Premium user conversion

"Free to Fee" Strategy

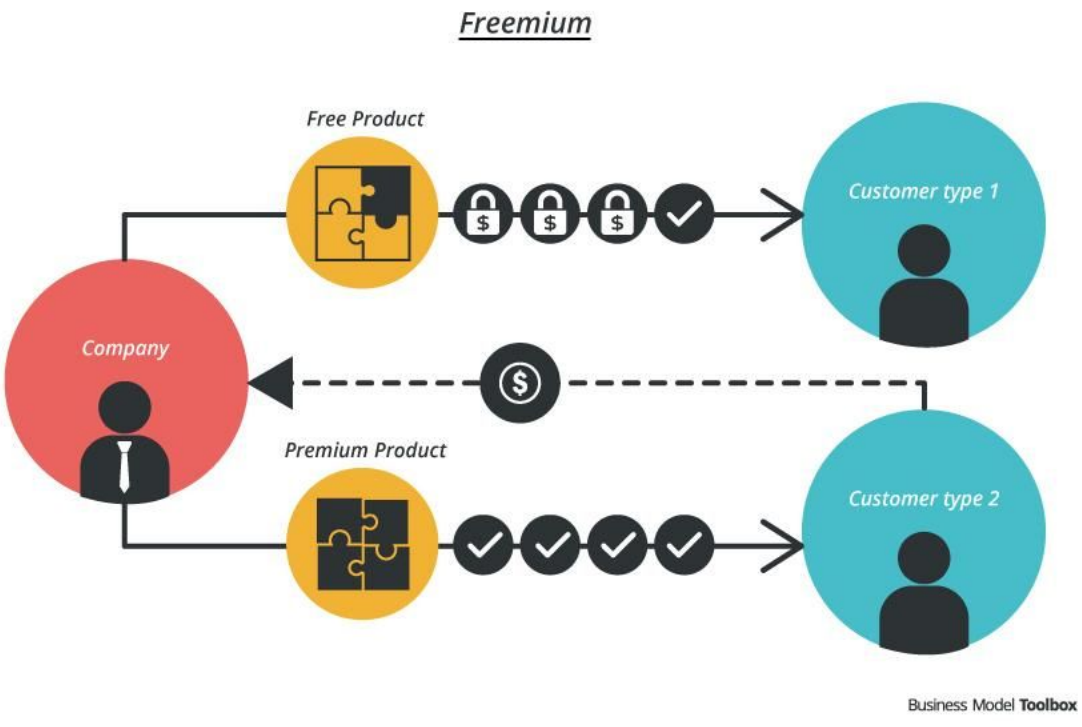
DEC. 2018, by Yuzi Liu

Overview

The "freemium" business model — widely used by online services such as LinkedIn, Match.com, Dropbox, and music-listening sites — divides user populations into groups that use the service for free and groups that pay a fee for additional features.

Given the higher profitability of premium subscribers, it is generally in the interest of company to motivate users to go from "free to fee"; that is, convert free accounts to premium subscribers.

This project intends to analyze the data from an APP of an anonymized real music streaming company for potential insight to inform a "free-to-fee" strategy.



PROCESS

In this project, I analyzed customer data of music app categorized by demographic, social network, and engagement, used PSM model to estimate potential treatment effect, conduct logistic regression analysis, determined "free-to-fee" strategy for converting free accounts to premium subscribers.

Summary Statistics

The first step is to generate descriptive statistics for the key variables in the dataset, summary the descriptive statistics of adopter and non-adopter.

adopters summary

Statistic	N	Mean	St. Dev.	Min	Pctl(25)	Median	Pctl(75)	Max
age	3,527	25.98	6.84	8	21	24	29	73
male	3,527	0.73	0.44	0	0	1	1	1
friend_cnt	3,527	39.73	117.27	1	7	16	40	5,089
avg_friend_age	3,527	25.44	5.21	12	22.1	24.4	27.6	62
avg_friend_male	3,527	0.64	0.25	0.00	0.50	0.67	0.81	1.00
friend_country_cnt	3,527	7.19	8.86	0	2	4	9	136
subscriber_friend_cnt	3,527	1.64	5.85	0	0	0	2	287
songsListened	3,527	33,758.04	43,592.73	0	7,804.5	20,908	43,989.5	817,290
lovedTracks	3,527	264.34	491.43	0	30	108	292	10,220
posts	3,527	21.20	221.99	0	0	0	2	8,506
playlists	3,527	0.90	2.56	0	0	1	1	118
shouts	3,527	99.44	1,156.07	0	2	9	41	65,872
adopter	3,527	1.00	0.00	1	1	1	1	1
tenure	3,527	45.58	20.04	0	32	46	60	111
good_country	3,527	0.29	0.45	0	0	0	1	1

Non-adopters summary

Statistic	N	Mean	St. Dev.	Min	Pctl(25)	Median	Pctl(75)	Max
age	40,300	23.95	6.37	8	20	23	26	79
male	40,300	0.62	0.48	0	0	1	1	1
friend_cnt	40,300	18.49	57.48	1	3	7	18	4,957
avg_friend_age	40,300	24.01	5.10	8.00	20.67	23.00	26.06	77.00
avg_friend_male	40,300	0.62	0.32	0.00	0.43	0.67	0.90	1.00
friend_country_cnt	40,300	3.96	5.76	0	1	2	4	129
subscriber_friend_cnt	40,300	0.42	2.42	0	0	0	0	309
songsListened	40,300	17,589.44	28,416.02	0	1,252	7,440	22,892.8	1,000,000
lovedTracks	40,300	86.82	263.58	0	1	14	72	12,522
posts	40,300	5.29	104.31	0	0	0	0	12,309
playlists	40,300	0.55	1.07	0	0	0	1	98
shouts	40,300	29.97	150.69	0	1	4	15	7,736
adopter	40,300	0.00	0.00	0	0	0	0	0
tenure	40,300	43.81	19.79	1	29	44	59	111
good_country	40,300	0.36	0.48	0	0	0	1	1

Then I conduct t-tests to compare difference in mean values of the variables for adopter and non-adopter.

```{r}
hn_cov <- c('age', 'male', 'friend_cnt', 'avg_friend_age', 'avg_friend_male', 'friend_country_cnt',
            'subscriber_friend_cnt', 'songsListened', 'lovedTracks', 'posts', 'playlists',
            'shouts', 'tenure', 'good_country')
Highnote %>%
  group_by(adopter) %>%
  select(one_of(hn_cov)) %>%
  summarise_all(funs(mean(., na.rm = T)))

lapply(hn_cov, function(v) {
  t.test(Highnote[, v] ~ Highnote$adopter)
})
```

