

```
##
##  Welch Two Sample t-test
##
## data:  dt_5_11[Assignment == "Treatment" & Treated == 0, Response] and dt_5_11[Assignment == "Placeb
## t = -0.29519, df = 18.939, p-value = 0.7711
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.5149448  0.3876721
## sample estimates:
## mean of x mean of y
## 0.3000000 0.3636364
```

The difference in the Never-Takers response rate between the treatment and placebo groups is not statistically significant since p = 0.7711. The data do not suggest that the never takers in the two different groups are attending at different rates. This is good because it helps us to know that there is no exclusion restriction violation as a part of being in the treatment.

Then, we estimate the CACE of receiving the placebo.

```
first_stage_lm <- lm(Treated ~ Assignment, dt_5_11[Assignment != "Treatment", ])
dt_5_11[Assignment != "Treatment", p := predict(first_stage_lm)]
second_stage_lm <- lm(Response ~ p, data = dt_5_11[Assignment != "Treatment", ])
summary(second_stage_lm)
```

```
##
## Call:
## lm(formula = Response ~ p, data = dt_5_11[Assignment != "Treatment",
##     ])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.3500 -0.3333 -0.3333  0.6500  0.6667
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.35000    0.10728   3.262  0.00172 **
## p           -0.02125    0.16139  -0.132  0.89563
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4798 on 69 degrees of freedom
## Multiple R-squared:  0.0002512,  Adjusted R-squared:  -0.01424
## F-statistic: 0.01734 on 1 and 69 DF,  p-value: 0.8956
```

The CACE of receiving the placebo is -0.021 (0.161). It is not statistically significant at all (p-value is 0.896), which is consistent with the assumption that the placebo has no effect on Response.

Next is to estimate the CACE of receiving the treatment. The first method is to use the conventional method of dividing the ITT by the ITT_{D}.

```
ITT_5_11d = dt_5_11[Assignment == "Treatment", mean(Response)] - dt_5_11[Assignment == "Baseline", mean

ITTD_5_11d = dt_5_11[Assignment== "Treatment", mean(Treated)]

#print (ITTD_5_11d)
```

The second method is by 2SLS.

```
first.stage <- lm(Treated ~ Assignment, dt_5_11[Assignment != "placebo", ])
dt_5_11 <- dt_5_11[Assignment != "placebo", p := predict(first.stage)]
second.stage <- lm(Response ~ p, data = dt_5_11[Assignment != "placebo", ])
summary(second.stage)
```

```
##
## Call:
## lm(formula = Response ~ p, data = dt_5_11[Assignment != "placebo",
##     ])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.3933 -0.3933 -0.3729  0.6067  0.6466
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.35336    0.10974   3.220  0.00168 **
## p            0.05091    0.15561   0.327  0.74413
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4909 on 112 degrees of freedom
## Multiple R-squared:  0.0009549,  Adjusted R-squared:  -0.007965
## F-statistic: 0.1071 on 1 and 112 DF,  p-value: 0.7441
```

The CACE is 0.050915

By comparing the response rates among the Compliers in both the treatment and placebo groups, we have

```
t.test(x = dt_5_11[Assignment == "Treatment" & Treated == 1, Response], y = dt_5_11[Assignment == "Plac
```

```
##
##  Welch Two Sample t-test
##
## data:  dt_5_11[Assignment == "Treatment" & Treated == 1, Response] and dt_5_11[Assignment == "Placeb
## t = 1.6408, df = 66.432, p-value = 0.1056
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.0412008  0.4215038
## sample estimates:
## mean of x mean of y
## 0.5151515 0.3250000
```

```
m <- lm(Response ~ Assignment, dt_5_11[Treated == 1 & Assignment != "baseline", ])
summary(m)
```

```
##
## Call:
## lm(formula = Response ~ Assignment, data = dt_5_11[Treated ==
##     1 & Assignment != "baseline", ])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.5151 -0.3250 -0.3250  0.4849  0.6750
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)          0.32500    0.07741   4.199 7.68e-05 ***
## AssignmentTreatment  0.19015    0.11513   1.652    0.103
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4896 on 71 degrees of freedom
## Multiple R-squared:  0.037,  Adjusted R-squared:  0.02344
## F-statistic: 2.728 on 1 and 71 DF,  p-value: 0.103
```

