High Note data analysis

Yuzi Liu

11/19/2018

```
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)
library(stargazer)
```

```
##
## Please cite as:
```

```
##  Hlavac, Marek (2018). stargazer: Well-Formatted Regression and Summary
Statistics Tables.
```

```
##  R package version 5.2.2. https://CRAN.R-project.org/package=stargazer
```

```
library(MatchIt)
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
library(car)
```

```
## Loading required package: carData
```

```
##
## Attaching package: 'car'
```

```
## The following object is masked from 'package:dplyr':
##
##     recode
```

```
library(corrplot)

## corrplot 0.84 loaded

library(knitr)
opts_chunk$set(echo = TRUE)
```

Summary Statistics

select all adopter assign to a new data frame “premium” select all non-adopter assign to a new data frame “free” deleting ID from adopter and non adopter since it’s not a key variable

```
getwd()

## [1] "/Users/yuziliu/Downloads"

Highnote <- read.csv("HighNote Data Midterm.csv", header = TRUE)
premium <- subset(Highnote, adopter == 1)
free <- subset(Highnote, adopter == 0)

premium$ID <- NULL
free$ID <- NULL
```

Generate descriptive statistics for the key variables in the data set summary the discriptive statistics of adopter and non-adopter

```
stargazer(premium, type="text", median=TRUE, digits=2, title="adopters
summary")

##
## 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
## ----------------------------------------------------------------------
-----------
```

```
stargazer(free, type="text", median=TRUE, digits=2, title="Non-adopters
summary")
```

```
##
## 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
## ----------------------------------------------------------------------------
--------------
```

do t-tests to compare difference in mean values of the variables for adopter and non-adopter

```
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)))

## Adding missing grouping variables: `adopter`

## # A tibble: 2 x 15
##   adopter   age  male friend_cnt avg_friend_age avg_friend_male
##     <int> <dbl> <dbl>      <dbl>          <dbl>           <dbl>
## 1       0  23.9 0.622       18.5           24.0           0.617
## 2       1  26.0 0.729       39.7           25.4           0.637
## # ... with 9 more variables: friend_country_cnt <dbl>,
## #   subscriber_friend_cnt <dbl>, songsListened <dbl>, lovedTracks <dbl>,
## #   posts <dbl>, playlists <dbl>, shouts <dbl>, tenure <dbl>,
## #   good_country <dbl>

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

## [[1]]
##
##  Welch Two Sample t-test
##
## data:  Highnote[, v] by Highnote$adopter
```

```
## t = -16.996, df = 4079.3, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -2.265768 -1.797097
## sample estimates:
## mean in group 0 mean in group 1
##        23.94844        25.97987
##
##
## [[2]]
##
##  Welch Two Sample t-test
##
## data:  Highnote[, v] by Highnote$adopter
## t = -13.654, df = 4295, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.12278707 -0.09195413
## sample estimates:
## mean in group 0 mean in group 1
##       0.6218610       0.7292316
##
##
## [[3]]
##
##  Welch Two Sample t-test
##
## data:  Highnote[, v] by Highnote$adopter
## t = -10.646, df = 3675.7, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -25.15422 -17.32999
## sample estimates:
## mean in group 0 mean in group 1
##        18.49166        39.73377
##
##
## [[4]]
##
##  Welch Two Sample t-test
##
## data:  Highnote[, v] by Highnote$adopter
## t = -15.658, df = 4140.9, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -1.608931 -1.250852
## sample estimates:
## mean in group 0 mean in group 1
##        24.01142        25.44131
##
```

```
##
## [[5]]
##
##  Welch Two Sample t-test
##
## data:  Highnote[, v] by Highnote$adopter
## t = -4.4426, df = 4591.6, p-value = 9.097e-06
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.02883955 -0.01117951
## sample estimates:
## mean in group 0 mean in group 1
##       0.6165888       0.6365983
##
##
## [[6]]
##
##  Welch Two Sample t-test
##
## data:  Highnote[, v] by Highnote$adopter
## t = -21.267, df = 3791.6, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -3.528795 -2.933081
## sample estimates:
## mean in group 0 mean in group 1
##        3.957891        7.188829
##
##
## [[7]]
##
##  Welch Two Sample t-test
##
## data:  Highnote[, v] by Highnote$adopter
## t = -12.287, df = 3632.2, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -1.413899 -1.024766
## sample estimates:
## mean in group 0 mean in group 1
##        0.417469        1.636802
##
##
## [[8]]
##
##  Welch Two Sample t-test
##
## data:  Highnote[, v] by Highnote$adopter
## t = -21.629, df = 3792.7, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
```

```
## 95 percent confidence interval:
##  -17634.24 -14702.96
## sample estimates:
## mean in group 0 mean in group 1
##        17589.44        33758.04
##
##
## [[9]]
##
##  Welch Two Sample t-test
##
## data:  Highnote[, v] by Highnote$adopter
## t = -21.188, df = 3705.6, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -193.9447 -161.0917
## sample estimates:
## mean in group 0 mean in group 1
##        86.82263       264.34080
##
##
## [[10]]
##
##  Welch Two Sample t-test
##
## data:  Highnote[, v] by Highnote$adopter
## t = -4.2151, df = 3663.5, p-value = 2.557e-05
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -23.30665  -8.50825
## sample estimates:
## mean in group 0 mean in group 1
##        5.293002       21.200454
##
##
## [[11]]
##
##  Welch Two Sample t-test
##
## data:  Highnote[, v] by Highnote$adopter
## t = -8.0816, df = 3634.7, p-value = 8.619e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.4367565 -0.2662138
## sample estimates:
## mean in group 0 mean in group 1
##       0.5492804       0.9007655
##
##
## [[12]]
```

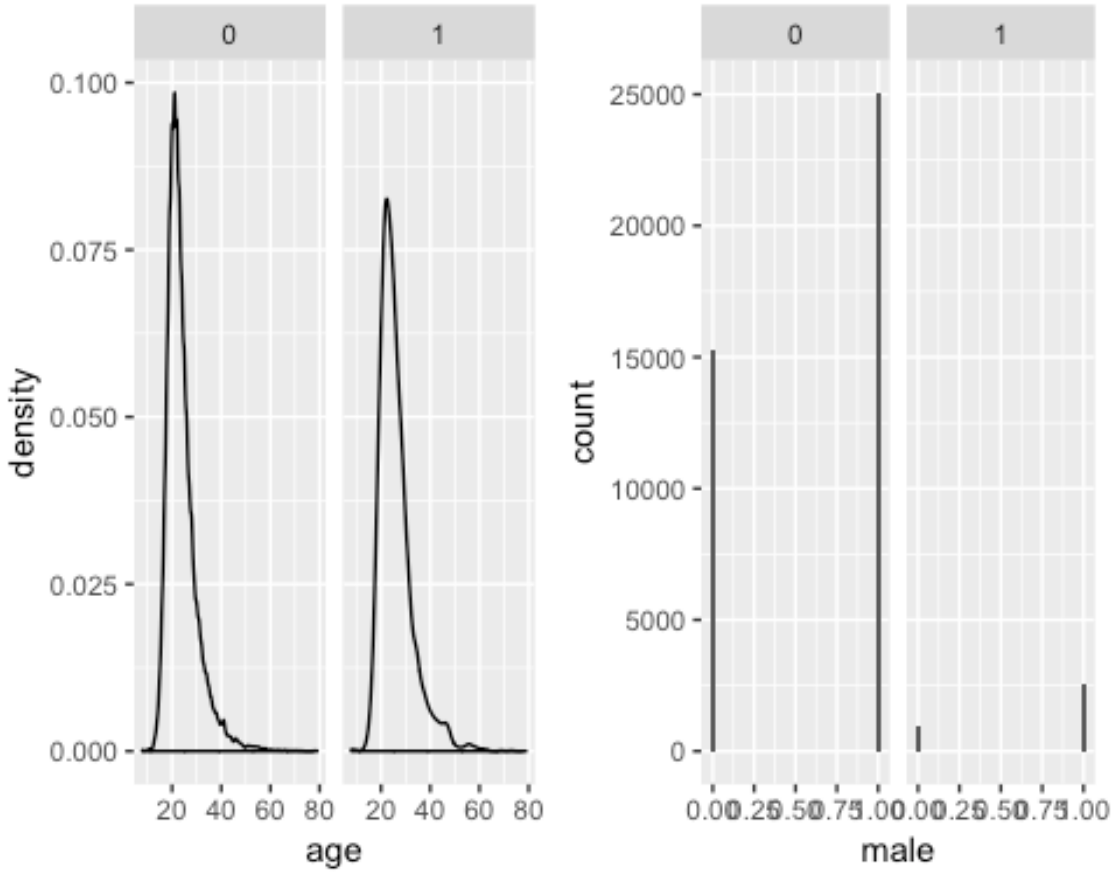
```
##
##  Welch Two Sample t-test
##
## data:  Highnote[, v] by Highnote$adopter
## t = -3.5659, df = 3536.5, p-value = 0.0003674
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -107.66170  -31.27249
## sample estimates:
## mean in group 0 mean in group 1
##        29.97266        99.43975
##
##
## [[13]]
##
##  Welch Two Sample t-test
##
## data:  Highnote[, v] by Highnote$adopter
## t = -5.0434, df = 4150.6, p-value = 4.768e-07
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -2.462620 -1.083959
## sample estimates:
## mean in group 0 mean in group 1
##        43.80993        45.58322
##
##
## [[14]]
##
##  Welch Two Sample t-test
##
## data:  Highnote[, v] by Highnote$adopter
## t = 8.8009, df = 4248.5, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.05463587 0.08595434
## sample estimates:
## mean in group 0 mean in group 1
##       0.3577916       0.2874965
```

We can see that the mean difference of all covariates are significant. From these comparisons, we can make a tenative 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.

