Jiancong Chen -Jerry.ai

This is Jiancong (Nigel) Chen's home assignment submission to Jerry.AI's interview. Question #1 was coded in R due to the richness of statistical packages as well as data analysis scripting packages. Question #2 was coded in Python and typed as text in the Rconsole.

Question 1

As no original/synthetic data was given, I constructed a synthetic dataset, that will enable me to show some data wrangling results, data visualization and statistical tools to make recommendations on whether the referral program should continue or stop rather than speaking on top of air about "what I want to do" or "how I can do".

Step 0: Construct synthetic tables

Note that this step is not needed when real data becomes available. This is only to make my further analysis more interpretative and representable. I generated a synthetic pool of ids (expressed as emails), name, and referring_user_id (also expressed as emails). The constructed table include situations where one user refer several new users.

For the Table Purchase, it includes ids that were referred by other users and ids who registered without referral. As the monthly premium for auto insurance may range between 0 (no need situation) and 120 (based on my own experience), I set the maximum total monthly cost for a user at 120. As I was not given any other coupon situations, so I assumed the discount is the referral (10/referral) + a random small number.

Note: there are 592 entries of User table.

```
rm(list = ls())
library(randNames)
set.seed(111)
## Generate the User table. The pool dataframe has N_user unique user names
N_user = 1000
Pool <- rand_names(n = N_user)[, c('email', 'name.first')]
```

```
## Warning: `as_data_frame()` is deprecated as of tibble 2.0.0.
## Please use `as_tibble()` instead.
## The signature and semantics have changed, see `?as_tibble`.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_warnings()` to see where this warning was generated.
```

```
names(Pool) <- c('id', 'name')
# Assuming a randN of the users were actually referred by other users.
# Note randN is generated randomly.
rand <- runif(1, min = 0, max = 1) # 0.59 for this case
idx_usr <- sample(N_user, size = N_user * rand, replace = TRUE)
idx_refer <- sample(N_user, size = N_user * rand, replace = FALSE)

id <- Pool$id[idx_usr]
refer <- Pool$id[idx_refer]
User <- data.frame(id = id, name = Pool$name[idx_usr], referring_user_id = refer)
```

The following code generates the data for the Table: purchase. It allows one user to have multiple purchase records. The total amount of purchase was randomly generated between 0 - 100. The discount was only simulated for the ones that succesfully refer other users as found in the User table. For the date variable, currently I am only simulating 2020 January - 2020 March for simplicity

```
N_purchase= 1000
total <- runif(N_purchase, 0, 10) * 10
daterange <- c(as.numeric(as.Date('2020-01-01')):as.numeric(as.Date('2020-03-31')))
date <- as.Date(sample(daterange, N_purchase, replace = TRUE), origin = '1970-01-01')
# Sample from the user pool. Also enable any users to have multiple purchase records
Purchase_id <- Pool[sample(1:N_user, N_purchase, replace = TRUE),]
Purchase <- data.frame(Purchase_id, total, date)
# For discount, only consider the ones with referral and give them $10
Purchase$discount <- 0 + runif(N_purchase, 0, 5)
Purchase$discount[which(Purchase$id %in% User$id)] = 10 + Purchase$discount[which(Purchase$id %in% User$id)]
```

Let us take a quick glimpse of the generated data

```
head(User)
```

	id <fct>	name <fct>	referring_user_id <fct>
1	mhrd.glshn@example.com	مهراد	nznyn.sdr@example.com
2	bojoura.kentie@example.com	Bojoura	guillaume.berger@example.com
3	ella.lehto@example.com	Ella	debra.dunne@example.com
4	baer.kauffman@example.com	Baer	julia.murto@example.com
5	martin.denis@example.com	Martin	tasmira.rocha@example.com
6	eric.tosse@example.com	Eric	mhrd.sdr@example.com

6 rows

```
head(Purchase)
```

	id <chr>	name <chr>	total <dbl>	date <date>	discount <dbl>
1	lima.desouza@example.com	Lima	97.54454	2020-02-05	13.194779
2	afsar.tokatlioglu@example.com	Afşar	37.36030	2020-02-02	12.459178
3	agnetha.donkersloot@example.com	Agnetha	25.11911	2020-02-09	1.729834
4	toni.dupuis@example.com	Toni	41.22814	2020-01-12	14.571302
5	harry.zhang@example.com	Harry	48.47339	2020-02-23	13.501691
6	cohen.hughes@example.com	Cohen	45.72990	2020-01-09	1.347864

6 rows

Step 1: What information would you derive from it, and how will you derive it?

1a Whether a referral lead to success acquisition of new customer, which is defined as a customer who makes purchase through Jerry.

```
# Filter the list of customers who were referred
Success_refer_usr <- unique(Purchase$id[which(Purchase$id %in% User$referring_user_id)])
# Glimpse the list
head(Success_refer_usr)
```

```
## [1] "lima.desouza@example.com"      "afsar.tokatlioglu@example.com"
## [3] "harry.zhang@example.com"       "cohen.hughes@example.com"
## [5] "kadir.agaoglu@example.com"     "eren.karaer@example.com"
```

1b The percentage of success acquisition

Ideally, we would want this metrix to be as high as possible. Based on the current synthetic dataset, the success rate is around 65%. I

```
print(length(Success_refer_usr)/length(User$referring_user_id))
```

```
## [1] 0.6165541
```

1c Risk perspective.