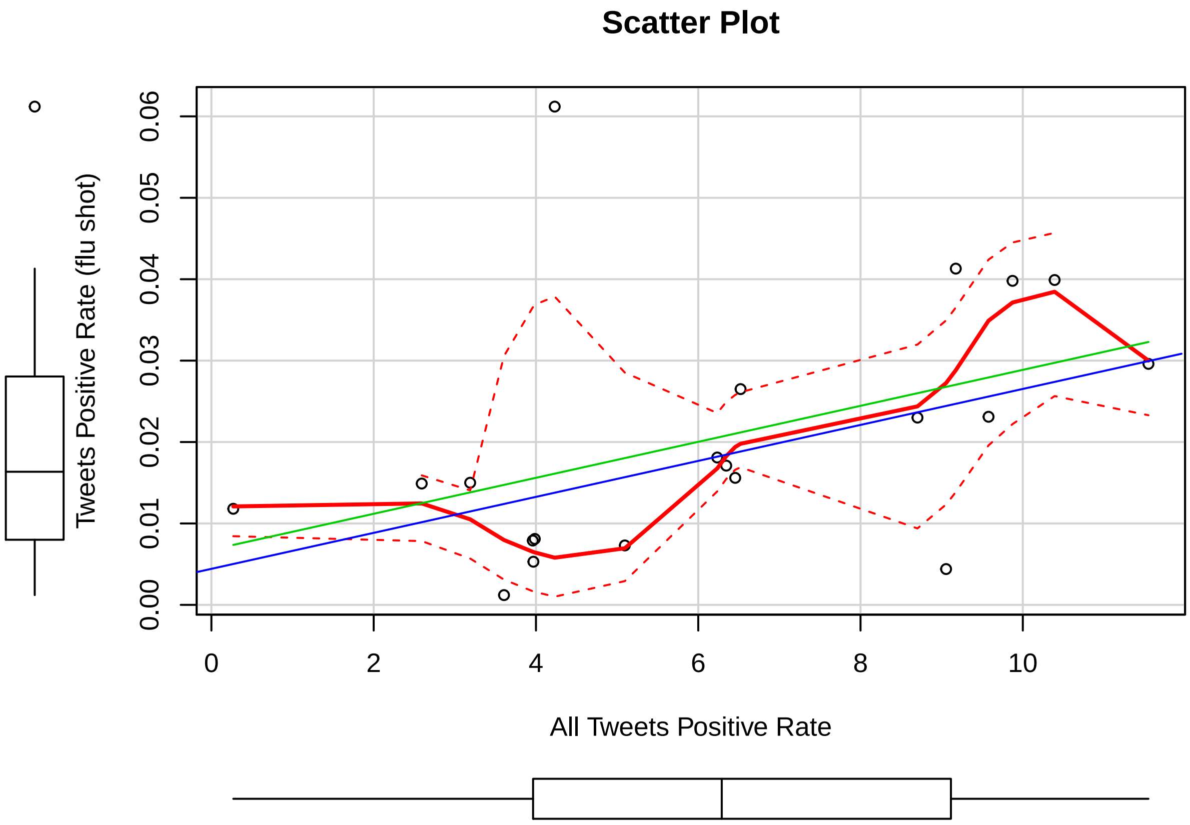
Hence, an estimated CACE is about 0.19. The reason why they are not equal to each other is that : (1) the proportion of Compliers based on subjects responses to the treatment or placebo may not exactly the same (though the difference is not statistically significant addressed in part (a) problem); (2) the Response rate among Never-Takers may not exactly the same among both groups (though, again, the difference is not statiscally significant addressed in part (b) problem).