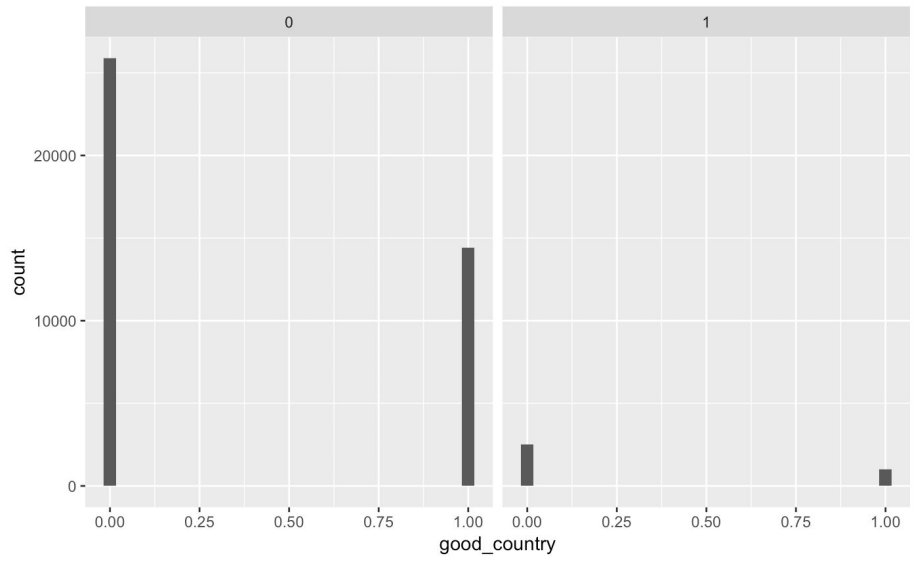
From the result, the mean difference of all covariates is significant.

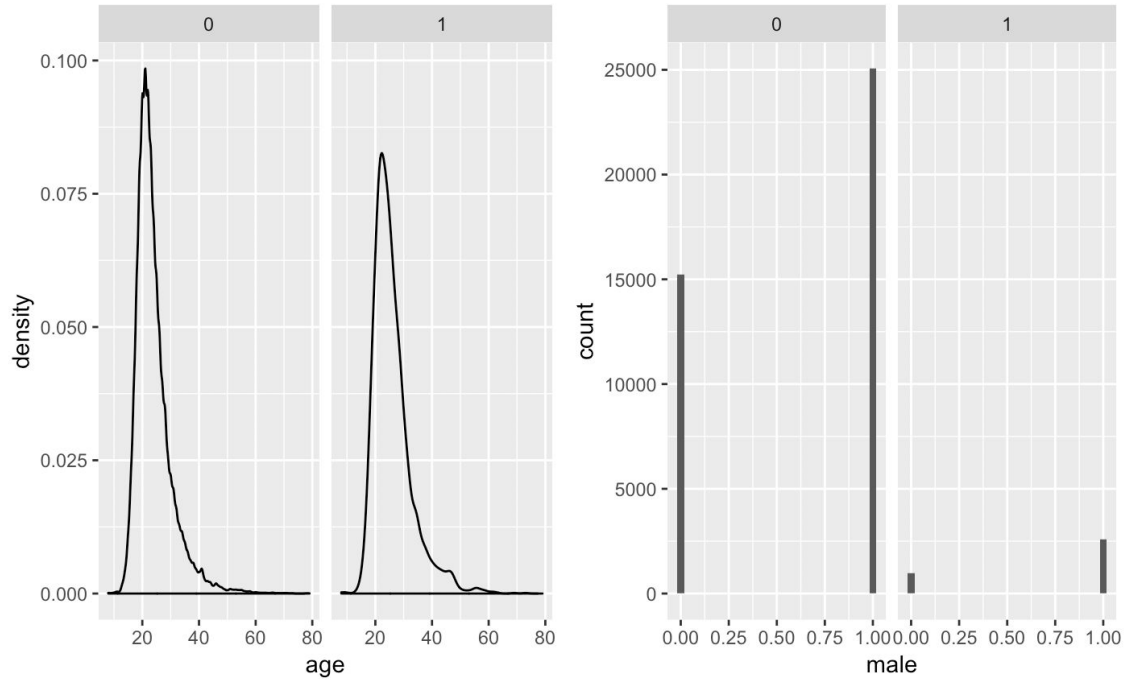
From these comparisons, we can make a tentative conclusion that:

- Users who are older, male, and their friends are older, tend to become fee-users
- Users who have more friends, and more friends from different countries, tend to become fee-users.
- Users who have more premium friends, tend to become fee-users.
- Users who are more engaging (listened more songs, loved more tracks, made more posts and playlists, received more shouts, been on the site longer) are more likely to become fee-users.