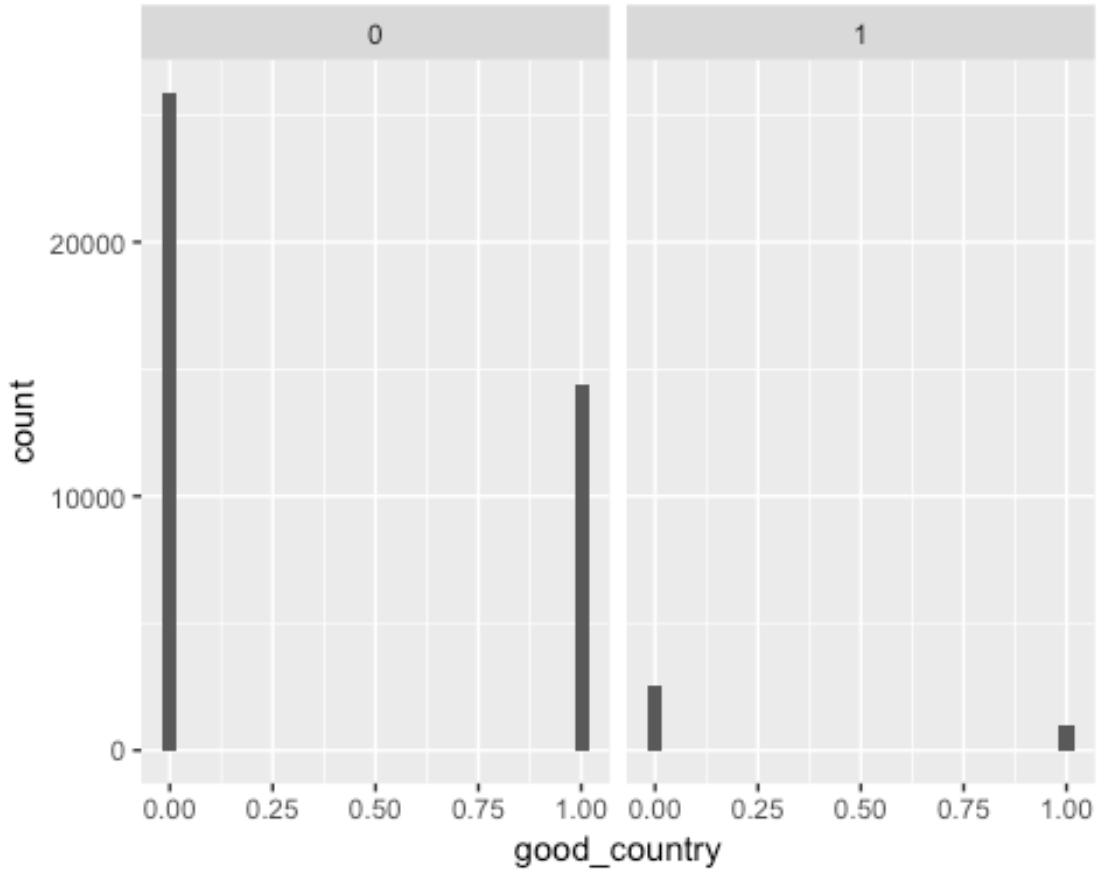
Data Visualization

(i) Demographics

```
plot1 <- ggplot(data = Highnote,
       mapping = aes(x = age)) +
  geom_density() +
  facet_wrap(~ adopter)

plot2 <-  ggplot(data = Highnote,
       mapping = aes(x = male)) +
  geom_histogram() +
  facet_wrap(~ adopter)

grid.arrange(plot1, plot2, ncol=2)

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

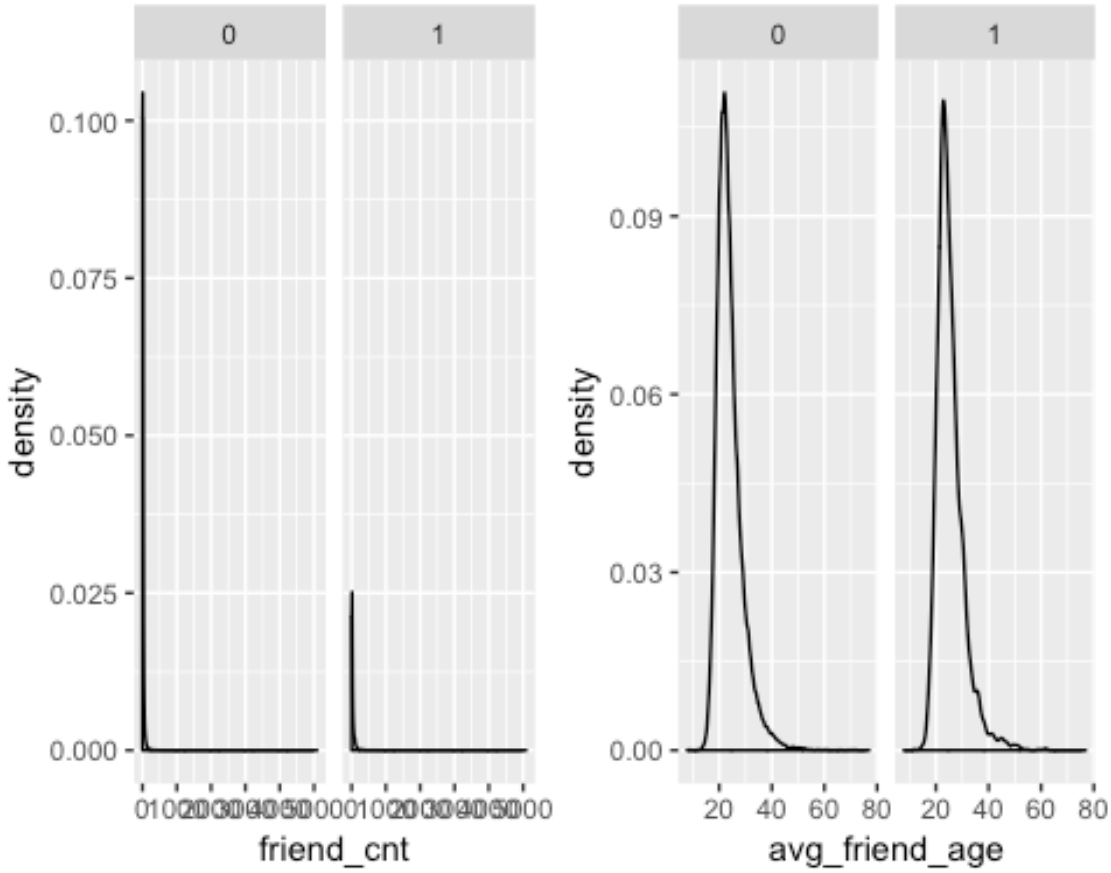



```
ggplot(data = Highnote,
       mapping = aes(x = good_country)) +
  geom_histogram() +
  facet_wrap(~ adopter)

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

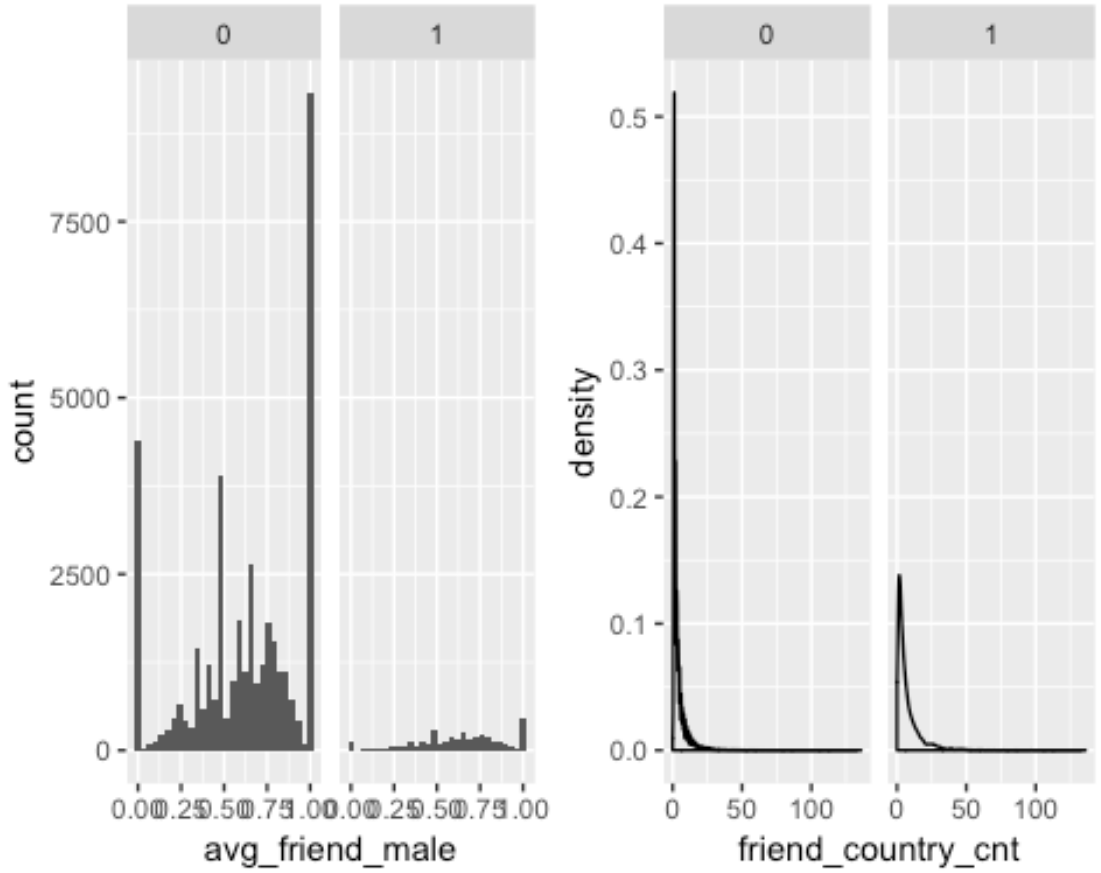


(ii) Peer Influence

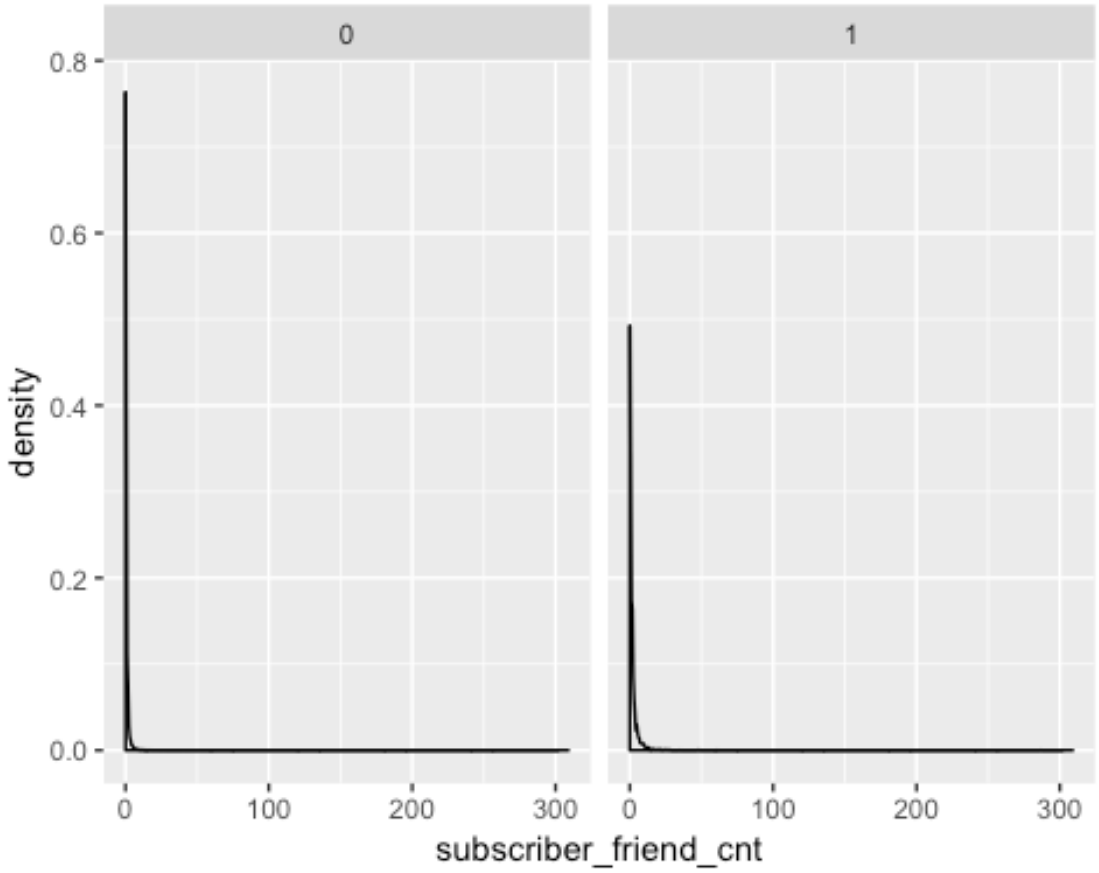
```
plot3 <- ggplot(data = Highnote,
       mapping = aes(x = friend_cnt)) +
  geom_density() +
  facet_wrap(~ adopter)

plot4 <- ggplot(data = Highnote,
       mapping = aes(x = avg_friend_age)) +
  geom_density() +
  facet_wrap(~ adopter)

grid.arrange(plot3, plot4, ncol=2)
```




```
plot5 <- ggplot(data = Highnote,
       mapping = aes(x = avg_friend_male)) +
  geom_histogram() +
  facet_wrap(~ adopter)

plot6 <- ggplot(data = Highnote,
       mapping = aes(x = friend_country_cnt)) +
  geom_density() +
  facet_wrap(~ adopter)

grid.arrange(plot5, plot6, ncol=2)

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

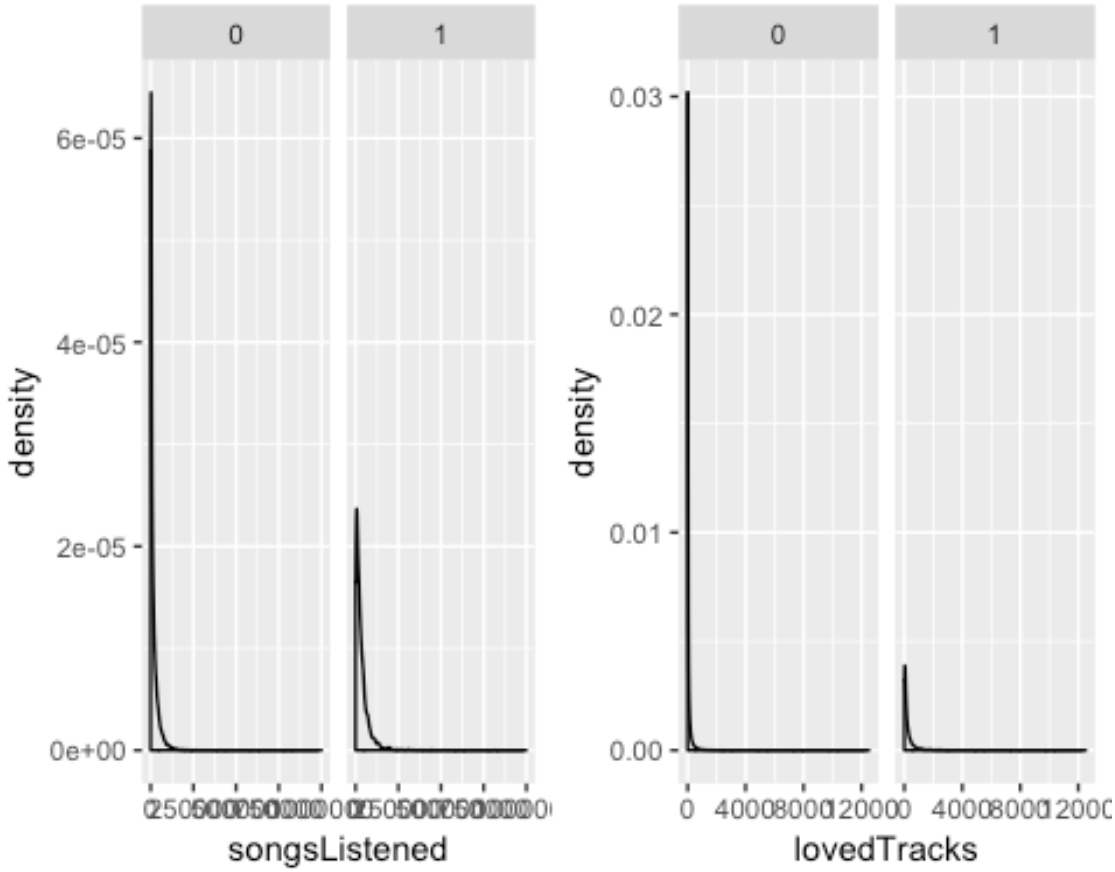