The percenrage of existing users who gambled the promotion, defined as a user who refered to multiple referrals but the new referrals are 'robot' who never make purchases. Based on the synthetic data, most users only have one referal, only 4 users have multiple referrals. If these referrals are not success users, these 4 users should be flagged and less promotions should be aviable for them for risk purposes and customer relations.

```
# Add a column to count the number of total refers for any specific user
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
```

```
library(ggplot2)
User_Agg <- User %>% group_by(referring_user_id) %>%
  mutate(count = n()) %>%
  mutate(count = replace(count, which(referring_user_id == 'No refer'), 0))

print(paste('Potential risky user:', User_Agg$id[which(User_Agg$count > 1)]))
```

```
## [1] "Potential risky user: ryan.levesque@example.com"
## [2] "Potential risky user: baer.kauffman@example.com"
```

1d Distribution of total between users who were refered and who were not refered

This can help us identify whether the referral system is actually helping us gain revenus. If there's significant differences in the total cost between the group of referred and non-referred, it indicates the referral program is contributing. Otherwise, we should dig dipper into the cause.

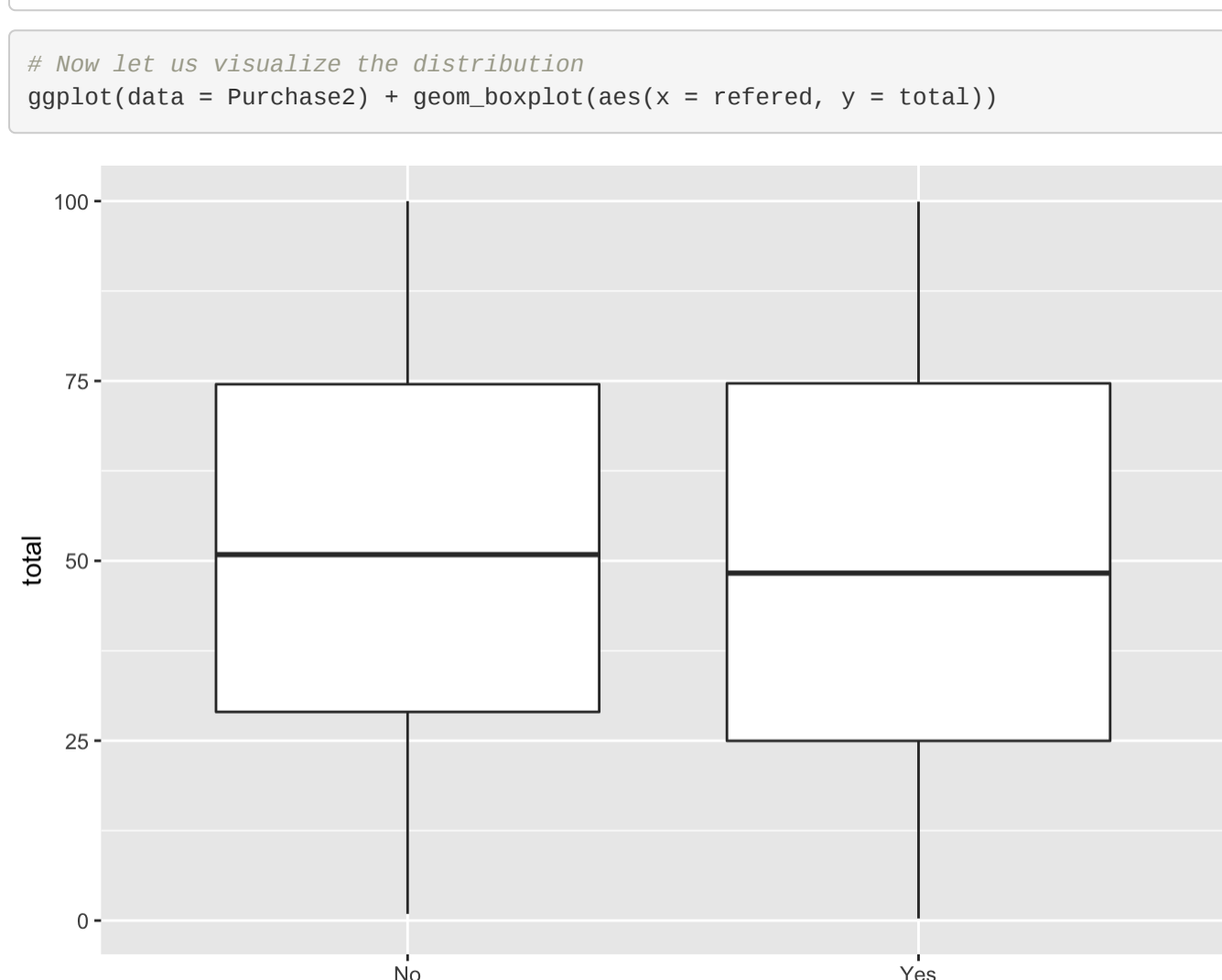
First, let us wrangle the data to create a new column for the purchase table that reflect whether this user is referred or not.

```
'%!in%' <- function(x,y)!('%in%'(x,y))
Purchase2 <- Purchase %>%
  mutate(refered = case_when(id %in% User$id ~ 'Yes',
                             id %!in% User$id ~ 'No'))
# Let us first take a look at the merged data frame.
head(Purchase2)
```

	id <chr>	name <chr>	total <dbl>	date <date>	discount <dbl>	refered <chr>
1	lima.desouza@example.com	Lima	97.54454	2020-02-05	13.194779	Yes
2	afsar.tokatlioglu@example.com	Afşar	37.36030	2020-02-02	12.459178	Yes
3	agnetha.donkersloot@example.com	Agnetha	25.11911	2020-02-09	1.729834	No
4	toni.dupuis@example.com	Toni	41.22814	2020-01-12	14.571302	Yes
5	harry.zhang@example.com	Harry	48.47339	2020-02-23	13.501691	Yes
6	cohen.hughes@example.com	Cohen	45.72990	2020-01-09	1.347864	No

