GitHub Net Promoter Score Exercise

Austin Cheng https://github.com/acheng4/NPS_Exercise (https://github.com/acheng4/NPS_Exercise)

1. What is our Net Promoter Score (NPS)?

```
read_in_data<-read.csv("/Users/Austin/Desktop/nps-exercise.csv")
analysis_data<-read_in_data

analysis_data$NPS_Category<-analysis_data$answer.10

analysis_data$NPS_Category[analysis_data$answer.10 >=9] <-"Promoter"
analysis_data$NPS_Category[7 <= analysis_data$answer.10 & analysis_data$answer.10 <= 8]
 <-"Passive"
analysis_data$NPS_Category[analysis_data$answer.10 <= 6] <-"Detractor"
analysis_data$NPS_Category[is.na(analysis_data$answer.10)]<-"Missing"

analysis_data$NPS_Assignment<-analysis_data$NPS_Category

analysis_data$NPS_Assignment[analysis_data$NPS_Category=="Promoter"]<-1
analysis_data$NPS_Assignment[analysis_data$NPS_Category=="Detractor"]<- -1
analysis_data$NPS_Assignment[analysis_data$NPS_Category=="Passive"]<- 0
analysis_data$NPS_Assignment[analysis_data$NPS_Category=="Missing"]<- -1

analysis_data$NPS_Assignment<-as.numeric(analysis_data$NPS_Assignment)

NPS = 100*sum(analysis_data$NPS_Assignment)/length(analysis_data$NPS_Assignment)
NPS
```

```
## [1] 59.3125
```

Rounding the calculation, the NPS is 59. In a common understanding of NPS, this is an excellent score, indicating high loyalty to GitHub.

As a conservative approach, I decided to label a missing value as a Detractor, because there are more possible values for a Detractor than for a Promoter or Passive.

2. What is the margin of error or confidence level of our NPS?

The Margin of Error is a measurement of possible sampling error based on the variance of the Random Variables and the Sample Size. The NPS is not a standard statistical measurement, but we can derive the variance based its underlying Multinomial Distribution. The NPS is calculated as a * Pr + b * Pa + c * D where Pr is the Promoters & a = +1, Pa is the Passives & b = 0, and D is the detractors & c = -1. Since b = 0 for the Passives, Var(NPS) = Var((100/n) * (a * Pr + c * D)) = (100/n)^2 * [a^2 * Var(Pr) + c^2 * Var(D) - 2 * a * c * Cov(Pr,D)]

The variance of a random variable for a multinomial distribution is n * p_i * (1-p_i) and the Cov(Pr,D) = -n * p_pr * p_d

Var(NPS) simplifes to (100/n)^2 * [(1^2) * Var(Pr) + (-1)^2 * Var(D) - 2 * 1 * (-1)*Cov(Pr,D)]

```
n=length(analysis_data$NPS_Assignment)

p_pr = sum(analysis_data$NPS_Assignment[analysis_data$NPS_Category=="Promoter"])/n
p_pr
```

```
## [1] 0.6625
```

```
p_d=length(which(analysis_data$NPS_Category=="Detractor" | analysis_data$NPS_Category==
"Missing" )) / n
p_d
```

```
## [1] 0.069375
```

```
var = n*p_pr*(1-p_pr) + n*p_d*(1-p_d) +n*2*p_d*p_pr

standard_error = qnorm(1-0.05/2) * sqrt(((100/n)^2 )* var)
standard_error
```

```
## [1] 3.020816
```

```
NPS+standard_error
```

```
## [1] 62.33332
```

```
NPS-standard_error
```

```
## [1] 56.29168
```

I am 95% confident that the true population NPS lies between 56.3 and 62.3. In layman terms, I would expect a sample NPS to be in this interval 95 times out 100 surveys.