```
ggplot(data = Highnote,
       mapping = aes(x = subscriber_friend_cnt)) +
  geom_density() +
  facet_wrap(~ adopter)
```

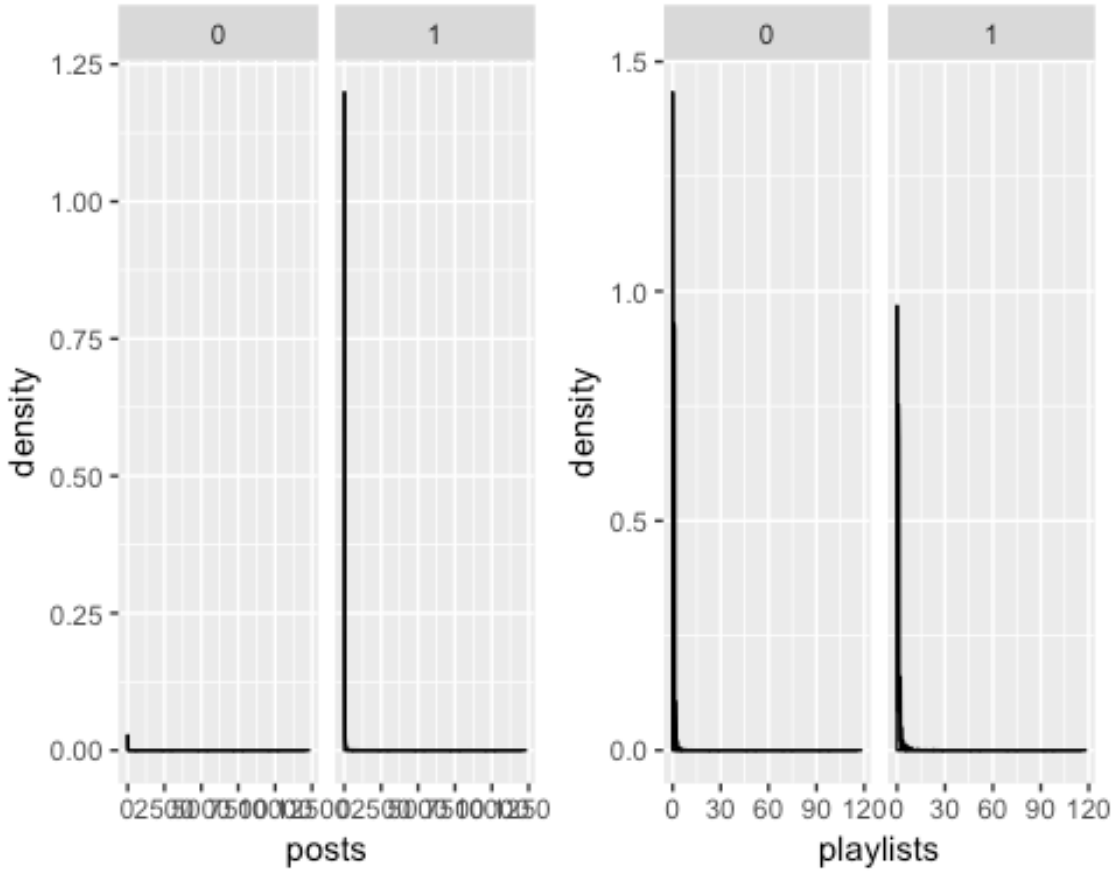


(iii) User Engagement

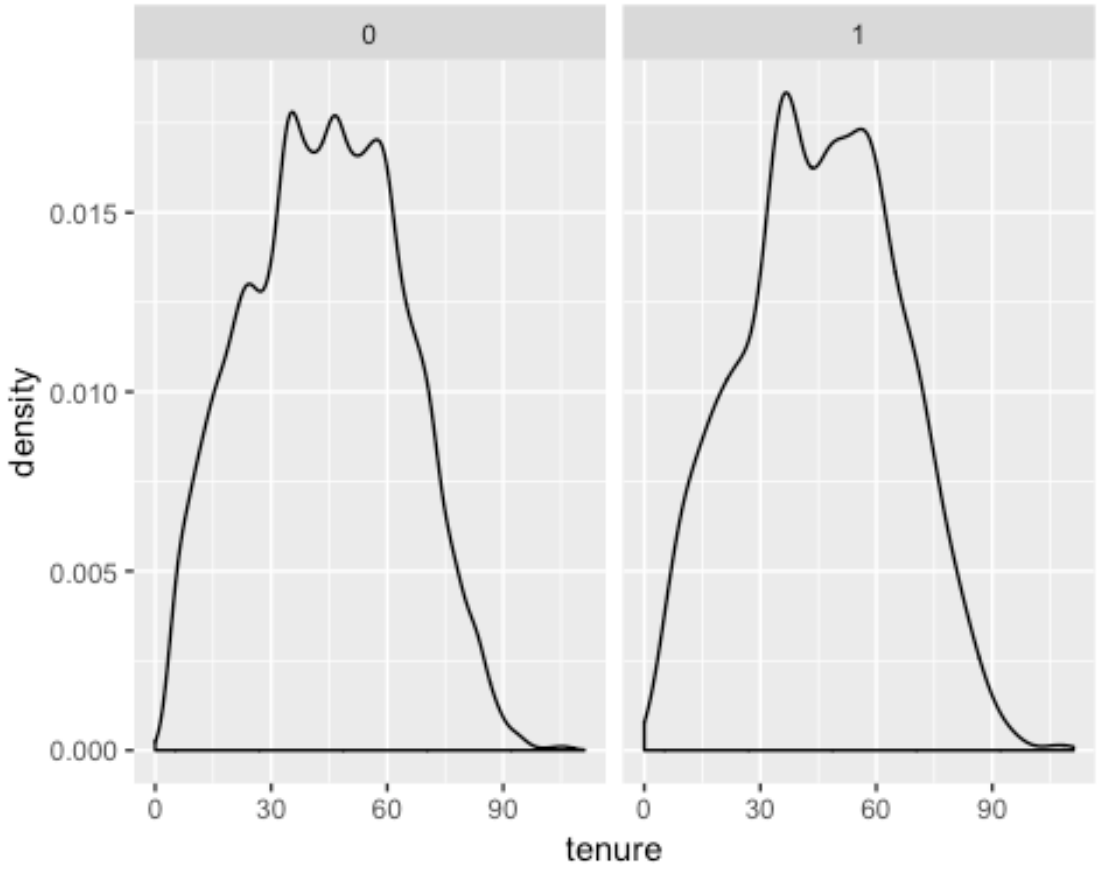
```
plot7 <- ggplot(data = Highnote,
       mapping = aes(x = songsListened)) +
  geom_density() +
  facet_wrap(~ adopter)

plot8 <- ggplot(data = Highnote,
       mapping = aes(x = lovedTracks)) +
  geom_density() +
  facet_wrap(~ adopter)

grid.arrange(plot7, plot8, ncol=2)
```




```
plot9 <- ggplot(data = Highnote,
       mapping = aes(x = posts)) +
  geom_density() +
  facet_wrap(~ adopter)

plot10 <- ggplot(data = Highnote,
       mapping = aes(x = playlists)) +
  geom_density() +
  facet_wrap(~ adopter)

grid.arrange(plot9, plot10, ncol=2)
```




```
ggplot(data = Highnote,
       mapping = aes(x = tenure)) +
  geom_density() +
  facet_wrap(~ adopter)
```



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

create treatment and control groups * “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)

```
Highnote$ynsf = ifelse(Highnote$subscriber_friend_cnt >= 1, 1, 0)
```

1. Pre-analysis using 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)

```
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:

```
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)))
```

```
## Adding missing grouping variables: `ynsf`
```

```
## # A tibble: 2 x 14
##    ynsf   age  male friend_cnt avg_friend_age avg_friend_male
##   <dbl> <dbl> <dbl>      <dbl>          <dbl>           <dbl>
## 1     0  23.7 0.629       10.4           23.8           0.613
## 2     1  25.4 0.636       54.0           25.4           0.636
## # ... with 8 more variables: friend_country_cnt <dbl>,
## #   songsListened <dbl>, lovedTracks <dbl>, posts <dbl>, playlists <dbl>,
## #   shouts <dbl>, tenure <dbl>, good_country <dbl>
```