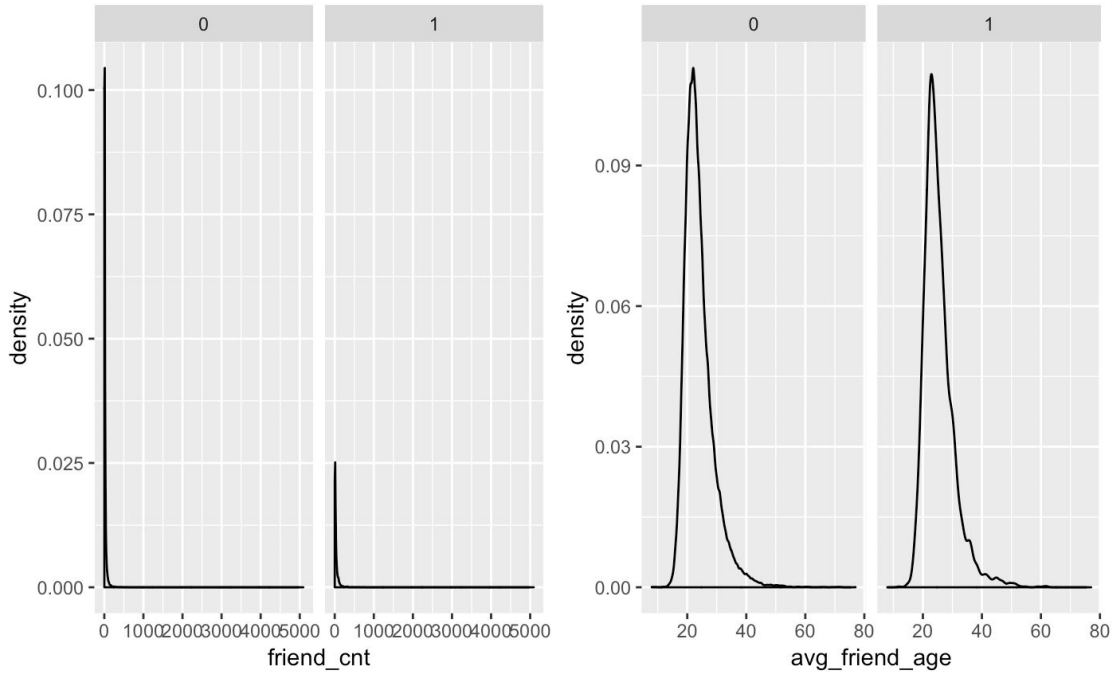
Data Visualization using *ggplot*

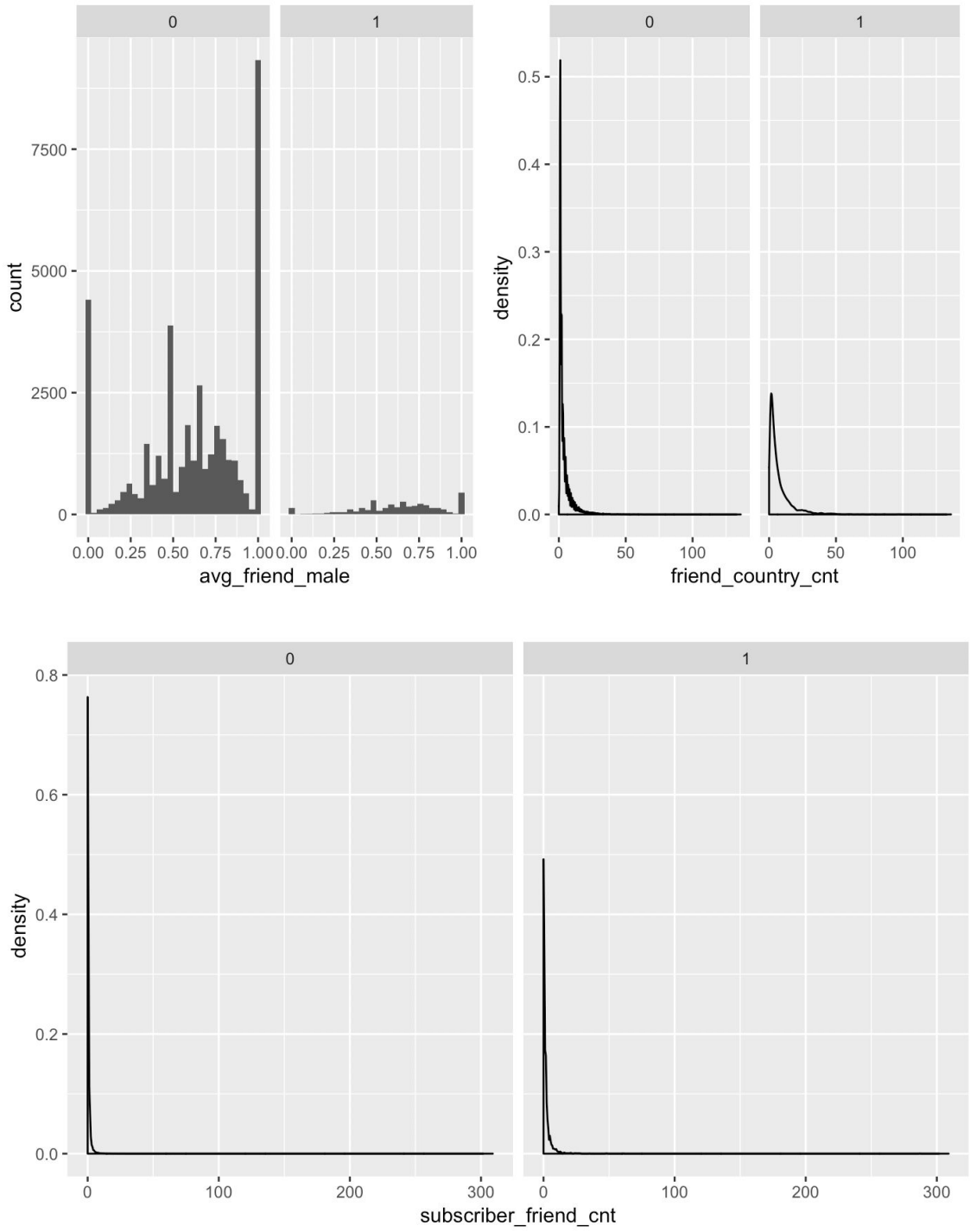
(i) Demographics



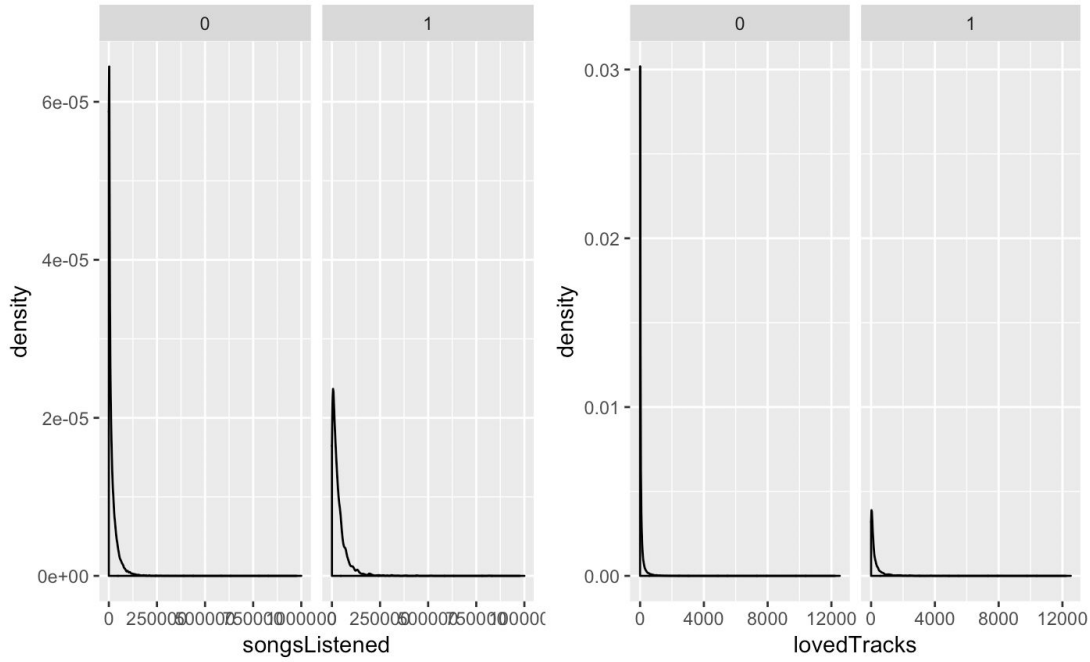


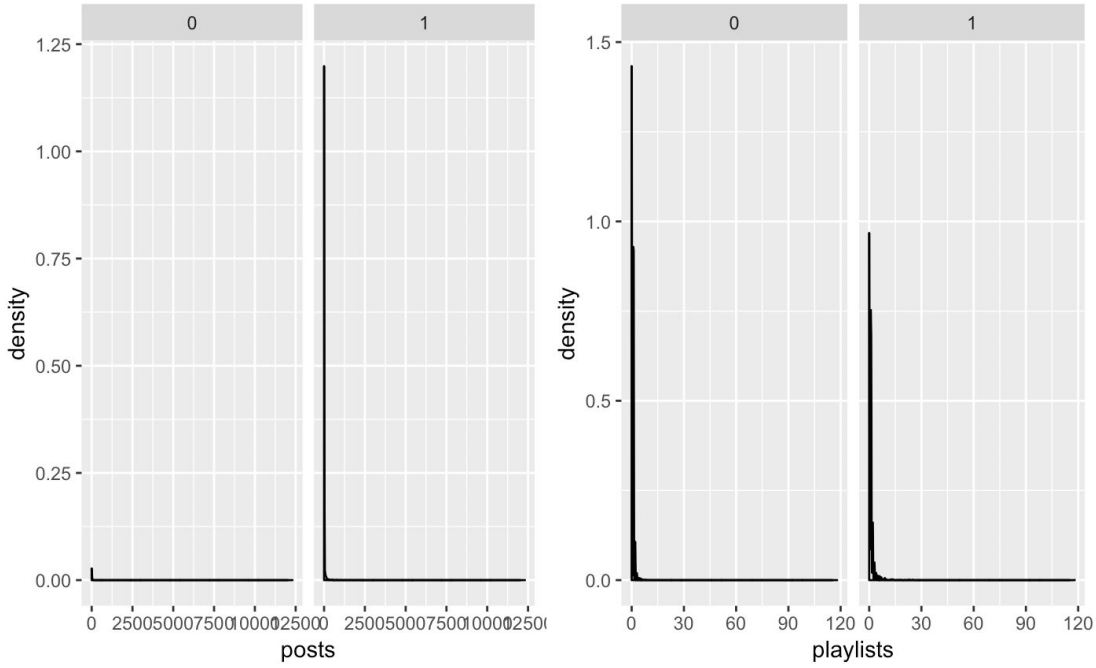
(ii) Peer Influence

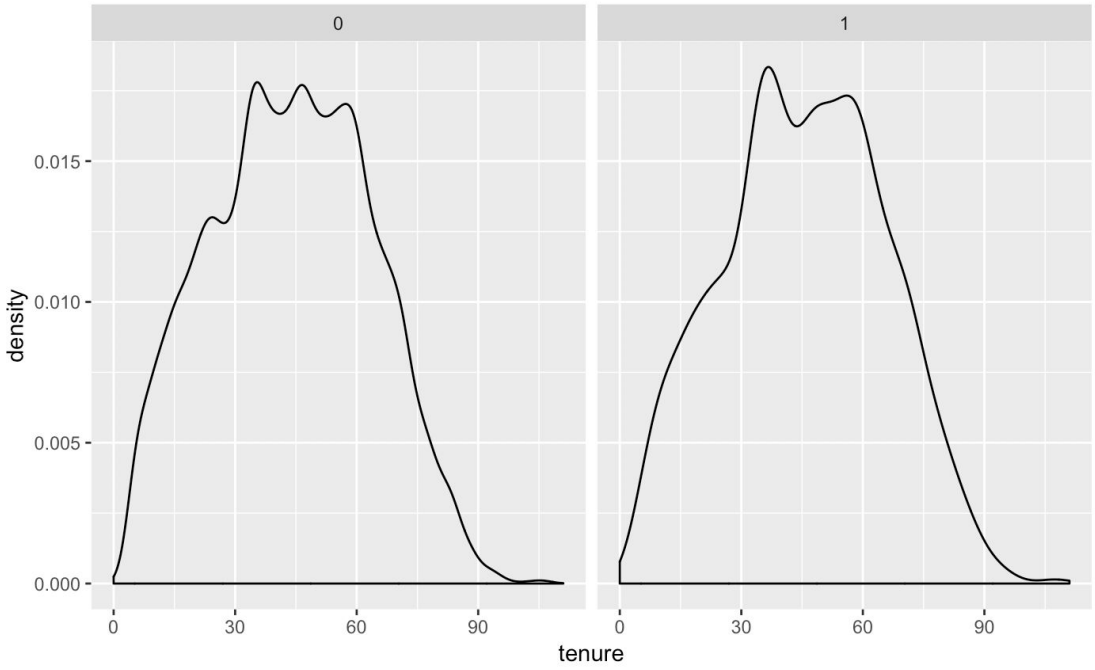




(iii) User Engagement


0
1
density
6e-05
4e-05
2e-05
0e+00
songsListened
0.03
0.02
0.01
0.00
lovedTracks
4000
8000
12000


0
1
density
1.25
1.00
0.75
0.50
0.25
0.00
posts
1.5
1.0
0.5
0.0
playlists
30
60
90
120



From the visualization, we can make **same conclusion** as the mean difference analysis that:

- Users who are older, male, and their friends are older, tend to become fee-users Users who have more friends, and more friends from different countries, tend to become fee-users.
- Users who have more premium friends, tend to become fee-users.
- Users who are more engaging (listened more songs, loved more tracks, made more posts and playlists, received more shouts, been on the site longer) are more likely to become fee-users.