3. How does a user’s account age impact NPS?:

Please refer to the images I placed in the GitHub Repository I created: https://github.com/acheng4/NPS_Exercise (https://github.com/acheng4/NPS_Exercise)

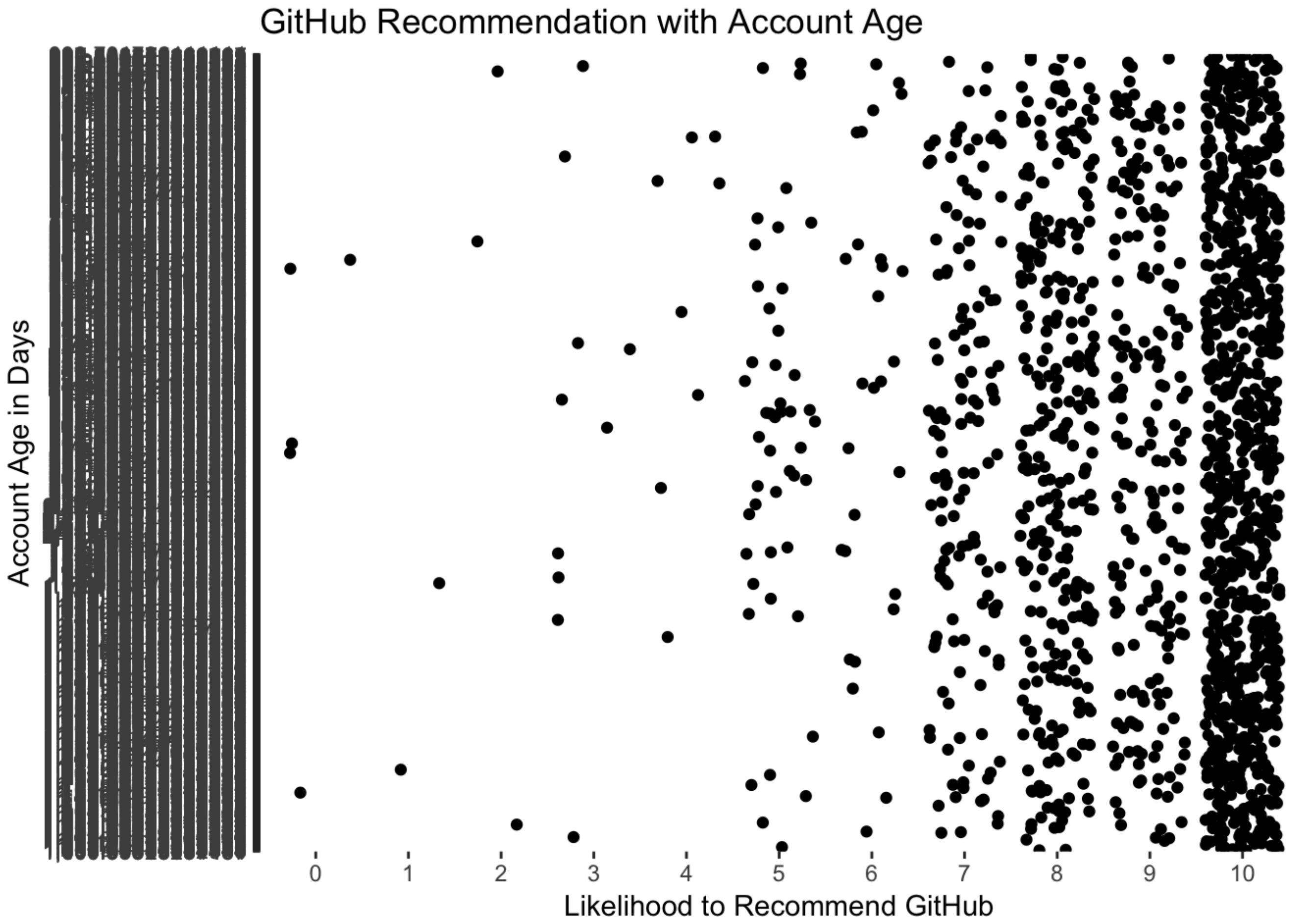
Since NPS is a heuristic metric, let’s look at just the Recommendation Score in relation to account age.

These two plots show us there is not much of an association between the account age and the Likehood to Recommend. On the higher end of recommendation, 9 & 10, there is no clump of data that trends to old or new. Everything is evenly distributed. Even at the middle recommendation range, 6 to 8, there is no discernible pattern. Looking at a box and whisker plot, which tells us where 50% of the data is centrally located, there is no discernible difference.