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

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

```
## [[1]]
##
##  Welch Two Sample t-test
##
## data:  Highnote[, v] by Highnote[, "ynsf"]
## t = -20.841, df = 14645, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -1.778544 -1.472749
## sample estimates:
## mean in group 0 mean in group 1
##        23.74756        25.37321
##
##
## [[2]]
##
##  Welch Two Sample t-test
##
## data:  Highnote[, v] by Highnote[, "ynsf"]
## t = -1.3459, df = 15986, p-value = 0.1784
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.018236129  0.003388028
## sample estimates:
## mean in group 0 mean in group 1
##       0.6288378       0.6362618
##
##
## [[3]]
##
##  Welch Two Sample t-test
##
## data:  Highnote[, v] by Highnote[, "ynsf"]
## t = -33.707, df = 9903.1, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -46.12459 -41.05469
## sample estimates:
## mean in group 0 mean in group 1
##        10.43133        54.02097
##
##
## [[4]]
##
##  Welch Two Sample t-test
##
## data:  Highnote[, v] by Highnote[, "ynsf"]
## t = -27.658, df = 15667, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
```

```
##  -1.744514 -1.513611
## sample estimates:
## mean in group 0 mean in group 1
##        23.76137        25.39043
##
##
## [[5]]
##
##  Welch Two Sample t-test
##
## data:  Highnote[, v] by Highnote[, "ynsf"]
## t = -7.7114, df = 23020, p-value = 1.294e-14
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.02846397 -0.01692672
## sample estimates:
## mean in group 0 mean in group 1
##       0.6131124       0.6358077
##
##
## [[6]]
##
##  Welch Two Sample t-test
##
## data:  Highnote[, v] by Highnote[, "ynsf"]
## t = -65.05, df = 10372, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -6.861271 -6.459857
## sample estimates:
## mean in group 0 mean in group 1
##        2.725062        9.385626
##
##
## [[7]]
##
##  Welch Two Sample t-test
##
## data:  Highnote[, v] by Highnote[, "ynsf"]
## t = -41.505, df = 11447, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -20037.04 -18229.80
## sample estimates:
## mean in group 0 mean in group 1
##        14602.22        33735.64
##
##
## [[8]]
##
```

```
##  Welch Two Sample t-test
##
## data:  Highnote[, v] by Highnote[, "ynsf"]
## t = -31.265, df = 10585, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -170.1918 -150.1102
## sample estimates:
## mean in group 0 mean in group 1
##       65.21365       225.36465
##
##
## [[9]]
##
##  Welch Two Sample t-test
##
## data:  Highnote[, v] by Highnote[, "ynsf"]
## t = -7.3649, df = 9933.6, p-value = 1.914e-13
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -22.76492 -13.19424
## sample estimates:
## mean in group 0 mean in group 1
##       2.543377       20.522956
##
##
## [[10]]
##
##  Welch Two Sample t-test
##
## data:  Highnote[, v] by Highnote[, "ynsf"]
## t = -10.492, df = 11238, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.2546958 -0.1745100
## sample estimates:
## mean in group 0 mean in group 1
##       0.5294671       0.7440700
##
##
## [[11]]
##
##  Welch Two Sample t-test
##
## data:  Highnote[, v] by Highnote[, "ynsf"]
## t = -11.426, df = 9888.1, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -100.04703  -70.74591
## sample estimates:
```

```
## mean in group 0 mean in group 1
##        16.42304       101.81951
##
##
## [[12]]
##
##  Welch Two Sample t-test
##
## data:  Highnote[, v] by Highnote[, "ynsf"]
## t = -14.696, df = 15805, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -3.792309 -2.899752
## sample estimates:
## mean in group 0 mean in group 1
##        43.20268        46.54871
##
##
## [[13]]
##
##  Welch Two Sample t-test
##
## data:  Highnote[, v] by Highnote[, "ynsf"]
## t = 2.0956, df = 16030, p-value = 0.03613
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.0007383591 0.0220968020
## sample estimates:
## mean in group 0 mean in group 1
##       0.3546936       0.3432760
```

We see that except for 'male', all mean value of other variables are statistically distinguishable. We should then 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.

```
Highnote <- Highnote %>% mutate(songsListened_1k = songsListened / 1000)

h_ps <- glm(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)

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
summary(h_ps)
```

```
##
## 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.

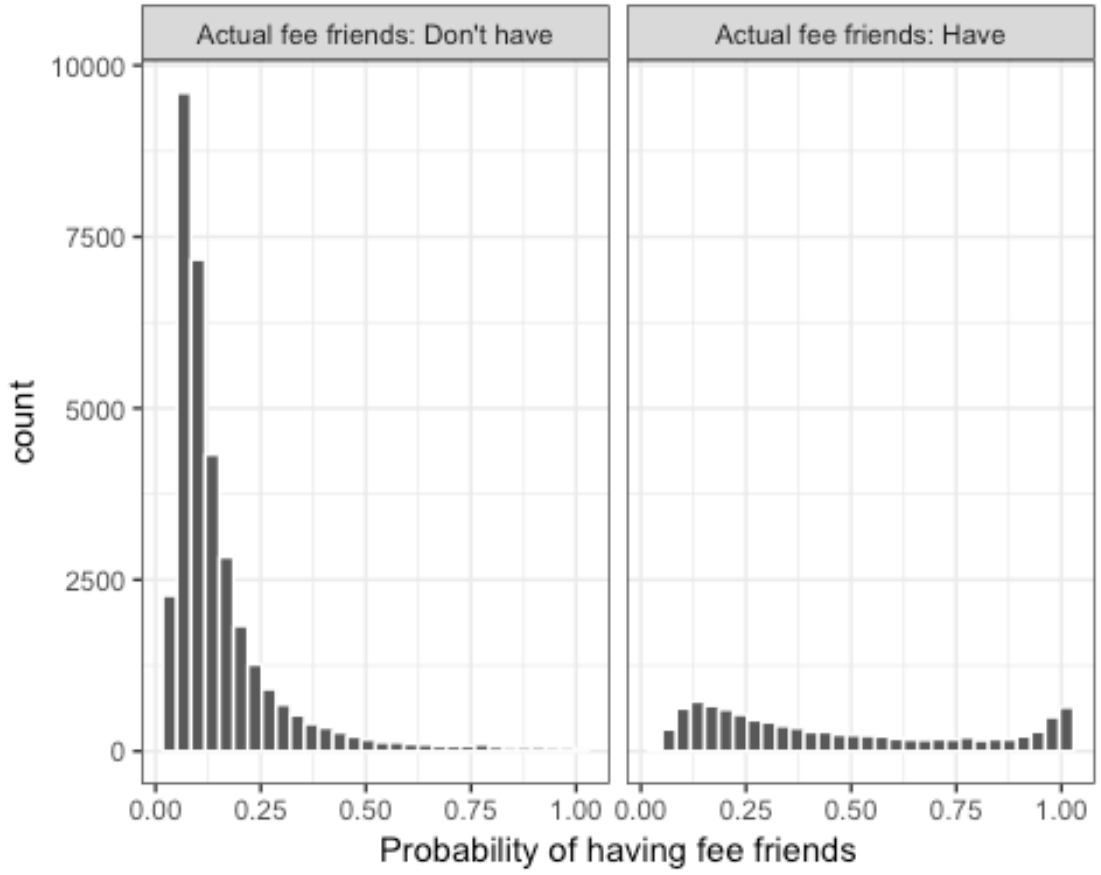
```
prs_df <- data.frame(pr_score = predict(h_ps, type = "response"),
                     ynsf = h_ps$model$ynsf)
head(prs_df)
```

```
##     pr_score ynsf
## 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:

```
labs <- paste("Actual fee friends:", c("Have", "Don't have"))
prs_df %>%
  mutate(ynsf = ifelse(ynsf == 1, labs[1], labs[2])) %>%
  ggplot(aes(x = pr_score)) +
  geom_histogram(color = "white") +
  facet_wrap(~ynsf) +
  xlab("Probability of having fee friends") +
  theme_bw()

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



3. Executing a matching algorithm

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

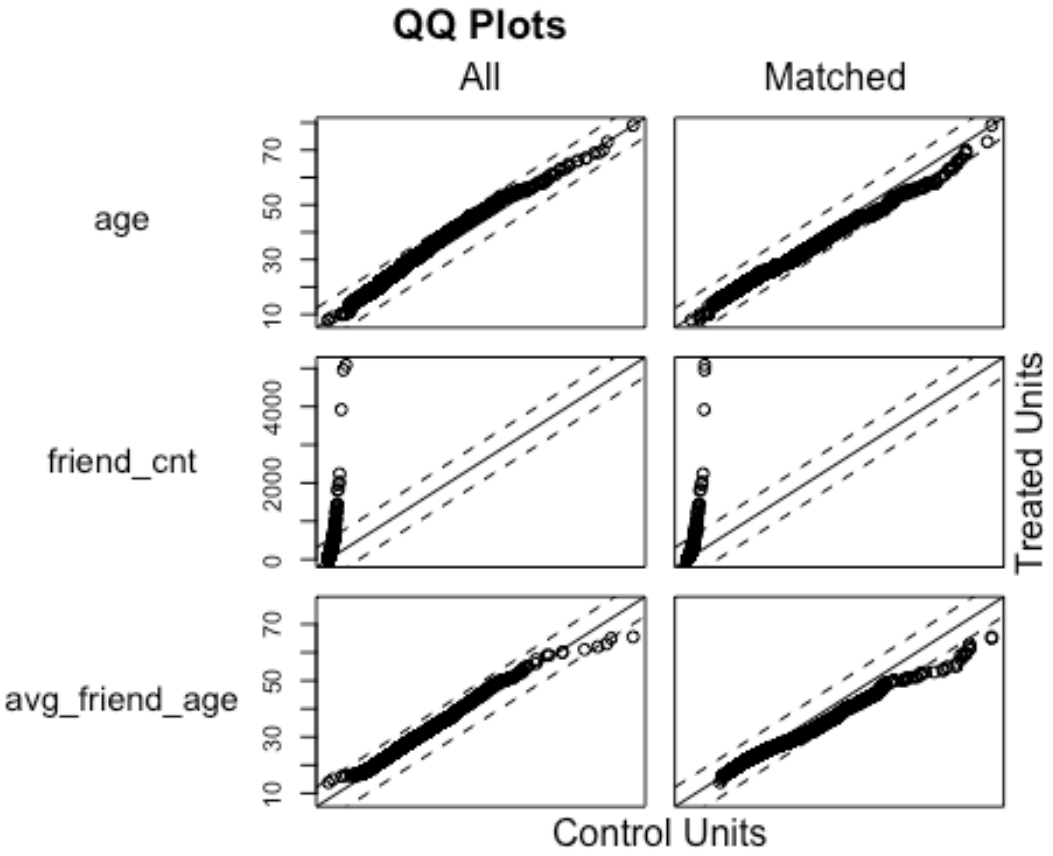
```
Highnote_nomiss <- Highnote %>%  # MatchIt does not allow missing values
  select(adopter, ynsf, one_of(hn_cov2)) %>%
  na.omit()

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)