Propensity Score Matching

This model intends to create treatment and control groups where:

- "treatment" group: users that have one or more subscriber friends (subscriber_friend_cnt >= 1)
- "control" group: users with zero subscriber friends (subscriber_friend_cnt = 0)

1. Pre-analysis use non-matched data

- **1.1: Difference-in-means: outcome variable**

Using adopter as the outcome variable of interest. (1 = adopter; 0 = non-adopter), the independent variable of interest is ynsf. (1 = having subcriber friends; 0 = not having)

```
```{r}
with(Highnote, t.test(adopter ~ ynsf))

```
```

```
        Welch Two Sample t-test

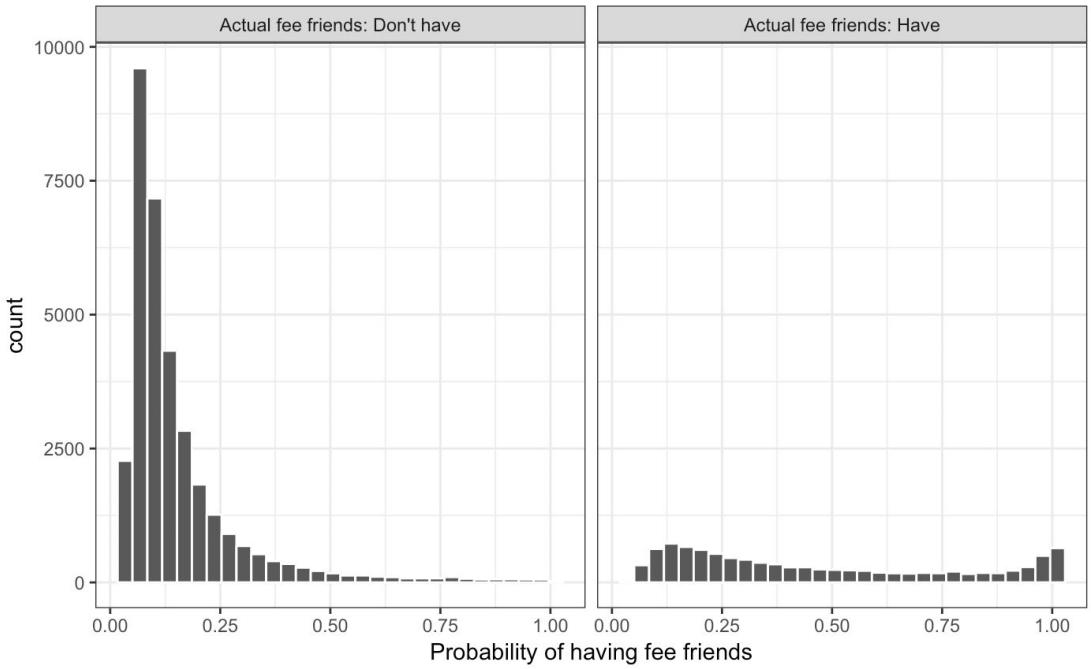
data:  adopter by ynsf
t = -30.961, df = 11815, p-value < 2.2e-16
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.1330281 -0.1171869
sample estimates:
mean in group 0 mean in group 1
     0.05243501      0.17754250
```

We see that the difference-in-means is statistically significant at conventional levels of confidence.

- **1.2: Difference-in-means: pre-treatment covariates**

Calculate the mean for each covariate by the treatment status:

```
```{r}
hn_cov2 <- c('age', 'male', 'friend_cnt', 'avg_friend_age', 'avg_friend_male', 'friend_country_cnt',
             'songsListened', 'lovedTracks', 'posts', 'playlists',
             'shouts', 'tenure', 'good_country')
Highnote %>%
  group_by(ynsf) %>%
  select(one_of(hn_cov2)) %>%
  summarise_all(funs(mean(., na.rm = T)))
```
```

Then we can carry out t-tests to evaluate whether these means are statistically distinguishable:

```
```{r}
lapply(hn_cov2, function(v) {
  t.test(Highnote[, v] ~ Highnote[, 'ynsf'])
})
```
```

From the result, we see that except for 'male', all mean values of other variables are statistically distinguishable.

We should therefore exclude 'male' in the PSM logit model.

2. Propensity Score Estimation

We estimate the propensity score by running a logit model, where the outcome variable is a binary variable indicating treatment status.

```
Call:
glm(formula = ynsf ~ age + friend_cnt + avg_friend_age + avg_friend_male +
    friend_country_cnt + songsListened_1k + lovedTracks + posts +
    playlists + shouts + tenure + good_country, family = binomial(),
    data = Highnote)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-4.4154  -0.5668  -0.4221  -0.3009   2.5520

Coefficients:
                     Estimate Std. Error z value Pr(>|z|)
(Intercept)        -5.124e+00  7.566e-02 -67.720  < 2e-16 ***
age                 2.043e-02  2.757e-03   7.409 1.27e-13 ***
friend_cnt          3.131e-02  1.033e-03  30.295  < 2e-16 ***
avg_friend_age      7.904e-02  3.460e-03  22.843  < 2e-16 ***
avg_friend_male     2.528e-01  5.027e-02   5.030 4.92e-07 ***
friend_country_cnt  1.105e-01  4.751e-03  23.266  < 2e-16 ***
songsListened_1k    7.012e-03  5.107e-04  13.731  < 2e-16 ***
lovedTracks         6.685e-04  5.644e-05  11.845  < 2e-16 ***
posts               5.753e-04  2.686e-04   2.142   0.0322 *
playlists           5.249e-03  1.191e-02   0.441   0.6593
shouts             -5.027e-05  3.678e-05  -1.367   0.1717
tenure             -2.534e-03  7.766e-04  -3.262   0.0011 **
good_country        3.088e-02  2.921e-02   1.057   0.2903
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

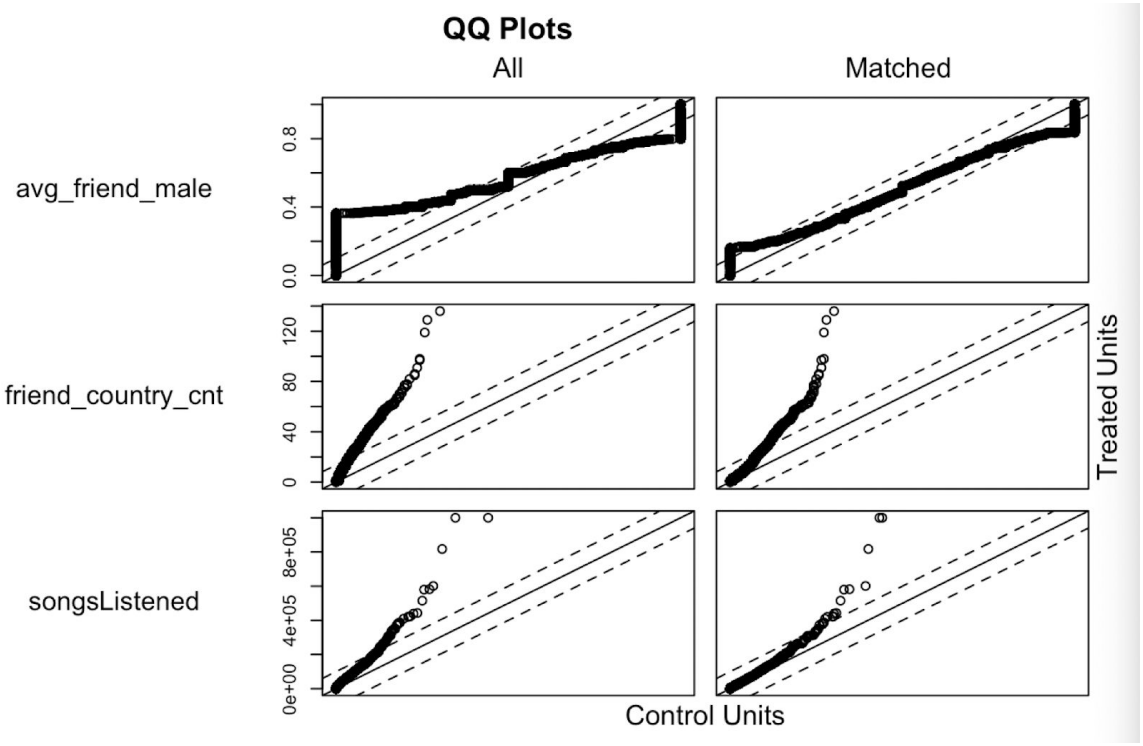
    Null deviance: 46640  on 43826  degrees of freedom
Residual deviance: 34173  on 43814  degrees of freedom
AIC: 34199

Number of Fisher Scoring iterations: 8
```