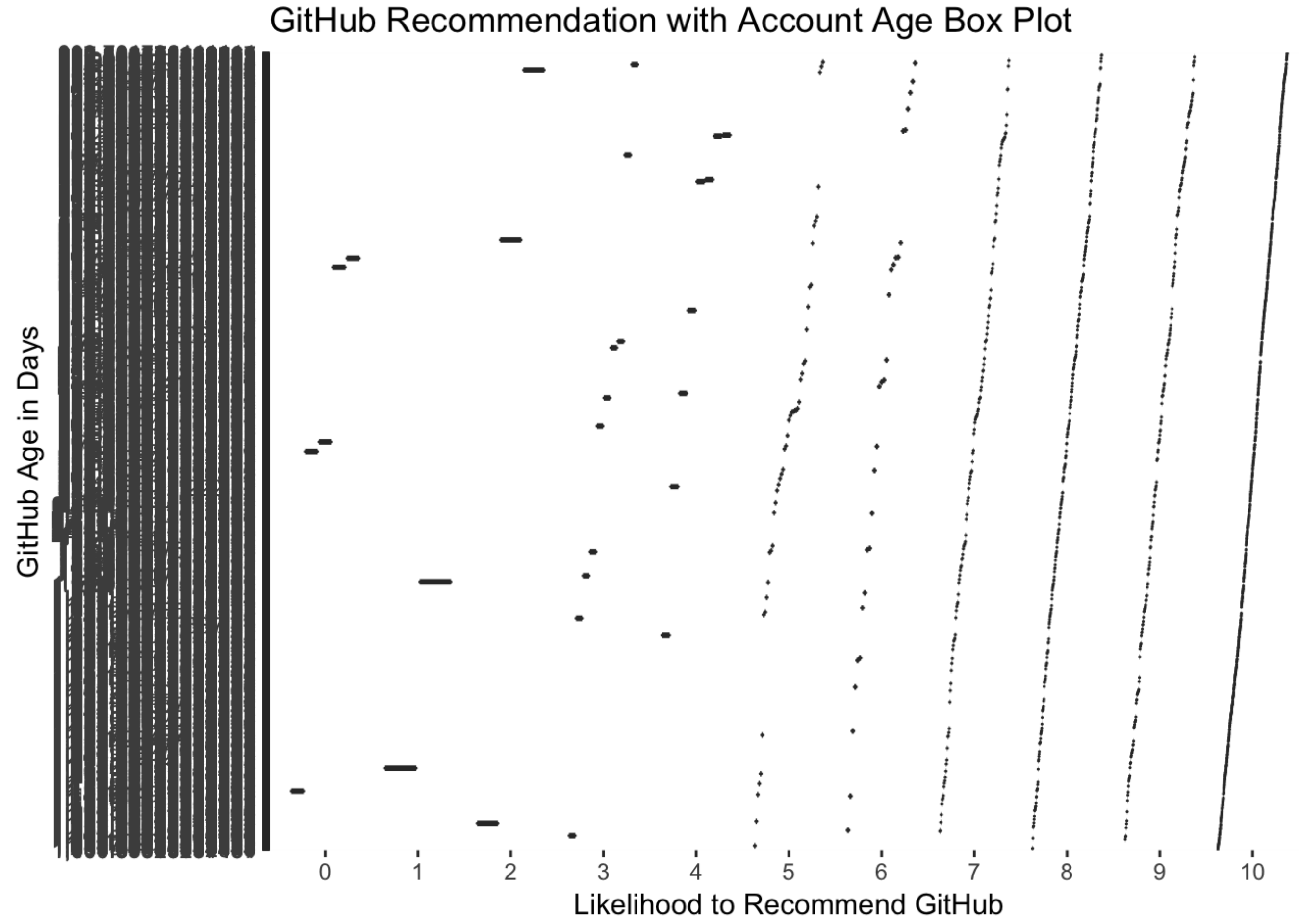
Apologies, for some reason the plot does not display properly in R Markdown. I tried to insert it as an image from a local file but also couldn't that to work. Please refer to the images I placed in the GitHub Repository I created: https://github.com/acheng4/NPS_Exercise (https://github.com/acheng4/NPS_Exercise)

```
#install.packages("ggplot2")
library(ggplot2)
age_analysis_data<-analysis_data[which(!is.na(analysis_data$answer.10)),]

qplot(factor(age_analysis_data$answer.10),age_analysis_data$github_age_in_days
      ,ylab = "Account Age in Days"
      ,xlab = "Likelihood to Recommend GitHub"
      ,geom = c("jitter")
      ,main = "GitHub Recommendation with Account Age"
      )
```



```
qplot(factor(age_analysis_data$answer.10),age_analysis_data$github_age_in_days
      ,ylab = "GitHub Age in Days"
      ,xlab = "Likelihood to Recommend GitHub"
      ,geom = c("boxplot")
      ,main = "GitHub Recommendation with Account Age Box Plot"
      )
```



```
# Please refer to images in GitHub Repository https://github.com/acheng4/NPS_Exercise
```