6 rows

```
# Now let us visualize the distribution
ggplot(data = Purchase2) + geom_boxplot(aes(x = refered, y = total))
```

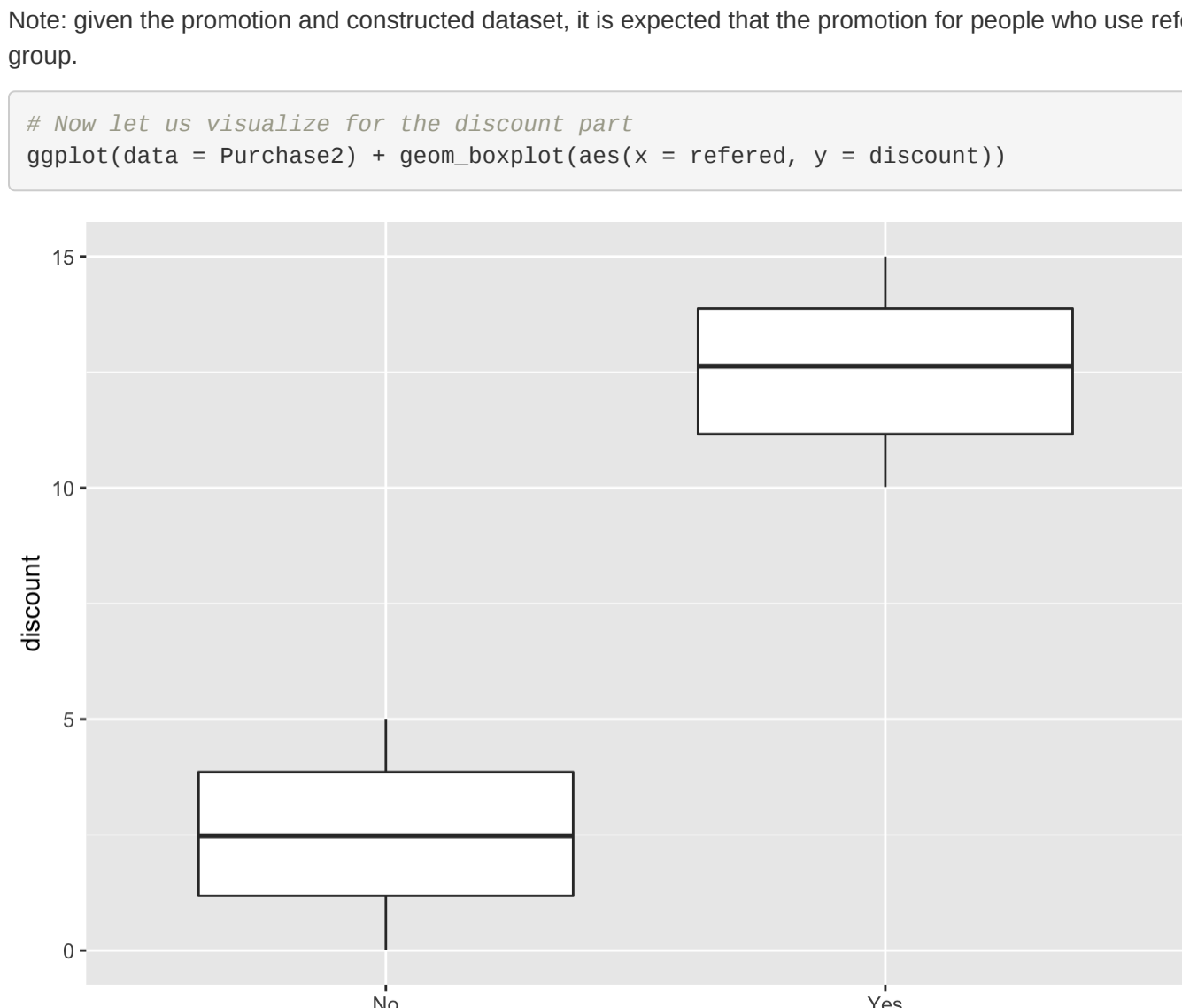


Note: as the total is generated randomly following uniform distribution, there should not be any significant differences among the groups.

1e Dsitribution of final bill between users who were refered and who were not refered

Note: given the promotion and constructed dataset, it is expected that the promotion for people who use refer will be higher than the non-refer group.