After that, we calculate the propensity score for each user. That is, the user's predicted probability of being Treated, given the estimates from the logit model.

	pr_score <dbl>	ynsf <dbl>
1	0.08810050	0
2	0.14832644	0
3	0.08121395	0
4	0.24291404	1
5	0.70270131	0
6	0.22199154	0

- ## 2.1 Examining the region of common support

We can plot histograms of the estimated propensity scores by treatment status:



3. Executing a Matching Algorithm

I find pairs of observations that have very similar propensity scores, but that differ in their treatment status.

```
```{r}
Highnote_nomiss <- Highnote %>%  # MatchIt does not allow missing values
  select(adopter, ynsf, one_of(hn_cov2)) %>%
  na.omit()

mod_match <- matchit(ynsf ~ age + friend_cnt + avg_friend_age + avg_friend_male + friend_country_cnt
                     + songsListened + lovedTracks + posts + playlists
                     + shouts + tenure + good_country,
                     method = "nearest", data = Highnote_nomiss)

```
```

Example of how successful the matching works:



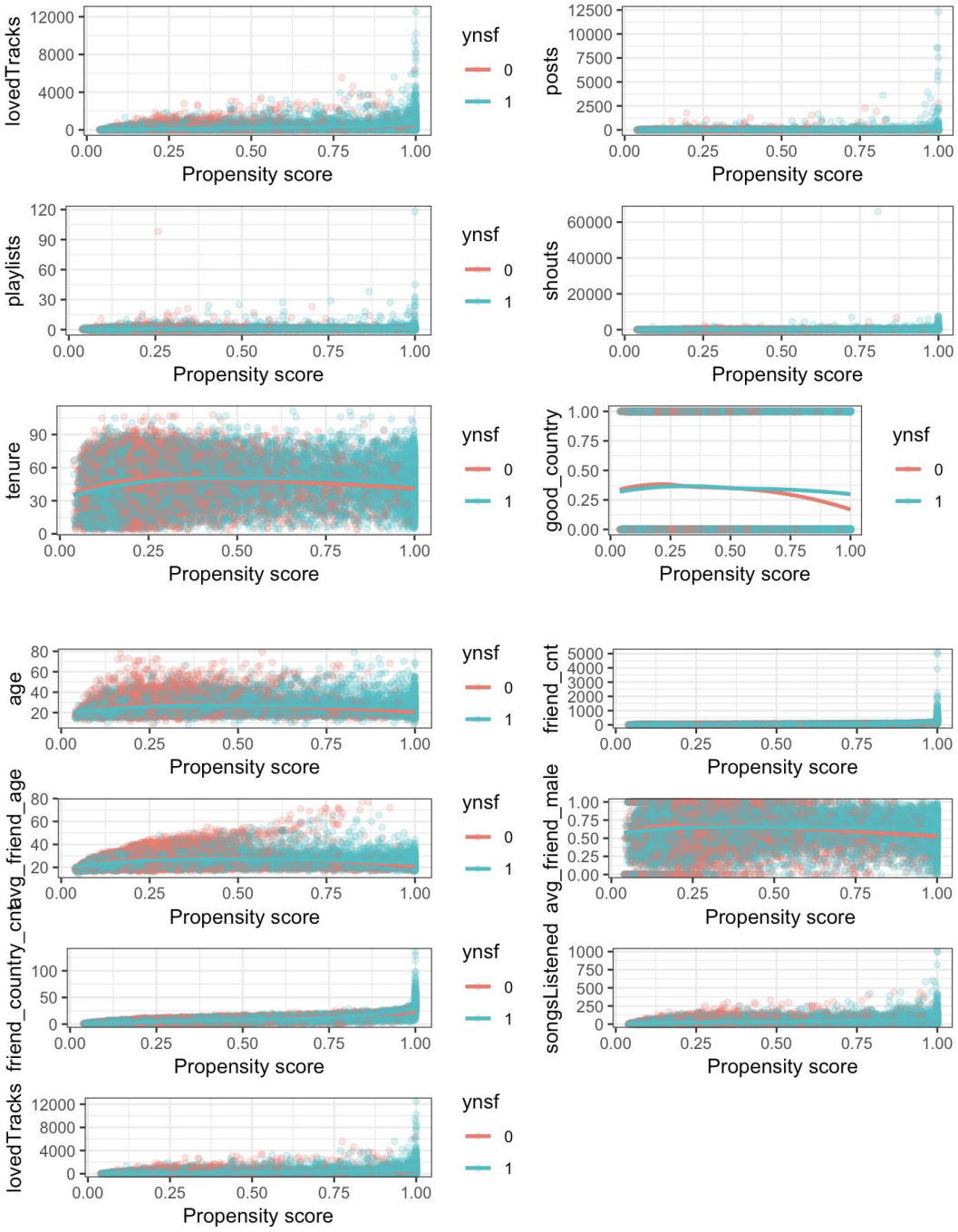
Then I create a dataframe containing only the matched observations: The final dataset is smaller than the original: it contains 19646 observations, meaning that 9823 pairs of treated and control observations were matched.

The final dataset contains a variable called distance, which is the propensity score.

4. Examining Covariate Balance in the Matched Sample

● 4.1: Visual inspection

plot the mean of each covariate against the estimated propensity score, separately by treatment status.