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

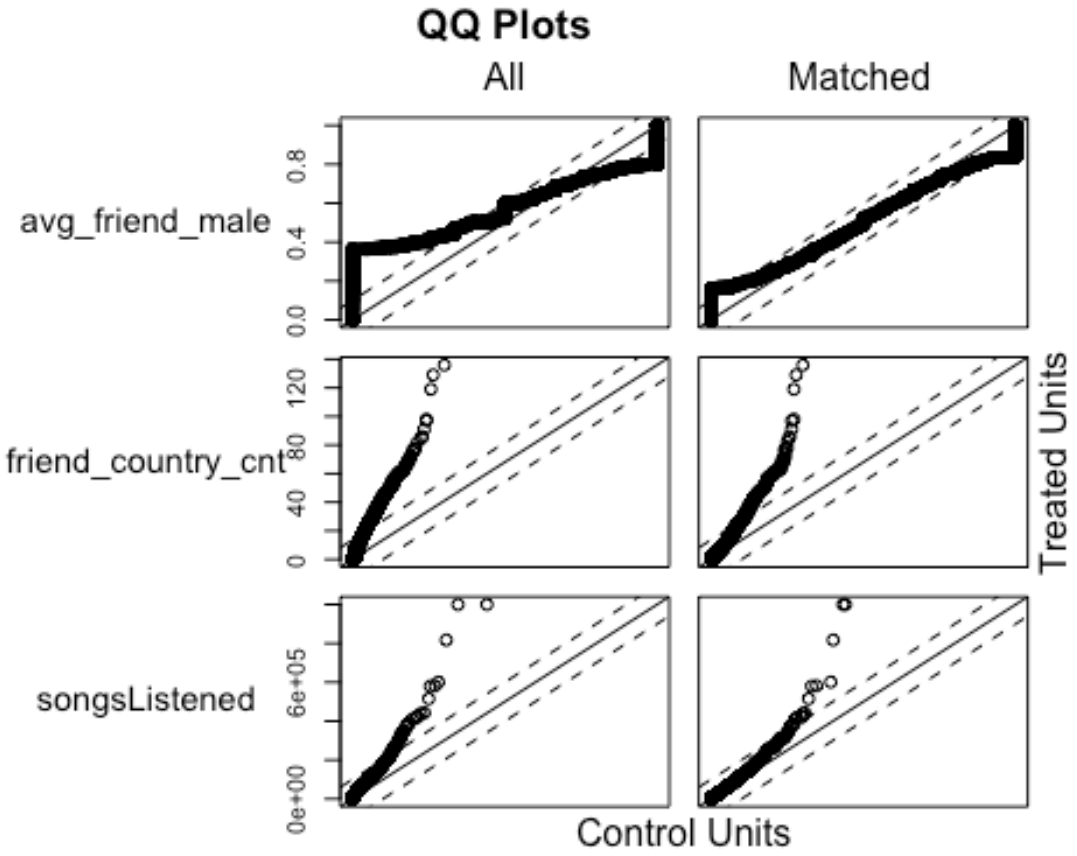
get some information about how successful the matching was:

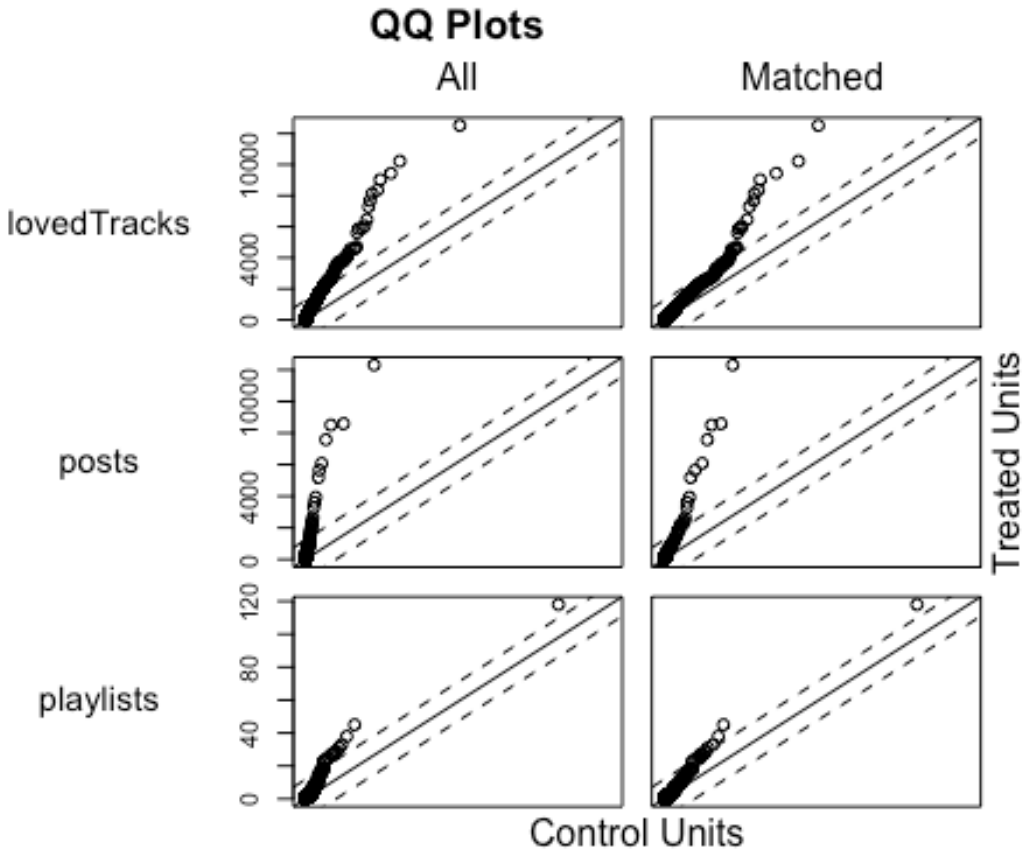
```
summary(mod_match)

##
## Call:
## matchit(formula = ynsf ~ age + friend_cnt + avg_friend_age +
##     avg_friend_male + friend_country_cnt + songsListened + lovedTracks +
##     posts + playlists + shouts + tenure + good_country, data =
Highnote_nomiss,
##     method = "nearest")
##
## Summary of balance for all data:
##                      Means Treated Means Control SD Control  Mean Diff
## distance                    0.4635        0.1550     0.1436     0.3085
## age                        25.3732       23.7476     6.2245     1.6256
## friend_cnt                 54.0210       10.4313    15.2769    43.5896
## avg_friend_age             25.3904       23.7614     5.0577     1.6291
## avg_friend_male             0.6358        0.6131     0.3343     0.0227
## friend_country_cnt          9.3856        2.7251     3.1024     6.6606
## songsListened           33735.6404    14602.2205 23214.2898 19133.4199
## lovedTracks               225.3647       65.2137   181.4812   160.1510
## posts                      20.5230        2.5434    33.7947    17.9796
## playlists                   0.7441        0.5295     0.9673     0.2146
## shouts                    101.8195       16.4230    79.7381    85.3965
## tenure                     46.5487       43.2027    19.7212     3.3460
## good_country                0.3433        0.3547     0.4784    -0.0114
##                         eQQ Med   eQQ Mean     eQQ Max
## distance                 0.2510     0.3085      0.6843
## age                      1.0000     1.6296      5.0000
## friend_cnt              22.0000    43.5838   4794.0000
## avg_friend_age           1.5909     1.6369     11.5000
## avg_friend_male          0.0738     0.0958      0.3636
## friend_country_cnt       5.0000     6.6598     95.0000
## songsListened        15471.0000 19126.1623 653702.0000
## lovedTracks             65.0000   159.9562   6343.0000
## posts                    0.0000    17.8829   9535.0000
## playlists                0.0000     0.2092     26.0000
## shouts                  15.0000    85.1764  59168.0000
```

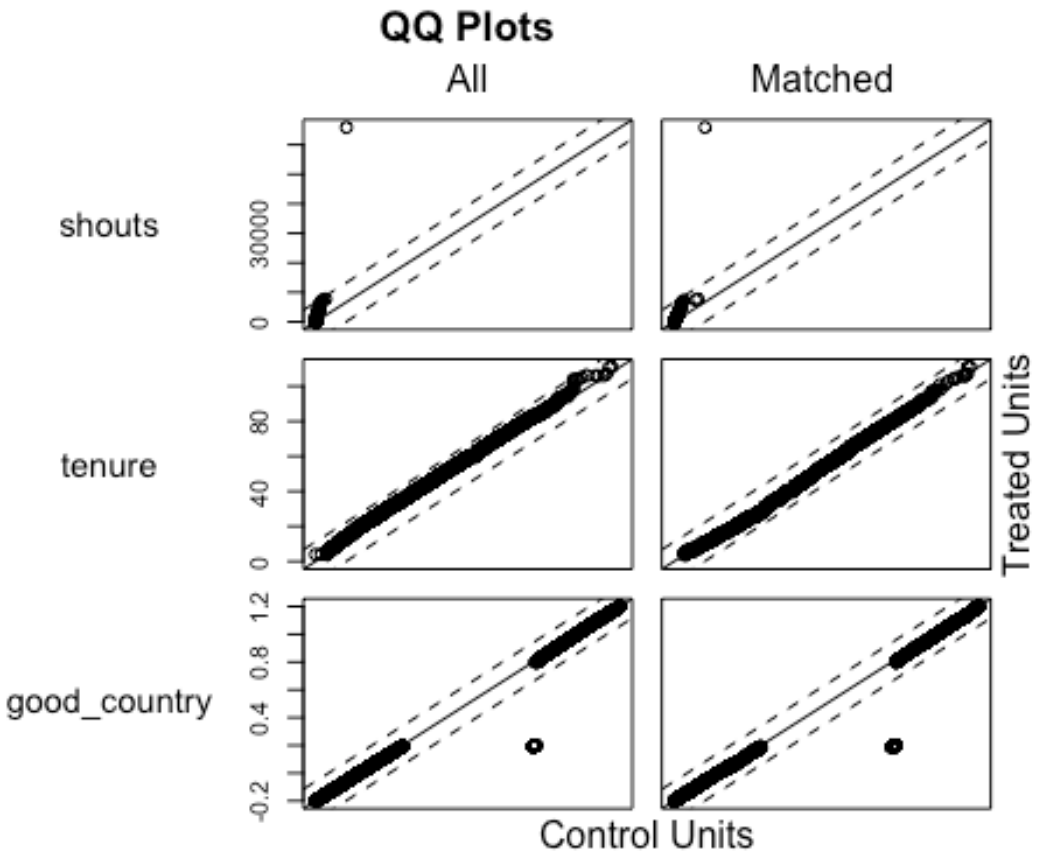
```
## tenure                  3.0000     3.3473     10.0000
## good_country            0.0000     0.0114      1.0000
##
##
## Summary of balance for matched data:
##                     Means Treated Means Control SD Control Mean Diff
## distance                   0.4635        0.3040     0.1914    0.1595
## age                       25.3732       26.4180     8.0005   -1.0448
## friend_cnt                54.0210       21.4052    23.5586   32.6158
## avg_friend_age            25.3904       26.5864     6.7140   -1.1960
## avg_friend_male            0.6358        0.6557     0.2647   -0.0199
## friend_country_cnt         9.3856        5.0768     4.6543    4.3089
## songsListened          33735.6404    27219.9089 33842.4008 6515.7315
## lovedTracks              225.3647      134.9342   299.0241   90.4304
## posts                     20.5230        6.2695    60.7689   14.2535
## playlists                  0.7441        0.6720     1.3948    0.0721
## shouts                   101.8195       37.2356   138.7147   64.5839
## tenure                    46.5487       47.6901    19.0755   -1.1414
## good_country               0.3433        0.3615     0.4805   -0.0182
##                       eQQ Med  eQQ Mean     eQQ Max
## distance               0.1087    0.1595      0.4520
## age                    1.0000    1.0448      8.0000
## friend_cnt            12.0000   32.6158   4794.0000
## avg_friend_age         0.5000    1.2839     14.0000
## avg_friend_male        0.0147    0.0329      0.1642
## friend_country_cnt     2.0000    4.3089     95.0000
## songsListened       4904.0000 6515.7315 566867.0000
## lovedTracks           38.0000   90.4304   6180.0000
## posts                  0.0000   14.2535   9535.0000
## playlists              0.0000    0.1106     22.0000
## shouts                10.0000   64.5839  59168.0000
## tenure                 1.0000    1.2792      4.0000
## good_country           0.0000    0.0182      1.0000
##
## Percent Balance Improvement:
##                     Mean Diff. eQQ Med eQQ Mean  eQQ Max
## distance               48.2996 56.7139  48.2975  33.9443
## age                    35.7306  0.0000  35.8883 -60.0000
## friend_cnt             25.1753 45.4545  25.1654   0.0000
## avg_friend_age         26.5857 68.5714  21.5672 -21.7391
## avg_friend_male        12.2518 80.0840  65.7176  54.8507
## friend_country_cnt     35.3078 60.0000  35.3001   0.0000
## songsListened          65.9458 68.3020  65.9329  13.2836
## lovedTracks            43.5343 41.5385  43.4655   2.5698
## posts                  20.7240  0.0000  20.2956   0.0000
## playlists              66.4144  0.0000  47.1533  15.3846
## shouts                 24.3717 33.3333  24.1763   0.0000
## tenure                 65.8879 66.6667  61.7834  60.0000
## good_country          -59.6007  0.0000 -59.8214   0.0000
##
```

```
## Sample sizes:
##           Control Treated
## All         34004    9823
## Matched      9823    9823
## Unmatched   24181       0
## Discarded       0       0

plot(mod_match)
```