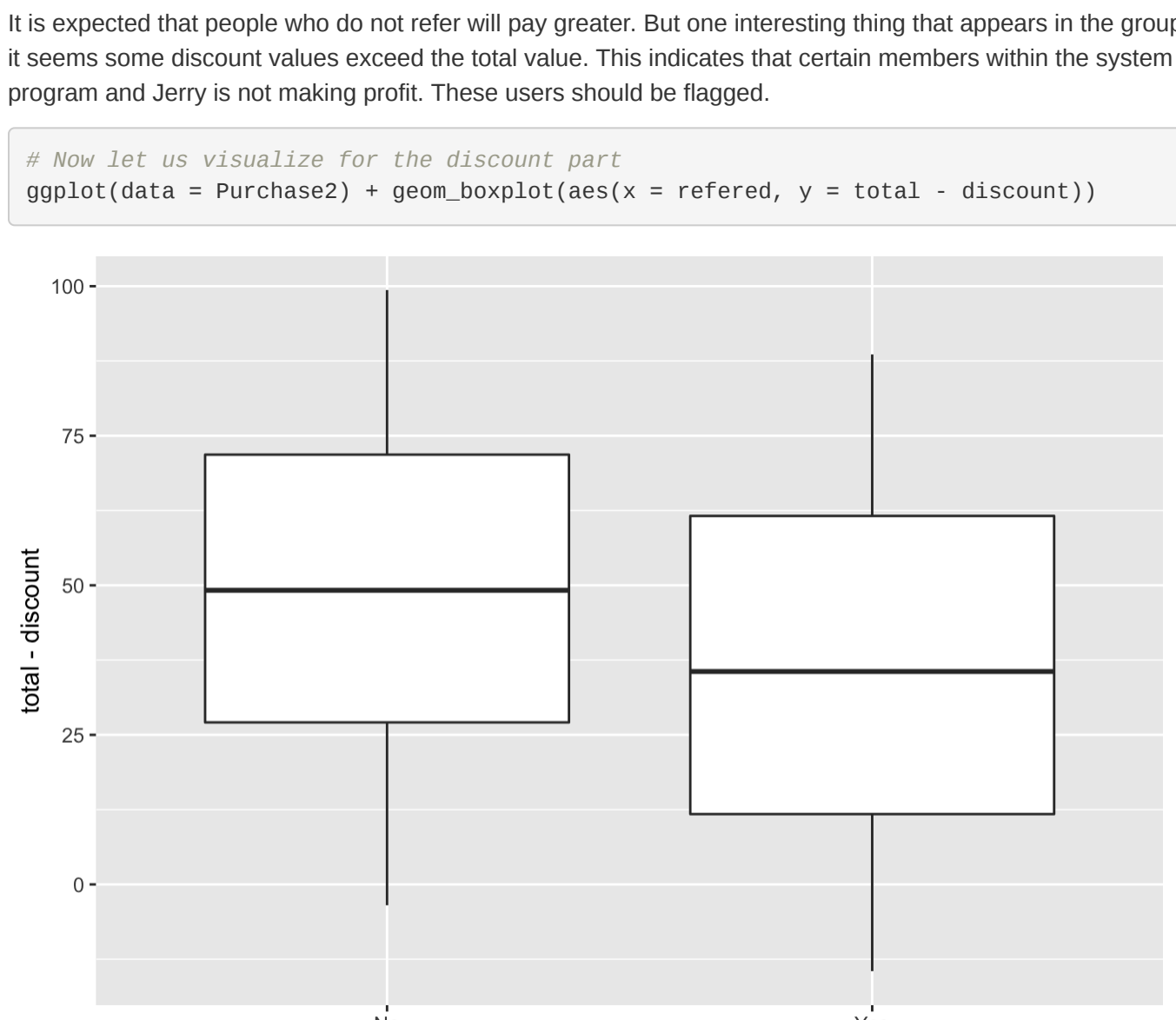
```
# Now let us visualize for the discount part
ggplot(data = Purchase2) + geom_boxplot(aes(x = refered, y = discount))
```



1f Are there any differences for the final bill among groups

It is expected that people who do not refer will pay greater. But one interesting thing that appears in the group boxplot is that for the referral group, it seems some discount values exceed the total value. This indicates that certain members within the system may be taking advantage of the program and Jerry is not making profit. These users should be flagged.

```
# Now let us visualize for the discount part
ggplot(data = Purchase2) + geom_boxplot(aes(x = refered, y = total - discount))
```



1g Month specific information? Do we observe any unique signal based on months?