- **4.2: Difference-in-means**

test mean difference for each covariate:

```r
data_m %>%
  group_by(ynsf) %>%
  select(one_of(hn_cov2)) %>%
  summarise_all(funs(mean))

lapply(hn_cov2, function(v) {
  t.test(data_m[, v] ~ data_m$ynsf)
})
```

Estimating treatment effects: Estimating the treatment effect is simple once we have a matched sample that we are happy with. We can use a t-test:

```
        Welch Two Sample t-test

data:  adopter by ynsf
t = -18.938, df = 18060, p-value < 2.2e-16
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.10009352 -0.08131745
sample estimates:
mean in group 0 mean in group 1
     0.08683702      0.17754250
```

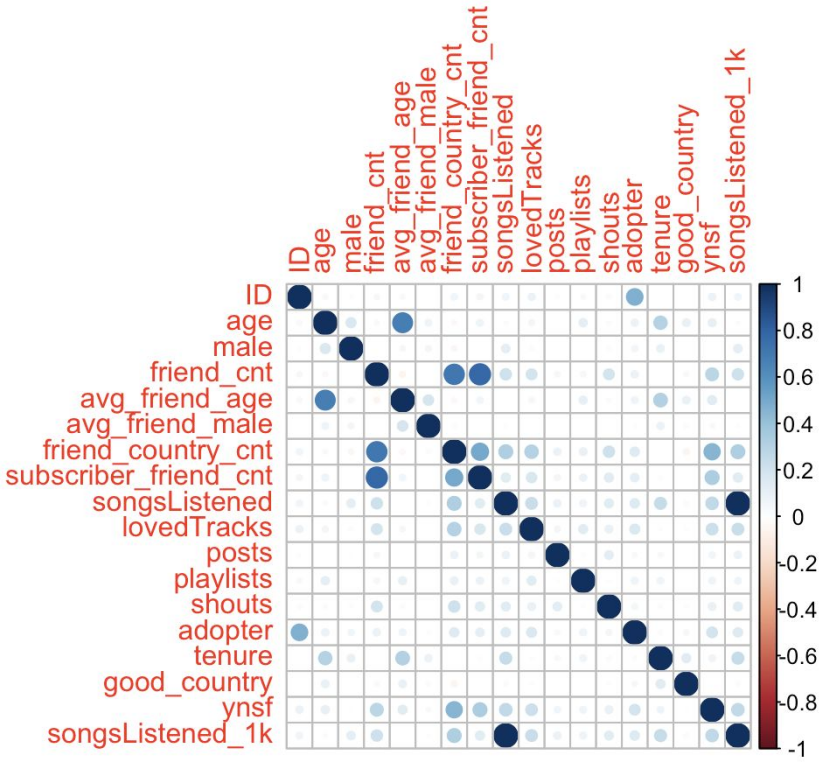
Here for matched data: adopter by ynsf, t = -18.938, comparing to before matching t = -30.961.

After I. eliminate the background variable differences for treatment and control group, (control for the differences). Having subscriber friends has higher probability of being adopter than don't have subscriber friends

Regression Analysis

Now, I will use a logistic regression approach to test which variables (including subscriber friends) are significant for explaining the likelihood of becoming an adopter.

Before fitting into the logistic regression model, let's see the correlation between the predictors.



Based on the analysis, we find that the following varaibles are relatively highly correlated:

age & avg_friend_age ;

male & avg_friend_male;

friend_cnt & friend_country_cnt ;

friend_cnt & subscriber_friend_cnt ;

friend_country_cnt & subscriber_firend_cnt.

In order to build a better regression model, we should not use independent variables which are relatively highly correlated.

Let's see what it shows when putting all the variables into the model.

```
Call:
glm(formula = adopter ~ age + male + friend_cnt + avg_friend_age +
    avg_friend_male + friend_country_cnt + subscriber_friend_cnt +
    lovedTracks + posts + playlists + songsListened_1k + shouts +
    tenure + good_country, family = binomial(), data = Highnote)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-5.3526  -0.4114  -0.3500  -0.2913   2.7018

Coefficients:
                       Estimate Std. Error z value Pr(>|z|)
(Intercept)          -4.179e+00  9.571e-02 -43.665  < 2e-16 ***
age                   1.962e-02  3.478e-03   5.641 1.69e-08 ***
male                  4.133e-01  4.169e-02   9.913  < 2e-16 ***
friend_cnt           -4.312e-03  4.920e-04  -8.765  < 2e-16 ***
avg_friend_age        2.954e-02  4.484e-03   6.588 4.45e-11 ***
avg_friend_male       1.162e-01  6.346e-02   1.831   0.0671 .
friend_country_cnt    4.326e-02  3.616e-03  11.962  < 2e-16 ***
subscriber_friend_cnt 9.132e-02  1.073e-02   8.512  < 2e-16 ***
lovedTracks           6.950e-04  4.933e-05  14.088  < 2e-16 ***
posts                 8.492e-05  9.580e-05   0.886   0.3754
playlists             5.920e-02  1.333e-02   4.441 8.97e-06 ***
songsListened_1k      7.626e-03  5.192e-04  14.687  < 2e-16 ***
shouts                1.108e-04  8.428e-05   1.314   0.1887
tenure               -4.476e-03  1.022e-03  -4.380 1.19e-05 ***
good_country         -4.152e-01  4.078e-02 -10.181  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 24537  on 43826  degrees of freedom
Residual deviance: 22613  on 43812  degrees of freedom
AIC: 22643

Number of Fisher Scoring iterations: 5
```

Multicollinearity can be detected using a statistic called the variance inflation factor (VIF). For any predictor variable, the square root of the VIF indicates the degree to which the confidence interval for that variable's regression parameter is expanded relative to a model with uncorrelated predictors (hence the name). VIF values are pro- vided by the vif() function in the car package. As a general rule, sqrt(vif) > 2 indicates a multicollinearity problem.

```{r}
vif(mod.fit1)
sqrt(vif(mod.fit1)) > 2
outlierTest(mod.fit1)
```

```
                age                male          friend_cnt      avg_friend_age     avg_friend_male
           2.028083            1.061966            4.295009            2.061113            1.042020
friend_country_cnt subscriber_friend_cnt         lovedTracks               posts           playlists
           2.621221            3.007514            1.150339            1.088116            1.044297
  songsListened_1k              shouts              tenure        good_country
           1.280630            1.337860            1.213634            1.029508
                age                male          friend_cnt      avg_friend_age     avg_friend_male
              FALSE               FALSE                TRUE               FALSE               FALSE
friend_country_cnt subscriber_friend_cnt         lovedTracks               posts           playlists
              FALSE               FALSE               FALSE               FALSE               FALSE
  songsListened_1k              shouts              tenure        good_country
              FALSE               FALSE               FALSE               FALSE
     rstudent unadjusted p-value Bonferonni p
32663 -5.837848         5.2879e-09   0.00023175
```

The results indicate that variable friend_cnt has a multicollinearity problem with these predictor variables.