QQ Plots
All
Matched
age
friend_cnt
avg_friend_age
10
30
50
70
0
2000
4000
Treated Units
Control Units


QQ Plots
All
Matched
avg_friend_male
friend_country_cnt
songsListened
0.0
0.4
0.8
0
40
80
120
0e+00
6e+05
Treated Units
Control Units


QQ Plots
All
Matched
lovedTracks
posts
playlists
0
4000
10000
0
4000
10000
0
40
80
120
Treated Units
Control Units



create a dataframe containing only the matched observations:

```
data_m <- match.data(mod_match)
dim(data_m)

## [1] 19646    17
```

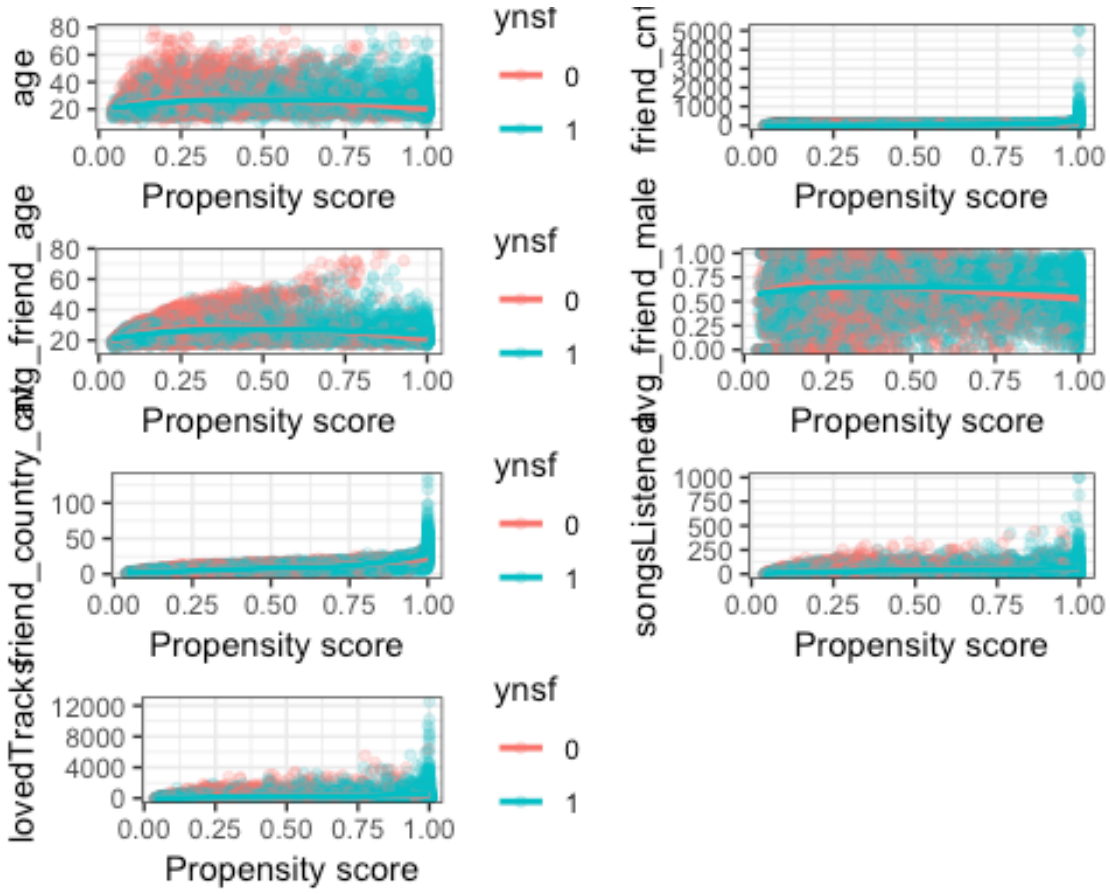
The final dataset is smaller than the original: it contains 19646 observations, meaning that 9823 pairs of treated and control observations were matched. The final dataset contains a variable called distance, which is the propensity score.

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.

```
fin_bal <- function(data, variable) {
  data$variable <- data[, variable]
  if (variable == 'songsListened') data$variable <- data$variable / 10^3
  data$ynsf <- as.factor(data$ynsf)
  support <- c(min(data$variable), max(data$variable))
  ggplot(data, aes(x = distance, y = variable, color = ynsf)) +
    geom_point(alpha = 0.2, size = 1.3) +
    geom_smooth(method = "loess", se = F) +
```

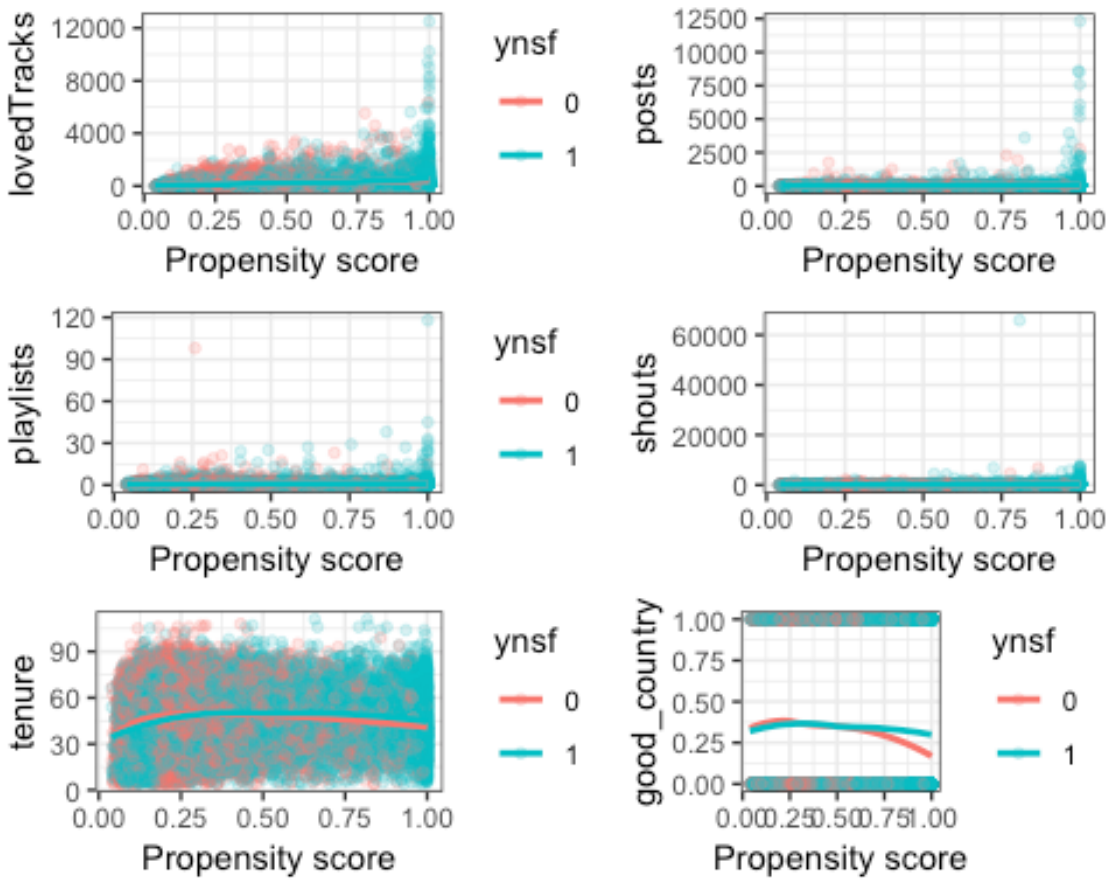
```
    xlab("Propensity score") +
    ylab(variable) +
    theme_bw() +
    ylim(support)
}

grid.arrange(
  fin_bal(data_m, "age"),
  fin_bal(data_m, "friend_cnt")  + theme(legend.position = "none"),
  fin_bal(data_m, "avg_friend_age"),
  fin_bal(data_m, "avg_friend_male") + theme(legend.position = "none"),
  fin_bal(data_m, "friend_country_cnt"),
  fin_bal(data_m, "songsListened") + theme(legend.position = "none"),
  fin_bal(data_m, "lovedTracks"),
    nrow =4, widths = c(1, 0.8)
)
```




```
grid.arrange(
  fin_bal(data_m, "lovedTracks"),
  fin_bal(data_m, "posts") + theme(legend.position = "none"),
  fin_bal(data_m, "playlists"),
  fin_bal(data_m, "shouts") + theme(legend.position = "none"),
  fin_bal(data_m, "tenure"),
```

```
  fin_bal(data_m, "good_country"),
  nrow = 3, widths = c(1, 0.8)
)
```

```
## Warning: Removed 4 rows containing missing values (geom_smooth).
```



- 4.2: Difference-in-means test mean difference for each covariate:

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

```
## Adding missing grouping variables: `ynsf`
```

```
## # A tibble: 2 x 14
##    ynsf   age  male friend_cnt avg_friend_age avg_friend_male
##   <dbl> <dbl> <dbl>      <dbl>          <dbl>           <dbl>
## 1     0  26.4 0.648       21.4           26.6           0.656
## 2     1  25.4 0.636       54.0           25.4           0.636
## # ... with 8 more variables: friend_country_cnt <dbl>,
## #   songsListened <dbl>, lovedTracks <dbl>, posts <dbl>, playlists <dbl>,
## #   shouts <dbl>, tenure <dbl>, good_country <dbl>
```

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

```
## [[1]]
##
##  Welch Two Sample t-test
##
## data:  data_m[, v] by data_m$ynsf
## t = 9.7592, df = 19282, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.8349505 1.2546352
## sample estimates:
## mean in group 0 mean in group 1
##        26.41800        25.37321
##
##
## [[2]]
##
##  Welch Two Sample t-test
##
## data:  data_m[, v] by data_m$ynsf
## t = 1.7116, df = 19643, p-value = 0.08699
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.001699904  0.025114339
## sample estimates:
## mean in group 0 mean in group 1
##       0.6479691       0.6362618
##
##
## [[3]]
##
##  Welch Two Sample t-test
##
## data:  data_m[, v] by data_m$ynsf
## t = -24.855, df = 10488, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -35.18808 -30.04352
## sample estimates:
## mean in group 0 mean in group 1
##        21.40517        54.02097
##
##
## [[4]]
##
##  Welch Two Sample t-test
##
```

```
## data:  data_m[, v] by data_m$ynsf
## t = 13.992, df = 18434, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  1.028425 1.363504
## sample estimates:
## mean in group 0 mean in group 1
##        26.58639        25.39043
##
##
## [[5]]
##
##  Welch Two Sample t-test
##
## data:  data_m[, v] by data_m$ynsf
## t = 5.6307, df = 19263, p-value = 1.82e-08
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.01298231 0.02684722
## sample estimates:
## mean in group 0 mean in group 1
##       0.6557225       0.6358077
##
##
## [[6]]
##
##  Welch Two Sample t-test
##
## data:  data_m[, v] by data_m$ynsf
## t = -38.685, df = 13879, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -4.527193 -4.090541
## sample estimates:
## mean in group 0 mean in group 1
##        5.076759        9.385626
##
##
## [[7]]
##
##  Welch Two Sample t-test
##
## data:  data_m[, v] by data_m$ynsf
## t = -11.642, df = 18439, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -7612.782 -5418.681
## sample estimates:
## mean in group 0 mean in group 1
##        27219.91        33735.64
```

```
##
##
## [[8]]
##
##  Welch Two Sample t-test
##
## data:  data_m[, v] by data_m$ynsf
## t = -15.424, df = 16085, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -101.9222  -78.9386
## sample estimates:
## mean in group 0 mean in group 1
##        134.9342        225.3647
##
##
## [[9]]
##
##  Welch Two Sample t-test
##
## data:  data_m[, v] by data_m$ynsf
## t = -5.6778, df = 11063, p-value = 1.399e-08
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -19.17429  -9.33268
## sample estimates:
## mean in group 0 mean in group 1
##         6.26947        20.52296
##
##
## [[10]]
##
##  Welch Two Sample t-test
##
## data:  data_m[, v] by data_m$ynsf
## t = -2.9701, df = 17743, p-value = 0.002981
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.11964186 -0.02450962
## sample estimates:
## mean in group 0 mean in group 1
##       0.6719943       0.7440700
##
##
## [[11]]
##
##  Welch Two Sample t-test
##
## data:  data_m[, v] by data_m$ynsf
## t = -8.5073, df = 10512, p-value < 2.2e-16
```

```
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -79.46493 -49.70295
## sample estimates:
## mean in group 0 mean in group 1
##        37.23557       101.81951
##
##
## [[12]]
##
##  Welch Two Sample t-test
##
## data:  data_m[, v] by data_m$ynsf
## t = 4.1015, df = 19607, p-value = 4.121e-05
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.5959372 1.6868684
## sample estimates:
## mean in group 0 mean in group 1
##        47.69012        46.54871
##
##
## [[13]]
##
##  Welch Two Sample t-test
##
## data:  data_m[, v] by data_m$ynsf
## t = 2.6737, df = 19641, p-value = 0.007509
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.004863394 0.031581684
## sample estimates:
## mean in group 0 mean in group 1
##       0.3614985       0.3432760
```