Gibbs sampling for missing new Tweets

Comparing to output measures of Tweets like retweet count and favorite count, the new Tweets prepared by subjects can reflect more about what he/she accept or reject our CDC treatment. During our experimental procedure, however, it is not easy to collect such data due to time constraints or other natural restrictions. Different from textbook approach, we adopt Bayesian methods to provide us a natural framework for accounting for missing data without need to rely on ad hoc imputation. The basic idea is that we first collect other availble variables, the **tweets positive rate (ALL)** generated by a subject from his/her ALL previous Tweets averaged by days. Here positive Tweets mean those Tweets contain pre-defined physical and mental positive related words, e.g., strong, happy, healthy. Since we discover that there is a linear model between the **tweets positive rate (ALL)** and **tweets positive rate (Flu Shot)** (positive Tweets also containing key words: flu shot), then we can apply Gibbs sampling to determine such linear model coefficients. After we have these model coefficients, we can get missing **tweets positive rate (Flu Shot)** for each subjects based on his/her **tweets positive rate (ALL)**. Python coding implementation of geting all Tweets, calculating positive rates, and getting these model coefficients is given by (https://github.com/shihyuch/w241-Fianl-Project/blob/master/Gibbs%20Sampling%20for%20Missing%20Tweets_v3.ipynb).

We collect 20 subjects different from experimental subjects to build linear model for variables **tweets positive rate (ALL)** and **tweets positive rate (Flu Shot)**. The coefficient is not large, about 0.002, but it is almost statistical significan with p-value close to 0.05. This is acceptable because flu shot is not very popular topic with respect to all Tweets generated by each users.

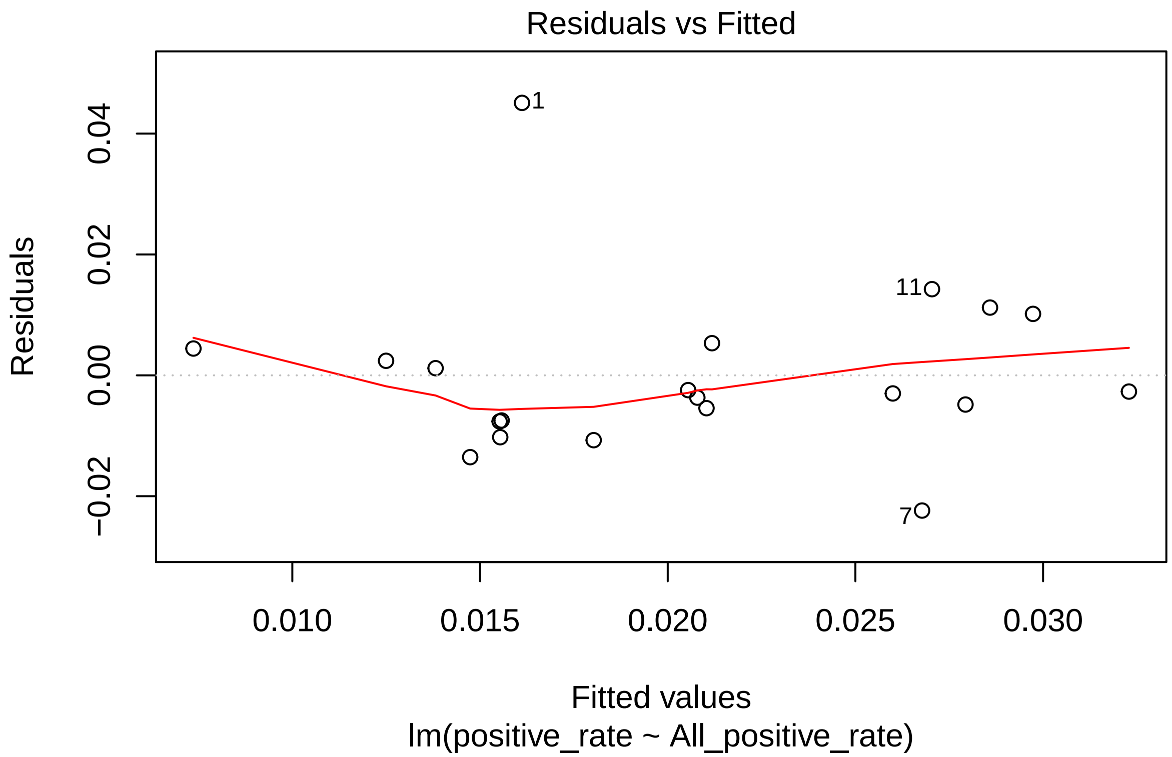
```
library(car)

# Positive Tweets also containing flu shot
positive_rate = c(0.0612, 0.0053, 0.0156,  0.0073, 0.0181,
                  0.0079, 0.0044, 0.0118, 0.0399, 0.023,
                  0.0413, 0.0012, 0.0265, 0.0398, 0.015,
                  0.0081, 0.0296, 0.0171, 0.0149, 0.0231)
All_positive_rate = c(4.2311, 3.9683, 6.4555, 5.0951, 6.234,
                      3.9607, 9.0545, 0.2693, 10.3923, 8.702,
                      9.1740, 3.6061, 6.5217, 9.8738, 3.189,
                      3.9854, 11.5486, 6.345, 2.592, 9.578)

tweets_Data <- data.frame(positive_rate, All_positive_rate)


scatterplot(positive_rate ~ All_positive_rate, data=tweets_Data,
   xlab="All Tweets Positive Rate", ylab="Tweets Positive Rate (flu shot)",
   main="Scatter Plot")

regression <- lm(positive_rate ~ All_positive_rate, data = tweets_Data)
abline(lm(positive_rate ~ All_positive_rate, data=tweets_Data), col="blue")
```