A quick correlation calculation near 0 tells us that the association of account age and Recommendation score is very low (-1 would be a strong negative association, 1 would be a strong positive association).

```
#install.packages("plyr")
library(plyr)
cor(as.numeric(age_analysis_data$github_age_in_days), age_analysis_data$answer.10)
```

```
## [1] -0.04027659
```

Lastly, comparing NPS categories to the average age of the account, there is no discernible difference. While the mean account age looks like it trends down with higher satisfaction, the standard deviation of the data is so great that we cannot make any conclusions.

```
#install.packages("plyr")
library(plyr)

ddply(age_analysis_data
      , .(NPS_Category)
      , summarize
      , Mean=mean(as.numeric(github_age_in_days))
      , Standard_Deviation=sd(as.numeric(github_age_in_days))
      )
```

```
##   NPS_Category     Mean Standard_Deviation
## 1    Detractor 853.5865           425.2691
## 2      Passive 812.5548           437.7983
## 3     Promoter 787.6453           474.2181
```

Nonetheless, we can try a linear regression model, treating the Recommendation Score as if it were a continuous variable and seeing how the age affects this.

```
age_regression <- lm(as.numeric(answer.10) ~ as.numeric(github_age_in_days), data=age_an
alysis_data)
summary(age_regression)
```

```
## 
## Call:
## lm(formula = as.numeric(answer.10) ~ as.numeric(github_age_in_days), 
##     data = age_analysis_data)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -8.9802 -0.8787  1.0136  1.1137  1.2225 
## 
## Coefficients:
##                                  Estimate Std. Error t value Pr(>|t|)    
## (Intercept)                     8.996e+00  7.857e-02 114.497   <2e-16 ***
## as.numeric(github_age_in_days) -1.370e-04  8.518e-05  -1.608    0.108    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 1.569 on 1591 degrees of freedom
## Multiple R-squared:  0.001622,   Adjusted R-squared:  0.0009947 
## F-statistic: 2.585 on 1 and 1591 DF,  p-value: 0.1081
```

The coefficient for age in this linear regression model is near 0. In addition, there is not a statistically significant impact that age has on the Recommendation Likelihood. If this were a model that I were actually going to evaluate, I would also have to look at the residuals of the model values relative to the true values, determine if they are normally distributed and do not have a trend.

Given the results from these different exploratory analyses, it would not be a wise use of time to fit a model of age to Recommendation Likelihood. A limitation to this model is that the Recommendation Likelihood variable is categorical and hierarchical but the linear regerssion model I used here treats it as if it were continuous.