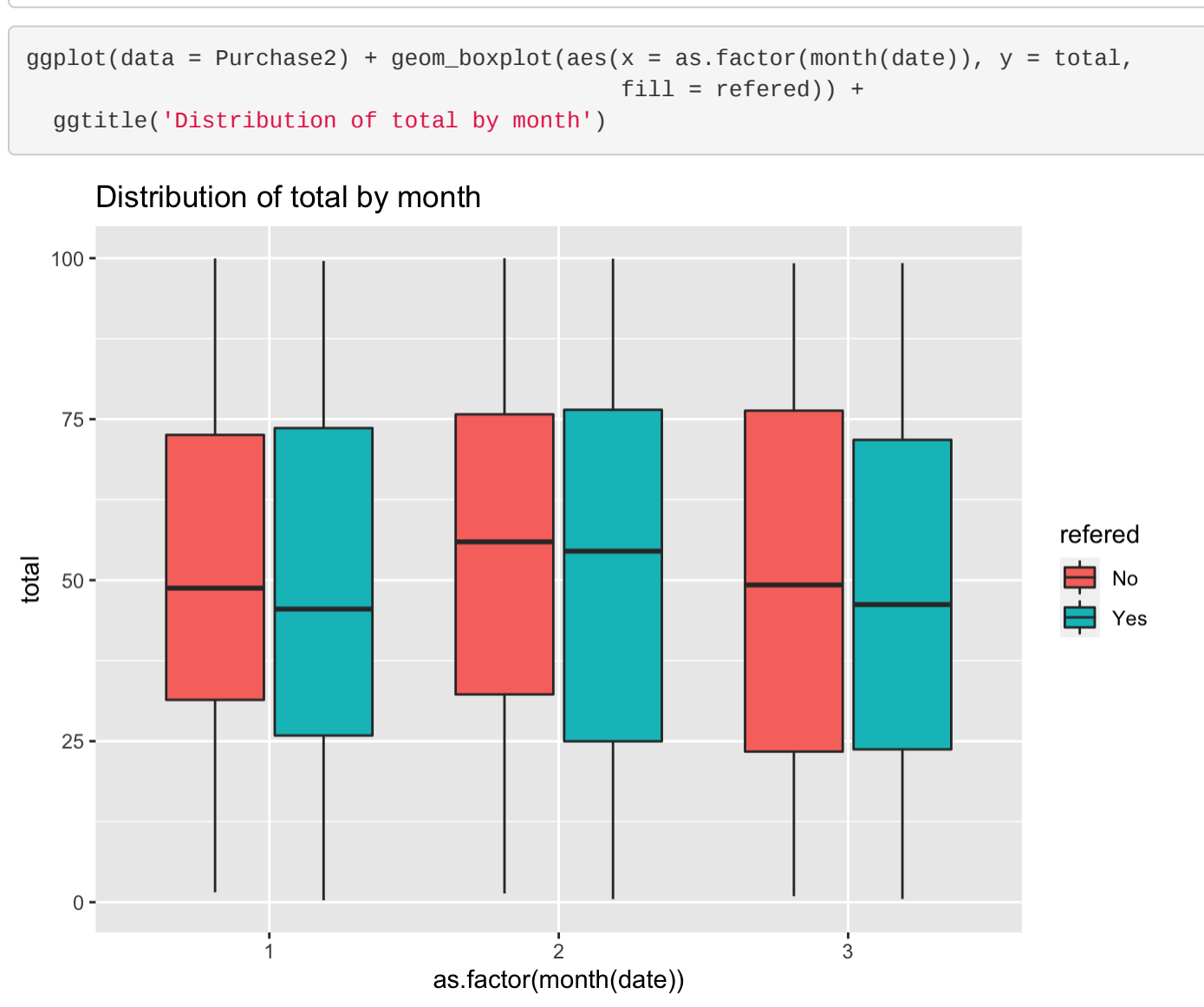
The last thing that I want to check is whether there's unique characteristics that we can sensor based on the time domain, though the synthetic data is currently treating each day/month equally plausible. But here is a sample code that could be useful. Even though it seems that January stands out between the groups, but this might be an artifact due to the random number generator.

```
library(lubridate)
```

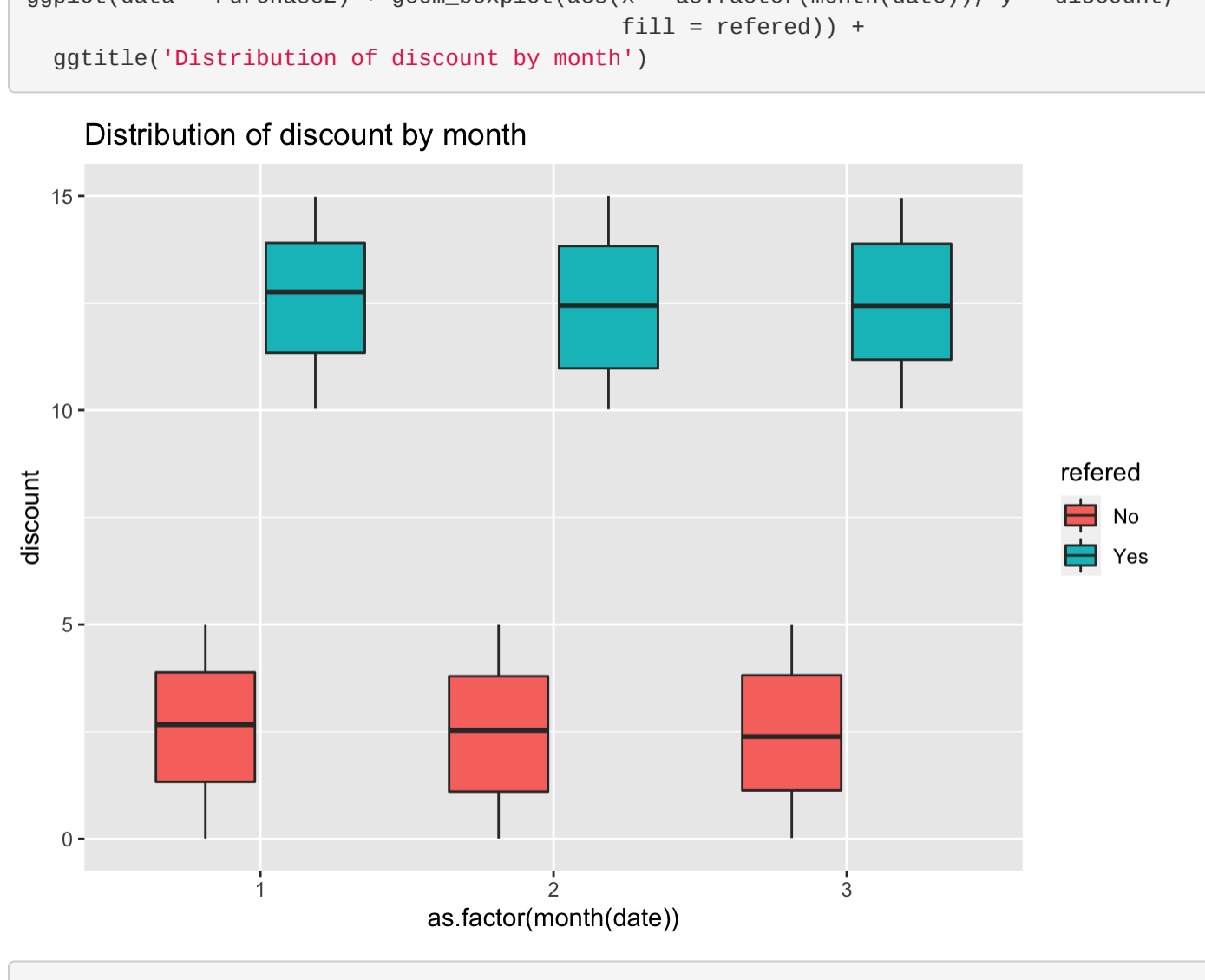
```
##
## Attaching package: 'lubridate'
```