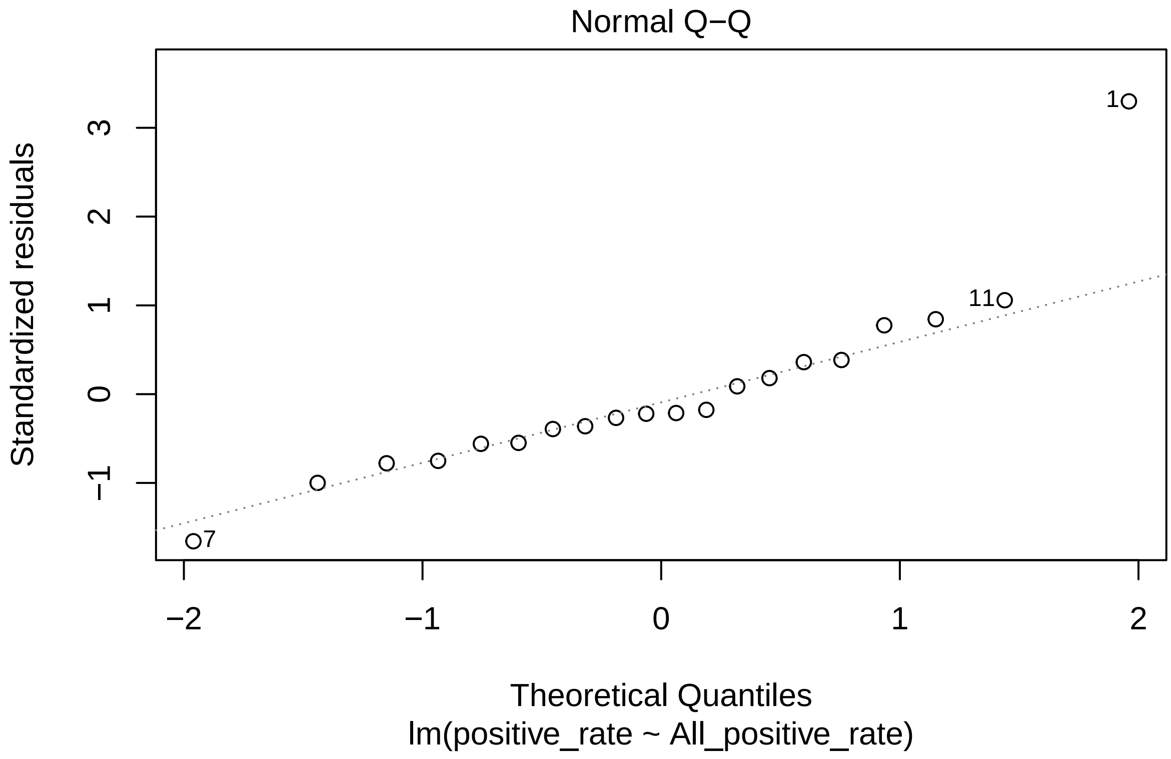


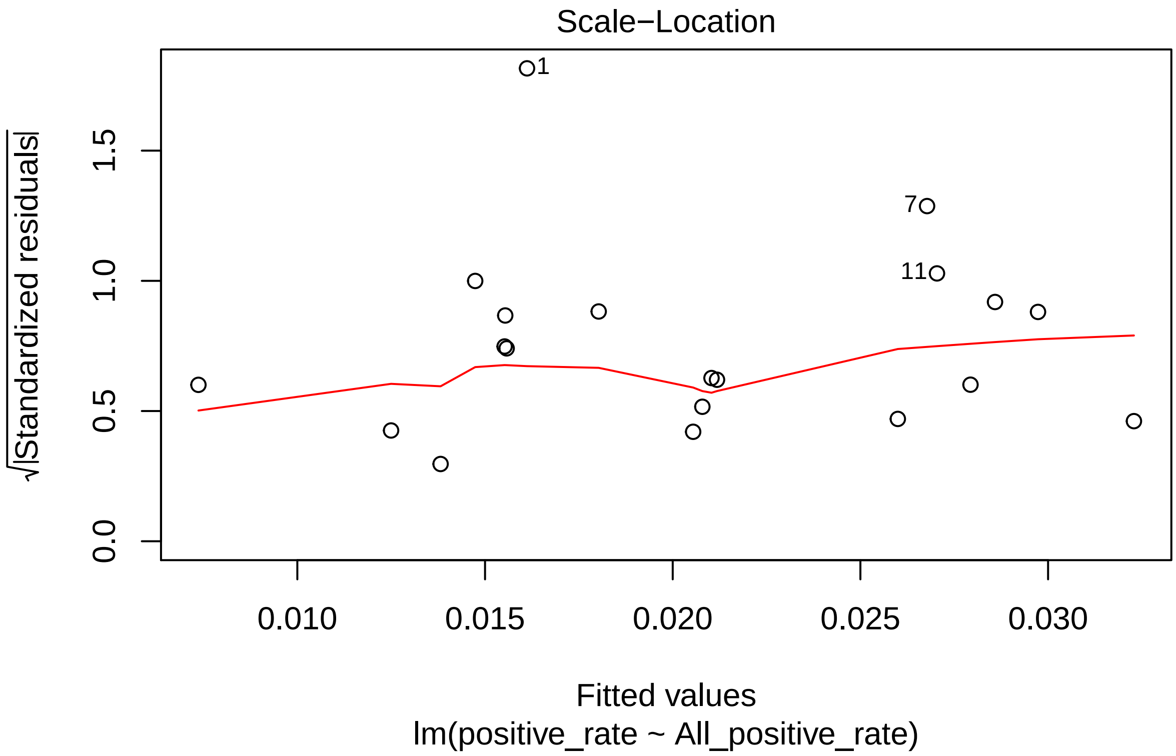
```r
summary(regression)
```

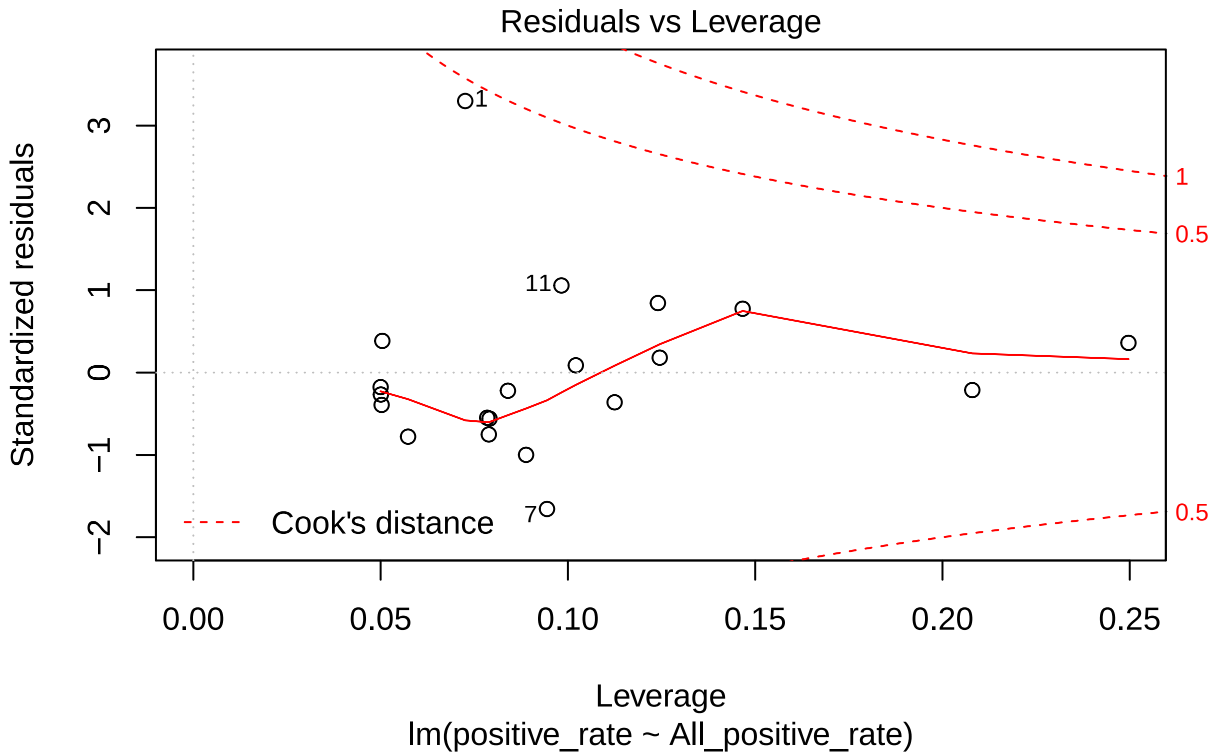
```
##
## Call:
## lm(formula = positive_rate ~ All_positive_rate, data = tweets_Data)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.022376 -0.007512 -0.002843  0.004656  0.045081
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       0.006770   0.007349   0.921   0.3691
## All_positive_rate 0.002210   0.001062   2.080   0.0521 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.01419 on 18 degrees of freedom
## Multiple R-squared:  0.1937, Adjusted R-squared:  0.1489
## F-statistic: 4.325 on 1 and 18 DF,  p-value: 0.05211
```

```r
plot(regression)
```




Normal Q−Q
Standardized residuals
−1
0
1
2
3
−2
−1
0
1
2
Theoretical Quantiles
lm(positive_rate ~ All_positive_rate)
1
11
7


Scale−Location
$\sqrt{|\text{Standardized residuals}|}$
0.0
0.5
1.0
1.5
0.010
0.015
0.020
0.025
0.030
Fitted values
lm(positive_rate ~ All_positive_rate)
1
7
11