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:

```
with(data_m, t.test(adopter ~ ynsf))
```

```
##
##  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.

We can also do binomial regression:

```
glm_treat1 <- glm(adopter ~ ynsf, family = binomial(), data = data_m)
summary(glm_treat1)

##
## Call:
## glm(formula = adopter ~ ynsf, family = binomial(), data = data_m)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.6252  -0.6252  -0.4262  -0.4262   2.2108
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -2.35288    0.03583  -65.67   <2e-16 ***
## ynsf         0.81979    0.04451   18.42   <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: 15345  on 19645  degrees of freedom
## Residual deviance: 14986  on 19644  degrees of freedom
## AIC: 14990
##
## Number of Fisher Scoring iterations: 5

glm_treat2 <- glm(adopter ~ ynsf + age + friend_cnt + avg_friend_age +
avg_friend_male + friend_country_cnt
                  + lovedTracks + posts + playlists
                 + shouts + tenure + good_country
                  +  I(songsListened / 10^3), family = binomial(), data =
data_m)
summary(glm_treat2)

##
## Call:
## glm(formula = adopter ~ ynsf + age + friend_cnt + avg_friend_age +
##     avg_friend_male + friend_country_cnt + lovedTracks + posts +
##     playlists + shouts + tenure + good_country + I(songsListened/10^3),
##     family = binomial(), data = data_m)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
```

```
## -3.2269  -0.5694  -0.4544  -0.3793   2.4961
##
## Coefficients:
##                          Estimate Std. Error z value Pr(>|z|)
## (Intercept)           -3.188e+00  1.225e-01 -26.030  < 2e-16 ***
## ynsf                   7.244e-01  4.675e-02  15.494  < 2e-16 ***
## age                    1.897e-02  3.912e-03   4.850 1.23e-06 ***
## friend_cnt            -1.054e-04  2.742e-04  -0.384 0.700784
## avg_friend_age         8.544e-03  5.308e-03   1.610 0.107470
## avg_friend_male        6.805e-02  9.266e-02   0.734 0.462696
## friend_country_cnt     5.273e-03  3.610e-03   1.461 0.144146
## lovedTracks            5.264e-04  4.687e-05  11.232  < 2e-16 ***
## posts                  1.216e-04  8.881e-05   1.369 0.170870
## playlists              4.567e-02  1.216e-02   3.757 0.000172 ***
## shouts                 9.836e-05  7.270e-05   1.353 0.176067
## tenure                -1.997e-03  1.217e-03  -1.641 0.100842
## good_country          -3.942e-01  4.809e-02  -8.198 2.44e-16 ***
## I(songsListened/10^3)  4.560e-03  5.293e-04   8.617  < 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: 15345  on 19645  degrees of freedom
## Residual deviance: 14530  on 19632  degrees of freedom
## AIC: 14558
##
## Number of Fisher Scoring iterations: 5
```

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

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

```
res2 <- cor(Highnote)
res2
```

```
##                                ID          age         male   friend_cnt
## ID                    1.000000000  0.037640058  0.016121071  0.042525111
## age                   0.037640058  1.000000000  0.169075297 -0.033964722
## male                  0.016121071  0.169075297  1.000000000 -0.004292194
## friend_cnt            0.042525111 -0.033964722 -0.004292194  1.000000000
```

```
## avg_friend_age          0.035581652  0.688102645  0.049227806 -0.051956870
## avg_friend_male         0.006456926  0.075157294  0.051880384 -0.009592896
## friend_country_cnt      0.063656866 -0.031242578 -0.042381576  0.718526722
## subscriber_friend_cnt   0.054907363  0.077454545  0.007156261  0.781243469
## songsListened           0.069063354  0.021965699  0.116487201  0.213134540
## lovedTracks             0.080093025  0.054059810  0.022807694  0.195800406
## posts                   0.010076761  0.005055075  0.008733265  0.046903177
## playlists               0.034083872  0.112464340 -0.008456892  0.047125916
## shouts                  0.023707630 -0.023501423 -0.017216850  0.195353714
## adopter                 0.471166209  0.085879158  0.060513120  0.089397803
## tenure                  0.006125125  0.300314069  0.093779479 -0.001366727
## good_country           -0.003869528  0.097712013  0.001332812 -0.031759499
## ynsf                    0.084054858  0.105352120  0.006414118  0.281218822
## songsListened_1k        0.069063354  0.021965699  0.116487201  0.213134540
##                       avg_friend_age avg_friend_male friend_country_cnt
## ID                       0.035581652    0.0064569255        0.063656866
## age                      0.688102645    0.0751572936       -0.031242578
## male                     0.049227806    0.0518803840       -0.042381576
## friend_cnt              -0.051956870   -0.0095928960        0.718526722
## avg_friend_age           1.000000000    0.1817757237       -0.037194340
## avg_friend_male          0.181775724    1.0000000000       -0.022870690
## friend_country_cnt      -0.037194340   -0.0228706898        1.000000000
## subscriber_friend_cnt    0.062994976    0.0098483093        0.508548615
## songsListened            0.001649563    0.0195449919        0.328554333
## lovedTracks              0.043798816   -0.0006237683        0.308362521
## posts                    0.006227368    0.0052870775        0.085229354
## playlists                0.104146564   -0.0025497456        0.095861697
## shouts                  -0.022668047   -0.0057908193        0.214654915
## adopter                  0.075862799    0.0173318849        0.143259895
## tenure                   0.317447021    0.0857552704        0.008143325
## good_country             0.094390733    0.0216240488       -0.051725026
## ynsf                     0.132493714    0.0301354328        0.452729496
## songsListened_1k         0.001649563    0.0195449919        0.328554333
##                       subscriber_friend_cnt songsListened   lovedTracks
## ID                            0.054907363   0.069063354  0.0800930255
## age                           0.077454545   0.021965699  0.0540598100
## male                          0.007156261   0.116487201  0.0228076943
## friend_cnt                    0.781243469   0.213134540  0.1958004056
## avg_friend_age                0.062994976   0.001649563  0.0437988161
## avg_friend_male               0.009848309   0.019544992 -0.0006237683
## friend_country_cnt            0.508548615   0.328554333  0.3083625207
## subscriber_friend_cnt         1.000000000   0.137199916  0.1762751646
## songsListened                 0.137199916   1.000000000  0.2331350138
## lovedTracks                   0.176275165   0.233135014  1.0000000000
## posts                         0.054450057   0.089020150  0.0582214108
## playlists                     0.082733792   0.074007584  0.1347662040
## shouts                        0.137848108   0.130239814  0.0999825871
## adopter                       0.115550333   0.145427363  0.1650097408
## tenure                        0.018916949   0.241593913  0.0108400035
## good_country                  0.008683892   0.027378568  0.0131821863
```

```
## ynsf                          0.334185344   0.263814198  0.2282066299
## songsListened_1k              0.137199916   1.000000000  0.2331350138
##                                    posts     playlists       shouts     adopter
## ID                          0.010076761  0.0340838716  0.023707630  0.47116621
## age                         0.005055075  0.1124643397 -0.023501423  0.08587916
## male                        0.008733265 -0.0084568920 -0.017216850  0.06051312
## friend_cnt                  0.046903177  0.0471259162  0.195353714  0.08939780
## avg_friend_age              0.006227368  0.1041465637 -0.022668047  0.07586280
## avg_friend_male             0.005287077 -0.0025497456 -0.005790819  0.01733188
## friend_country_cnt          0.085229354  0.0958616970  0.214654915  0.14325989
## subscriber_friend_cnt       0.054450057  0.0827337920  0.137848108  0.11555033
## songsListened               0.089020150  0.0740075837  0.130239814  0.14542736
## lovedTracks                 0.058221411  0.1347662040  0.099982587  0.16500974
## posts                       1.000000000  0.0120232306  0.122411385  0.03658764
## playlists                   0.012023231  1.0000000000  0.015808382  0.07572234
## shouts                      0.122411385  0.0158083825  1.000000000  0.05266217
## adopter                     0.036587643  0.0757223372  0.052662169  1.00000000
## tenure                      0.039933777  0.0716343706  0.021943838  0.02434451
## good_country               -0.001882863 -0.0008543083 -0.017101503 -0.04003519
## ynsf                        0.063393565  0.0708736019  0.099241045  0.19178523
## songsListened_1k            0.089020150  0.0740075837  0.130239814  0.14542736
##                                  tenure  good_country         ynsf
## ID                          0.006125125 -0.0038695281  0.084054858
## age                         0.300314069  0.0977120133  0.105352120
## male                        0.093779479  0.0013328121  0.006414118
## friend_cnt                 -0.001366727 -0.0317594993  0.281218822
## avg_friend_age              0.317447021  0.0943907326  0.132493714
## avg_friend_male             0.085755270  0.0216240488  0.030135433
## friend_country_cnt          0.008143325 -0.0517250258  0.452729496
## subscriber_friend_cnt       0.018916949  0.0086838916  0.334185344
## songsListened               0.241593913  0.0273785683  0.263814198
## lovedTracks                 0.010840004  0.0131821863  0.228206630
## posts                       0.039933777 -0.0018828628  0.063393565
## playlists                   0.071634371 -0.0008543083  0.070873602
## shouts                      0.021943838 -0.0171015027  0.099241045
## adopter                     0.024344506 -0.0400351891  0.191785227
## tenure                      1.000000000  0.1320492934  0.070417882
## good_country                0.132049293  1.0000000000 -0.009968337
## ynsf                        0.070417882 -0.0099683365  1.000000000
## songsListened_1k            0.241593913  0.0273785683  0.263814198
##                            songsListened_1k
## ID                               0.069063354
## age                              0.021965699
## male                             0.116487201
## friend_cnt                       0.213134540
## avg_friend_age                   0.001649563
## avg_friend_male                  0.019544992
## friend_country_cnt               0.328554333
## subscriber_friend_cnt            0.137199916
## songsListened                    1.000000000
```

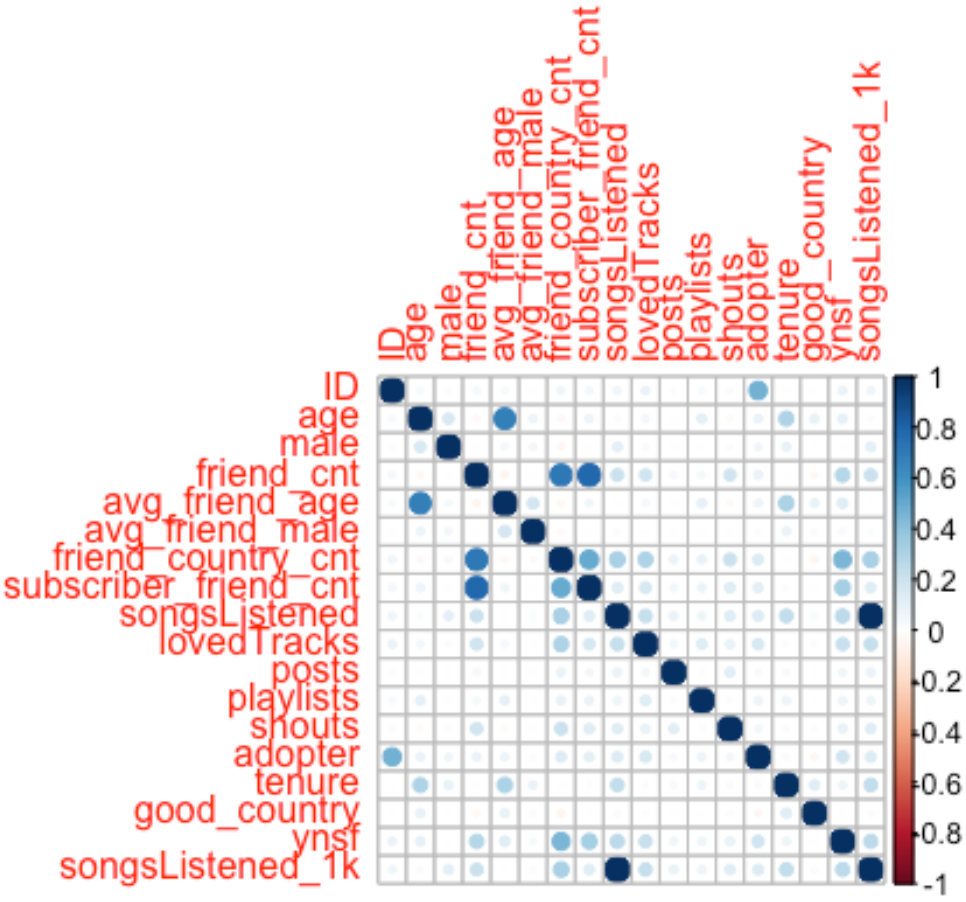
```
## lovedTracks                0.233135014
## posts                      0.089020150
## playlists                  0.074007584
## shouts                     0.130239814
## adopter                    0.145427363
## tenure                     0.241593913
## good_country               0.027378568
## ynsf                       0.263814198
## songsListened_1k           1.000000000
```