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

```
ggplot(data = Purchase2) + geom_boxplot(aes(x = as.factor(month(date)), y = total,
                                            fill = refered)) +
  ggtitle('Distribution of total by month')
```



```
ggplot(data = Purchase2) + geom_boxplot(aes(x = as.factor(month(date)), y = discount,
                                            fill = refered)) +
  ggtitle('Distribution of discount by month')
```



```
ggplot(data = Purchase2) + geom_boxplot(aes(x = as.factor(month(date)), y = total - discount,
                                            fill = refered)) +
  ggtitle('Distribution of final bill by month')
```

Step 2: How would you make a recommendation

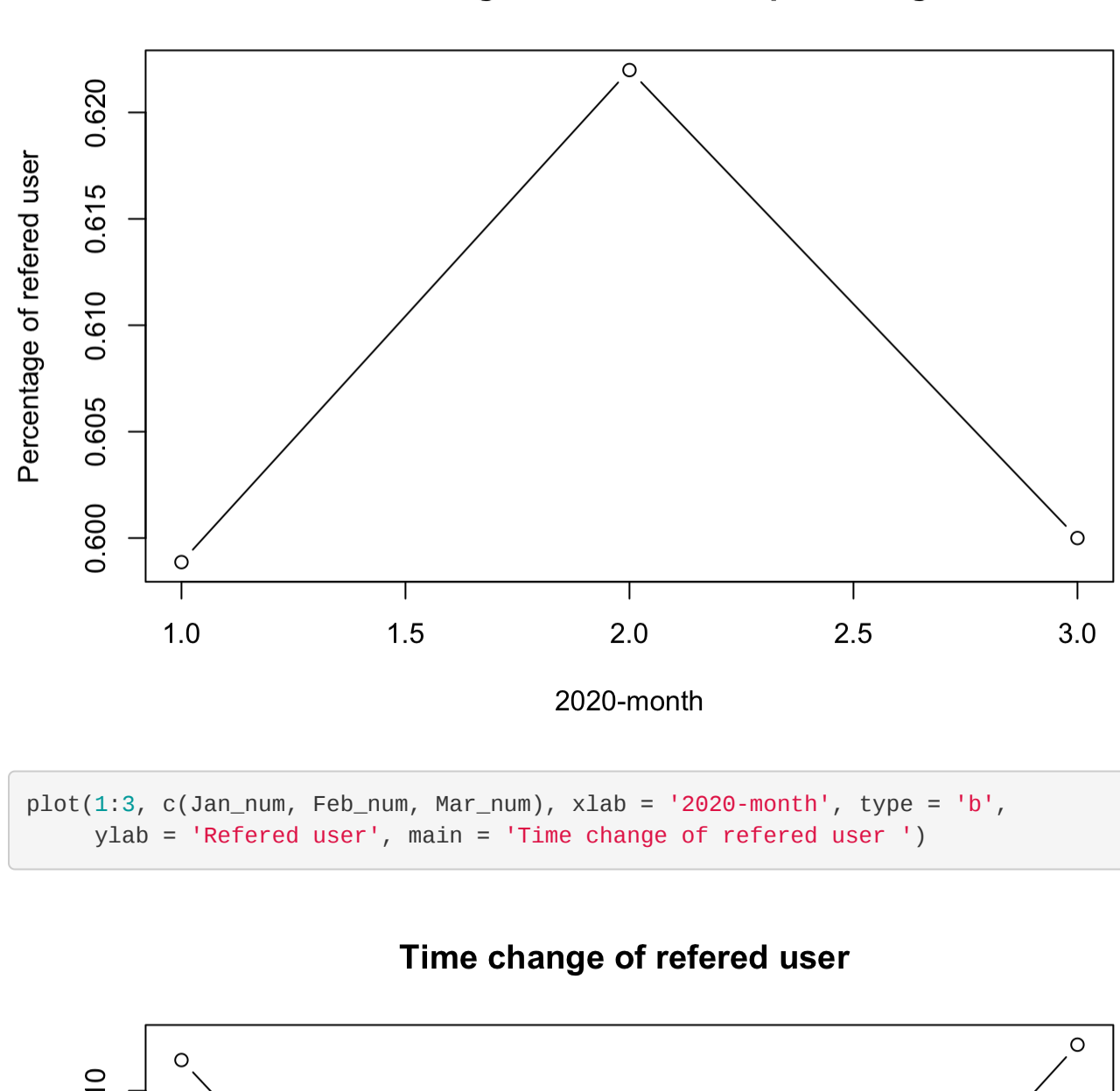
Use this informatin from Step 1), how would you make a recommendation on whether this rewards program should be continued or discontinued.