I'll take out this variable to further analyze.

Further, based on the mean analysis graph, logical assumption, and mod.fit1, I choose to include variable the following model:

age, subscriber_friend_cnt, lovedTracks, playlists, songsListened_1k, good_country

```
Call:
glm(formula = adopter ~ age + subscriber_friend_cnt + lovedTracks +
    playlists + songsListened_1k + good_country, family = binomial(),
    data = Highnote)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-7.3540  -0.4065  -0.3553  -0.3124   2.6222

Coefficients:
                      Estimate Std. Error z value Pr(>|z|)
(Intercept)         -3.613e+00  6.537e-02 -55.275  < 2e-16 ***
age                  3.627e-02  2.449e-03  14.815  < 2e-16 ***
subscriber_friend_cnt  9.476e-02  8.250e-03  11.487  < 2e-16 ***
lovedTracks          7.808e-04  4.923e-05  15.859  < 2e-16 ***
playlists            6.589e-02  1.352e-02   4.874 1.09e-06 ***
songsListened_1k     8.306e-03  4.757e-04  17.460  < 2e-16 ***
good_country        -4.408e-01  4.044e-02 -10.902  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 24537  on 43826  degrees of freedom
Residual deviance: 22880  on 43820  degrees of freedom
AIC: 22894

Number of Fisher Scoring iterations: 5
```

```{r}
vif(mod.fit2)
sqrt(vif(mod.fit2)) > 2
outlierTest(mod.fit2)
```

```
              age subscriber_friend_cnt          lovedTracks            playlists     songsListened_1k
         1.041901              1.125002             1.121123             1.038990             1.086087
     good_country
         1.018351
              age subscriber_friend_cnt          lovedTracks            playlists     songsListened_1k
            FALSE                 FALSE                FALSE                FALSE                FALSE
     good_country
            FALSE
      rstudent unadjusted p-value Bonferonni p
32663 -7.781185        7.1848e-15   3.1489e-10
21293 -6.125326        9.0498e-10   3.9663e-05
10623 -4.906967        9.2495e-07   4.0538e-02
```

Note that the model is no longer suffered from multicollinearity problem, but still, we have some outliers, I will delete these outliers from the data set, and do a regression based on the new data set.

```
Call:
glm(formula = adopter ~ age + subscriber_friend_cnt * age + lovedTracks +
    playlists + songsListened_1k + good_country, family = binomial(),
    data = HighnoteNew)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-4.0100  -0.4036  -0.3467  -0.3041   2.6563

Coefficients:
                             Estimate Std. Error z value Pr(>|z|)
(Intercept)                -3.813e+00  6.907e-02 -55.207  < 2e-16 ***
age                         4.248e-02  2.551e-03  16.655  < 2e-16 ***
subscriber_friend_cnt       3.508e-01  2.523e-02  13.904  < 2e-16 ***
lovedTracks                 7.822e-04  4.975e-05  15.723  < 2e-16 ***
playlists                   6.982e-02  1.364e-02   5.118  3.1e-07 ***
songsListened_1k            7.480e-03  4.804e-04  15.569  < 2e-16 ***
good_country               -4.267e-01  4.060e-02 -10.510  < 2e-16 ***
age:subscriber_friend_cnt -6.920e-03  7.786e-04  -8.888  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 24535  on 43817  degrees of freedom
Residual deviance: 22580  on 43810  degrees of freedom
AIC: 22596

Number of Fisher Scoring iterations: 5
```

The AIC changed from 22894 to 22596, indicating it's a better model.

The expected variance for data drawn from a binomial distribution is $\sigma 2 = n\pi(1 - \pi)$, where n is the number of observations and π is the probability of belonging to the Y = 1 group. Overdispersion occurs when the observed variance of the response variable is larger than what would be expected from a binomial distribution. Overdispersion can lead to distorted test standard errors and inaccurate tests of significance.

We can also test if there is an overdispersion problem with the model using the following code:

```
```{r}
deviance(mod.fit3)/df.residual(mod.fit3)
```
```

```
[1] 0.5153999
```

With logistic regression, overdispersion is suggested if the ratio of the residual deviance to the residual degrees of freedom is much larger than 1, which is not our case here.

By looking at p-value, all the variables, including the intercept, are significant with p-value less than 0.01.

Let's look at the regression coefficients:

```{r}
coef(mod.fit3)

```

(Intercept)	age	subscriber_friend_cnt	lovedTracks
-3.8133120779	0.0424823372	0.3507614447	0.0007821799
playlists	songsListened_1k	good_country	age:subscriber_friend_cnt
0.0698206735	0.0074795246	-0.4266934107	-0.0069202213

In a logistic regression, the response being modeled is the log(odds) that Y = 1. The regression coefficients give the change in log(odds) in the response for a unit change in the predictor variable, holding all other predictor variables constant.

Because log(odds) are difficult to interpret, we can exponentiate them to put the results on an odds scale:

```{r}
exp(coef(mod.fit3))

```

(Intercept)	age	subscriber_friend_cnt	lovedTracks
0.02207494	1.04339763	1.42014850	1.00078249
playlists	songsListened_1k	good_country	age:subscriber_friend_cnt
1.07231587	1.00750757	0.65266362	0.99310367

Now we can see that the odds of a fee-user conversion are increased by a factor of 1.00078249 for a one-unit increase in 'lovedTracks', (holding 'subscriber_friend_cnt', 'lovedTracks', 'playlists', 'songsListened_1k', 'good_country' constant). Conversely, the odds of a fee-user conversion are multiplied by a factor of 0.0007821799 for a one-unit increase in 'lovedTracks'.

The odds of a fee-user conversion increase with 'age', 'subscriber_friend_cnt', 'lovedTracks', 'playlists', 'songsListened_1k', and decrease with 'good_country', 'age:subscriber_friend_cnt'

A negative interaction coefficient in 'age:subscriber_friend_cnt' means that the effect of the combined action of two predictors is less than the sum of the individual effects.

Because the predictor variables can't equal 0, the intercept isn't meaningful in this case.

TAKEAWAYS

From the above analysis. The results inform a "free-to-fee" strategy for High Note as follows:

When the company trying to put money in converting fee-users, try to:

- Targeting users in middle or late 20's.
- Targeting users with higher user engagement, (loved more tracks, made more playlists, listened more songs, however, posts made and shouts received are not necessarily important).
- Targeting users have (more) subscriber friend since there is peer influence exist.
- Targeting more on users that from countires other than US, UK or Germany.

For more details of this project, please find here