4. Which survey answers or other user attributes correlate meaningfully with a higher or lower NPS?

I ran chi-square tests between NPS Categorizations and the various survey questions. Using an alpha of 0.05, meaning that I am 95% confident that the results are not generated due to chance alone, the following responses were found to be statistically significant.

```
chi_square_data<-analysis_data[which(!is.na(analysis_data$answer.10)),]

improve<-table(chi_square_data$answer.14,chi_square_data$NPS_Category)
improve_2<-improve[2:dim(improve)[1],]
chisq.test(improve_2)
```

```
##
##  Pearson's Chi-squared test
##
## data:  improve_2
## X-squared = 37.329, df = 12, p-value = 0.0001975
```

```
improve_2
```

```
##
##                               Detractor Passive Promoter
##   Code review                      10      72      174
##   GitHub desktop apps              11      62      142
##   Help documentation               14      32       75
##   Other (please specify)           28      64      177
##   Pricing                          26      67      146
##   Project management tools          8      97      269
##   User administration tools         6      30       71
```

```
prop.table(improve_2,1)
```

```
##
##                                Detractor    Passive   Promoter
##   Code review               0.03906250 0.28125000 0.67968750
##   GitHub desktop apps       0.05116279 0.28837209 0.66046512
##   Help documentation        0.11570248 0.26446281 0.61983471
##   Other (please specify)    0.10408922 0.23791822 0.65799257
##   Pricing                   0.10878661 0.28033473 0.61087866
##   Project management tools  0.02139037 0.25935829 0.71925134
##   User administration tools 0.05607477 0.28037383 0.66355140
```

A respondent’s improvement recommendation for using GitHub was found to be statistically significant with NPS Categorization. Promoters believe that the Project Management Tools need the most improvement. Detractors have a higher association with believing Help documentation and Pricing need improvement.

```
role<-table(chi_square_data$answer.3,chi_square_data$NPS_Category)
role_2<-role[2:dim(role)[1],]
role_2
```

```
##
##                             Detractor Passive Promoter
##   Designer                        4      13       20
##   Experienced programmer         59     244      656
##   New programmer                 21     122      259
##   Other (please specify)         11      28       51
##   Project/product manager         4      19       62
##   Writer                          5       3       12
```

```
chisq.test(role_2)
```

```
## Warning in chisq.test(role_2): Chi-squared approximation may be incorrect
```

```
##
##  Pearson's Chi-squared test
##
## data:  role_2
## X-squared = 27.016, df = 10, p-value = 0.002589
```

```
prop.table(role_2,1)
```

```
##
##                              Detractor    Passive   Promoter
##   Designer                0.10810811 0.35135135 0.54054054
##   Experienced programmer  0.06152242 0.25443170 0.68404588
##   New programmer          0.05223881 0.30348259 0.64427861
##   Other (please specify)  0.12222222 0.31111111 0.56666667
##   Project/product manager 0.04705882 0.22352941 0.72941176
##   Writer                  0.25000000 0.15000000 0.60000000
```

Project/product managers and Experienced programmers have the higest associateion with being Promoters, while Writers and Designers are the greatest Detractors, although there are very few of them.

```
at_job<-table(chi_square_data$answer.9,chi_square_data$NPS_Category)
at_job_2<-at_job[2:dim(at_job)[1],]
at_job_2
```

```
##
##       Detractor Passive Promoter
##   No          67     203      389
##   Yes         37     226      670
```

```
chisq.test(at_job_2)
```

```
##
##  Pearson's Chi-squared test
##
## data:  at_job_2
## X-squared = 38.429, df = 2, p-value = 4.521e-09
```

```
prop.table(at_job_2,1)
```

```
##
##        Detractor    Passive   Promoter
##   No  0.10166920 0.30804249 0.59028832
##   Yes 0.03965702 0.24222937 0.71811361
```

The greatest Promoters have an association with using GitHub at work. Detractors responded 2.5x more frequently with not using it at work.

5. Based on your discoveries, what would you research next?

GitHub has an Excellent NPS, but there are ways to excel still more. Based on the Chi-Squared tests done in the previous section, I would reserach how Help Documentation and Pricing could be improved, since Detractors responded this is where it can be best improved. Keeping our Promoters happy is important as well, and they noted that Project Management Tools is the best way to improve. This indicates how we can continue adding value to those who are already our greatest advocates.

While the Writers and Designers were the greatest Detractors, they also represent an incredibly small population of GitHub users, and addressing their needs wouldn’t be the best focus. Since high satisfaction is associated with using GitHub at work, I would research what aspects of GitHub bring highest satisfaction on the job. Do the people who don’t use GitHub in their jobs use another Git product? If so, what aspects of GitHub stand out compared to competitors? There can be whitespace in selling GitHub Enterprise, further influencing GitHub’s NPS and overall product perception.

Lastly, I would want to analyze utilization metrics in relation to satisfaction. Some metrics I would be interested in are the number of repositories a user has, frequency of use, commits made, communication frequency, and social networking effects.