Answer:

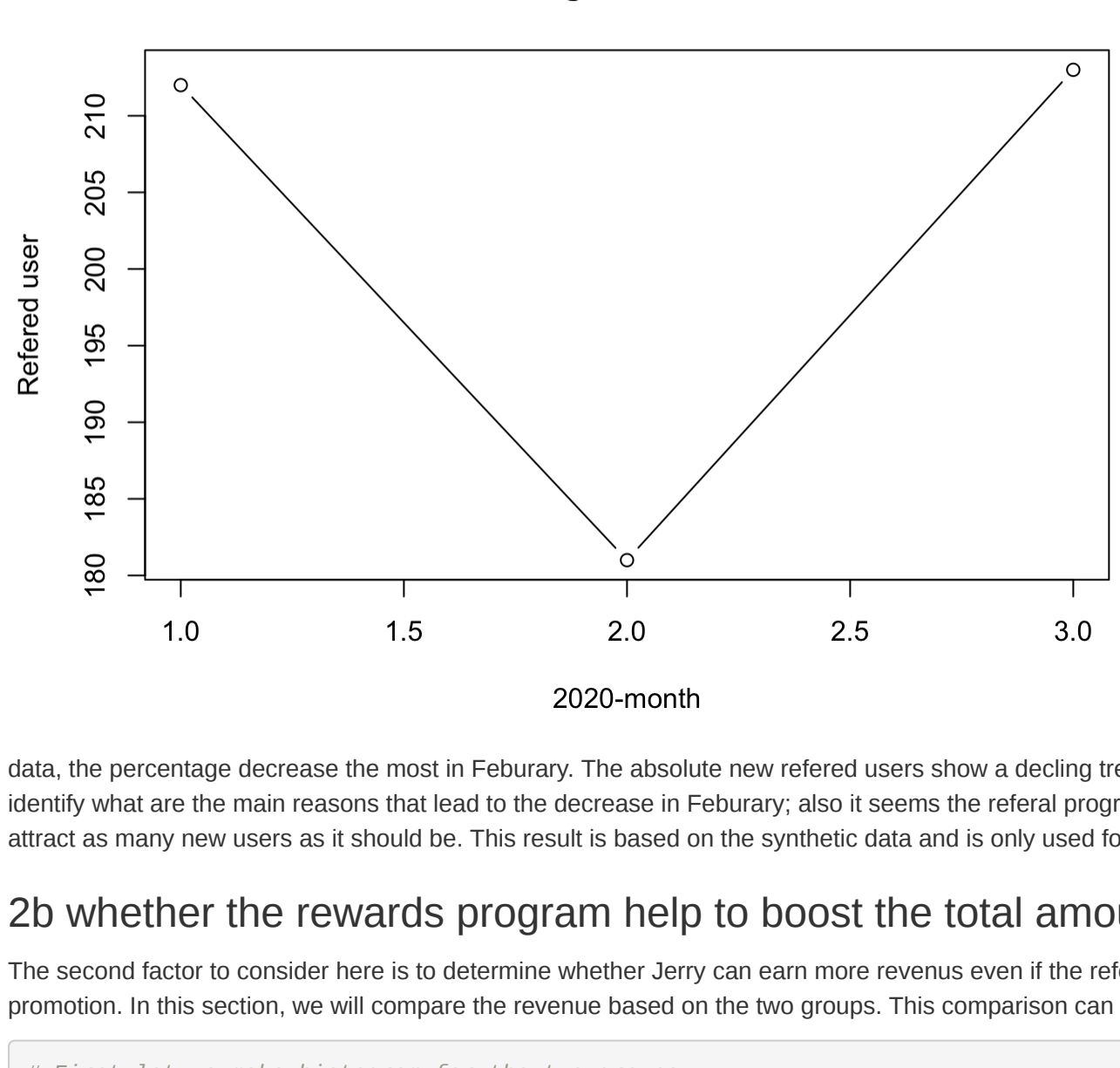
From the metrics we derived in part 1 and with data visualization, I would like to decompose the problem into the following sub-questions, which will be discussed with statistical tools and approaches. ### 2a whether the rewards program has significantly increased our successful candidates. I will use hypothesis testing to answer this sub-question. The null hypothesis is that: with the referral system, it can attract more new candidates into the system compared to no referral system.