By assuming that the missing values are missing at random (MAR), so that missingness is conditionally independent of the unmeasured value given the observed data. To accommodate missing values, we need to specify a joint distribution for $\mathbf{x}_i$ (collected tweets positive rate (ALL)), typically selected from normal or a sequence of conditional generalized linear models. Then, the missing values are simply additional unknowns to be updated in the Markov chain Monte Carlo (MCMC) algorithm. When the predictors have a normal likelihood and we have a linear regression model, i.e., $y_i = \beta \ x_i + \sigma^2$, missing new Tweets positive rate (containing key words flu shot) can be obtained by two steps:

(1) using simple Gibbs sampler by following update procedures to get linear model coefficients:

0. Starting with an initial value for β, σ^2, where β are model coefficients and σ^2 is variance;
1. Sample the missing values from their normal full conditional;
2. Given the imputed data, sample β and σ^2 from their full conditional posterior distributions.

(2) After linear model coefficients are obtained, the missing new Tweets positive rate (containing key words flu shot) of a subject can be obtained from linear regression and his/her Tweets positive rate (ALL).

The massive advantage of Gibbs sampling over other MCMC methods is that no tuning parameters are required. The drawbacks is the need of a fair bit of maths to derive the updates, sometimes they are not guaranteed to exist. The estiamted Tweets positive rates with key words flu shot are summarized at column Tweets-pos-rate.

```
Flu_Data_CDC <- Flu_Data[Flu_Data$Assign_Ind == 1, ]
Flu_Data_Placebo <- Flu_Data[Flu_Data$Assign_Ind == 0, ]
est_ATE_Tweets_pos_rate <- mean(Flu_Data_CDC$Tweets_pos_rate, na.rm = TRUE) - mean(Flu_Data_Placebo$Twe
print(est_ATE_Tweets_pos_rate)
```

```
## [1] -1.033194
```

```
model_pos_rate <- lm(Tweets_pos_rate ~ Assign_Ind, data = Flu_Data)
summary(model_pos_rate)
```

```
##
## Call:
## lm(formula = Tweets_pos_rate ~ Assign_Ind, data = Flu_Data)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -1.156 -1.109 -0.965 -0.095 53.166
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.1564     0.7844   1.474    0.144
## Assign_Ind   -1.0332     1.1598  -0.891    0.375
##
## Residual standard error: 5.602 on 92 degrees of freedom
## Multiple R-squared:  0.008552,   Adjusted R-squared:  -0.002224
## F-statistic: 0.7936 on 1 and 92 DF,  p-value: 0.3753
```

```
coef_model_pos_rate <- summary(model_pos_rate)$coef[2,]
#print(coef_model_pos_rate)
print(CI(coef_model_pos_rate))
```

```
## $ate.point.estimate
## Estimate
##  -1.0332
##
## $CI
## [1] "[-3.3064,1.24]"
##
## $p.value
## Pr(>|t|)
##   0.3753
```

```
t.test(Flu_Data_CDC$Tweets_pos_rate, Flu_Data_Placebo$Tweets_pos_rate, alternative = "two.sided")
```

```
##
##  Welch Two Sample t-test
##
## data:  Flu_Data_CDC$Tweets_pos_rate and Flu_Data_Placebo$Tweets_pos_rate
## t = -0.9708, df = 50.139, p-value = 0.3363
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
```

```
##  -3.170700  1.104313
## sample estimates:
## mean of x mean of y
## 0.1232244 1.1564181
```

So **the estimated effect of CDC is -1.0332**, with confidence inferval [-3.3064,1.24]. **The associated *p*-value is 0.3753** (statistically significant at the 5 percent level). According to our causal analysis based on new Tweets positive rate as an outcome measure, contrary to previous results in US area, the causal effect is negative.

Reference:

1. http://www.forbes.com/sites/federicoguerrini/2017/01/16/facebook-will-flag-and-filter-fake-news-in-germany/#76771f1a60e3
2. https://www.cdc.gov/flu/about/qa/misconceptions.htm
3. http://www.dailymail.co.uk/health/article-203142/Why-water-ward-flu.html