```
round(res2, 4)
```

```
##                             ID     age    male friend_cnt avg_friend_age
## ID                      1.0000  0.0376  0.0161     0.0425         0.0356
## age                     0.0376  1.0000  0.1691    -0.0340         0.6881
## male                    0.0161  0.1691  1.0000    -0.0043         0.0492
## friend_cnt              0.0425 -0.0340 -0.0043     1.0000        -0.0520
## avg_friend_age          0.0356  0.6881  0.0492    -0.0520         1.0000
## avg_friend_male         0.0065  0.0752  0.0519    -0.0096         0.1818
## friend_country_cnt      0.0637 -0.0312 -0.0424     0.7185        -0.0372
## subscriber_friend_cnt   0.0549  0.0775  0.0072     0.7812         0.0630
## songsListened           0.0691  0.0220  0.1165     0.2131         0.0016
## lovedTracks             0.0801  0.0541  0.0228     0.1958         0.0438
## posts                   0.0101  0.0051  0.0087     0.0469         0.0062
## playlists               0.0341  0.1125 -0.0085     0.0471         0.1041
## shouts                  0.0237 -0.0235 -0.0172     0.1954        -0.0227
## adopter                 0.4712  0.0859  0.0605     0.0894         0.0759
## tenure                  0.0061  0.3003  0.0938    -0.0014         0.3174
## good_country           -0.0039  0.0977  0.0013    -0.0318         0.0944
## ynsf                    0.0841  0.1054  0.0064     0.2812         0.1325
## songsListened_1k        0.0691  0.0220  0.1165     0.2131         0.0016
##                        avg_friend_male friend_country_cnt
## ID                              0.0065             0.0637
## age                             0.0752            -0.0312
## male                            0.0519            -0.0424
## friend_cnt                     -0.0096             0.7185
## avg_friend_age                  0.1818            -0.0372
## avg_friend_male                 1.0000            -0.0229
## friend_country_cnt             -0.0229             1.0000
## subscriber_friend_cnt           0.0098             0.5085
## songsListened                   0.0195             0.3286
## lovedTracks                    -0.0006             0.3084
## posts                           0.0053             0.0852
## playlists                      -0.0025             0.0959
## shouts                         -0.0058             0.2147
## adopter                         0.0173             0.1433
## tenure                          0.0858             0.0081
## good_country                    0.0216            -0.0517
## ynsf                            0.0301             0.4527
## songsListened_1k                0.0195             0.3286
```

```
##                         subscriber_friend_cnt songsListened lovedTracks
## ID                                     0.0549        0.0691      0.0801
## age                                    0.0775        0.0220      0.0541
## male                                   0.0072        0.1165      0.0228
## friend_cnt                             0.7812        0.2131      0.1958
## avg_friend_age                         0.0630        0.0016      0.0438
## avg_friend_male                        0.0098        0.0195     -0.0006
## friend_country_cnt                     0.5085        0.3286      0.3084
## subscriber_friend_cnt                  1.0000        0.1372      0.1763
## songsListened                          0.1372        1.0000      0.2331
## lovedTracks                            0.1763        0.2331      1.0000
## posts                                  0.0545        0.0890      0.0582
## playlists                              0.0827        0.0740      0.1348
## shouts                                 0.1378        0.1302      0.1000
## adopter                                0.1156        0.1454      0.1650
## tenure                                 0.0189        0.2416      0.0108
## good_country                           0.0087        0.0274      0.0132
## ynsf                                   0.3342        0.2638      0.2282
## songsListened_1k                       0.1372        1.0000      0.2331
##                           posts playlists  shouts adopter  tenure
## ID                       0.0101    0.0341  0.0237  0.4712  0.0061
## age                      0.0051    0.1125 -0.0235  0.0859  0.3003
## male                     0.0087   -0.0085 -0.0172  0.0605  0.0938
## friend_cnt               0.0469    0.0471  0.1954  0.0894 -0.0014
## avg_friend_age           0.0062    0.1041 -0.0227  0.0759  0.3174
## avg_friend_male          0.0053   -0.0025 -0.0058  0.0173  0.0858
## friend_country_cnt       0.0852    0.0959  0.2147  0.1433  0.0081
## subscriber_friend_cnt    0.0545    0.0827  0.1378  0.1156  0.0189
## songsListened            0.0890    0.0740  0.1302  0.1454  0.2416
## lovedTracks              0.0582    0.1348  0.1000  0.1650  0.0108
## posts                    1.0000    0.0120  0.1224  0.0366  0.0399
## playlists                0.0120    1.0000  0.0158  0.0757  0.0716
## shouts                   0.1224    0.0158  1.0000  0.0527  0.0219
## adopter                  0.0366    0.0757  0.0527  1.0000  0.0243
## tenure                   0.0399    0.0716  0.0219  0.0243  1.0000
## good_country            -0.0019   -0.0009 -0.0171 -0.0400  0.1320
## ynsf                     0.0634    0.0709  0.0992  0.1918  0.0704
## songsListened_1k         0.0890    0.0740  0.1302  0.1454  0.2416
##                         good_country    ynsf songsListened_1k
## ID                           -0.0039  0.0841           0.0691
## age                           0.0977  0.1054           0.0220
## male                          0.0013  0.0064           0.1165
## friend_cnt                   -0.0318  0.2812           0.2131
## avg_friend_age                0.0944  0.1325           0.0016
## avg_friend_male               0.0216  0.0301           0.0195
## friend_country_cnt           -0.0517  0.4527           0.3286
## subscriber_friend_cnt         0.0087  0.3342           0.1372
## songsListened                 0.0274  0.2638           1.0000
## lovedTracks                   0.0132  0.2282           0.2331
## posts                        -0.0019  0.0634           0.0890
```

```
## playlists                 -0.0009  0.0709           0.0740
## shouts                    -0.0171  0.0992           0.1302
## adopter                   -0.0400  0.1918           0.1454
## tenure                     0.1320  0.0704           0.2416
## good_country               1.0000 -0.0100           0.0274
## ynsf                      -0.0100  1.0000           0.2638
## songsListened_1k           0.0274  0.2638           1.0000

corrplot(res2)
```



Based on the analysis, we find that the following varaibles are relatively highly correlated: age & avg_friend_age; 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.

```
mod.fit1 <- glm(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)
summary(mod.fit1)

##
## 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.

```
vif(mod.fit1)
```

```
##                   age                  male            friend_cnt
##              2.028083              1.061966              4.295009
##        avg_friend_age       avg_friend_male    friend_country_cnt
##              2.061113              1.042020              2.621221
## subscriber_friend_cnt           lovedTracks                 posts
##              3.007514              1.150339              1.088116
##             playlists      songsListened_1k                shouts
##              1.044297              1.280630              1.337860
##                tenure          good_country
##              1.213634              1.029508
```

```
sqrt(vif(mod.fit1)) > 2
```

```
##                   age                  male            friend_cnt
##                 FALSE                 FALSE                  TRUE
##        avg_friend_age       avg_friend_male    friend_country_cnt
##                 FALSE                 FALSE                 FALSE
## subscriber_friend_cnt           lovedTracks                 posts
##                 FALSE                 FALSE                 FALSE
##             playlists      songsListened_1k                shouts
##                 FALSE                 FALSE                 FALSE
##                tenure          good_country
##                 FALSE                 FALSE
```

```
outlierTest(mod.fit1)
```

```
##        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. We'll take out this variable to further analyze. Further, based on the mean analysis graph, logical assumption, and mod.fit1, we choose to include variable the following model: age, subscriber_friend_cnt, lovedTracks, playlists, songsListened_1k, good_country

```
mod.fit2 <- glm(adopter ~ age + subscriber_friend_cnt
                + lovedTracks + playlists + songsListened_1k
                 + good_country, family = binomial(), data = Highnote)
summary(mod.fit2)
```

```
##
## 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

vif(mod.fit2)

##                   age subscriber_friend_cnt           lovedTracks
##              1.041901              1.125002              1.121123
##             playlists      songsListened_1k          good_country
##              1.038990              1.086087              1.018351

sqrt(vif(mod.fit2)) > 2

##                   age subscriber_friend_cnt           lovedTracks
##                 FALSE                 FALSE                 FALSE
##             playlists      songsListened_1k          good_country
##                 FALSE                 FALSE                 FALSE

outlierTest(mod.fit2)

##        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, we will delete these outliers from the data set, and do a regression based on the new data set.

```
HighnoteNew <- Highnote[c(-32663,-21293,-10623,-37360,-3364,-12898,-27575,-
30653,-12277),]

mod.fit3 <- glm(adopter ~ age + subscriber_friend_cnt*age
                + lovedTracks + playlists + songsListened_1k
                 + good_country, family = binomial(), data = HighnoteNew)
summary(mod.fit3)
```

```
##
## 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:

```
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:

```
coef(mod.fit3)
```

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

```
exp(coef(mod.fit3))
```

```
##               (Intercept)                       age
##                0.02207494                1.04339763
##     subscriber_friend_cnt                lovedTracks
##                1.42014850                1.00078249
##                 playlists          songsListened_1k
##                1.07231587                1.00750757
##              good_country age:subscriber_friend_cnt
##                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 then the sum of the individual effects.

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

Takeaways

From my 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.