We can look at the growth of referred users and non-referred users over time; and see if there have been a significant change.

```
## Subsetting Purchase into Jan Feb and March
Jan_Pur <- Purchase2 %>%
  filter(month(date) == 1)

Feb_Pur <- Purchase2 %>%
  filter(month(date) == 2)

Mar_Pur <- Purchase2 %>%
  filter(month(date) == 3)

# Let us consider the percentage of refered users
Jan_percentage <- length(which(Jan_Pur$id %in% User$referring_user_id))/length(Jan_Pur$id)
Feb_percentage <- length(which(Feb_Pur$id %in% User$referring_user_id))/length(Feb_Pur$id)
Mar_percentage <- length(which(Mar_Pur$id %in% User$referring_user_id))/length(Mar_Pur$id)

# Let us also consider the absolute number of user change
Jan_num <- length(which(Jan_Pur$id %in% User$referring_user_id))
Feb_num <- length(which(Feb_Pur$id %in% User$referring_user_id))
Mar_num <- length(which(Mar_Pur$id %in% User$referring_user_id))

plot(1:3, c(Jan_percentage, Feb_percentage, Mar_percentage), xlab = '2020-month', type = 'b',
     ylab = 'Percentage of refered user', main = 'Time change of refered user percentage')
```



```
plot(1:3, c(Jan_num, Feb_num, Mar_num), xlab = '2020-month', type = 'b',
     ylab = 'Refered user', main = 'Time change of refered user ')
```



Based on the synthetic generated data, the percentage decrease the most in Feburary. The absolute new refered users show a decling trend. These results indicate that we need to identify what are the main reasons that lead to the decrease in Feburary; also it seems the referal program is not working promptly, as it does not attract as many new users as it should be. This result is based on the synthetic data and is only used for illustration.

2b whether the rewards program help to boost the total amount

The second factor to consider here is to determine whether Jerry can earn more revenus even if the referal program is giving out 10 dollar promotion. In this section, we will compare the revenue based on the two groups. This comparison can be done with hypothesis testing

```
# First let us make histogram for the two groups
ggplot(data = Purchase2) + geom_histogram(aes(total, group = refered, color = refered))
```

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

```
# Now let us set up the hypothesis testing (t-test)
test1 <- t.test(total ~ refered, data = Purchase2)
test1
```

```
##
##  Welch Two Sample t-test
##
## data:  total by refered
## t = 1.0452, df = 935.35, p-value = 0.2962
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -1.652090  5.417223
## sample estimates:
##  mean in group No mean in group Yes
##          51.04327          49.16070
```

```
# Let us also try tukey's test
model <- lm(total ~ refered, data = Purchase2)
m_anova <- aov(model)
m_tukey <- TukeyHSD(x = m_anova)
m_tukey
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = model)
##
## $refered
##             diff       lwr      upr     p adj
## Yes-No -1.882567 -5.401025 1.635892 0.2939904
```

Based on the synthetic data, we did not observe a significant difference of total cost between referred groups and non-referred group. The t-test returned a p-value of 0.4383, indicating the null hypothesis should not be rejected. There's no significant difference between the referred group versus the non-referred group. The ANOVA and Tukey's HSD test also indicate a p-valud at 0.439. There is no significant differences between the two group.

From this information, it seems by prompting the 10 dollar bonus/coupon will not really boost our revenue. There might be people who only take them as a way of lowering their expense.

2c Should we be considerate about certain users who may risky and only take the promotion?

This category can be very interesting as there might be a group of people who post their referal link online and get the promo code. But in reality, they never really refer to people. In this severe case, the promotion is mostly wasted and will not result in any boost in revenue or increase in new users. We can take a look at the amount of refered people who may not be real user.

```
length(which(User$referring_user_id %in% Purchase$id))/length(User$referring_user_id)
```

```
## [1] 0.6182432
```

This simple algebra indicates only ~ 64.9% of the referred user are actually qualified users who will make a purchase through Jerry at a later time. If we believe 64.9% is not sufficient, then I will recommend canceling the program as it does not yield a satisfactory user attraction rate.

Question 2

Code block

The following code implements the add and remove function in Python that manages disjointed intervals of integers. I have tested all the exam sequence with the satisfactory result as shown.

```
class AddnRemove():
    def __init__(self):
        self.intervals = []
    def add(self, fr, to):
        newItvs = []
        start = fr
        end = to
        for interval in self.intervals:
            if interval[0] < fr:
                if interval[1] < fr:
                    newItvs.append(interval)
                elif interval[1] >= fr and interval[1] <= to:
                    start = interval[0]
                else:
                    end = interval[1]
            elif interval[0] >= fr and interval[0] <= to:
                if interval[1] <= to:
                    continue
                else:
                    end = interval[1]
            else:
                newItvs.append(interval)
        newItvs.append([start, end])
        self.intervals = newItvs
        return self.intervals

    def remove(self, fr, to):
        newItvs = []
        for interval in self.intervals:
            if interval[0] < fr:
                if interval[1] < fr:
                    newItvs.append(interval)
                elif interval[1] >= fr and interval[1] <= to:
                    newItvs.append([interval[0], fr])
                else:
                    newItvs.append([interval[0], fr])
                    newItvs.append([to, interval[1]])
            elif interval[0] >= fr and interval[0] <= to:
                if interval[1] <= to:
                    continue
                else:
                    newItvs.append([to, interval[1]])
            else:
                newItvs.append(interval)
        self.intervals = newItvs
        return self.intervals
```

Now, we can execute the codes. The results indicate the add/remove functions are working promptly.

```
obj = AddnRemove()
print(obj.add(1, 5))
```

```
## [[1, 5]]
```

```
print(obj.remove(2, 3))
```

```
## [[1, 2], [3, 5]]
```

```
print(obj.add(6, 8))
```

```
## [[1, 2], [3, 5], [6, 8]]
```

```
print(obj.remove(4, 7))
```

```
## [[1, 2], [3, 4], [7, 8]]
```

```
print(obj.add(2, 7))
```

```
## [[1, 8]]
```

```
print(obj.add(10, 11))
```

```
## [[1, 8], [10, 11]]
```

```
print(obj.remove(9, 10))
```

```
## [[1, 8], [10, 11]]
```