Text analysis

Junyan Yao

Research question: Whether chat engagement is associated with test outcomes(see how students' collaborative performance can be associated with students' performance in these math problems).

Data: Chat data and test outcome data

```
library(corpus)
library(Matrix)
```

Load and make the data

```
data<-read.csv("~/Documents/NYU/Fall 2017/Text Analysis Project/cpsv_text_project/chat_time_series.csv")
#data<- read.csv("C:/Users/jyao/Documents/Text Analysis/chat_time_series.csv") office comp
data<- data[,c(2,5,8)] #extract needed columns
head(data) #we have 135 groups in this dataset
```

```
##   group_id type                                                    content
## 1        1 chat                                  So how should we do this?
## 2        1 chat So I guess one of us should pick c and one should pick a?
## 3        1 chat                                                       Yes
## 4        1 chat                                                Ill pick a
## 5        1 chat                                               I'll take a
## 6        1 chat                                                    c then
```

```
#subset the data
chatdata<- data[which(data$type=="chat"),] #this is what we want to look at for now
problemdata<- data[which(data$type=="problem"),]
head(chatdata)
```

```
##   group_id type                                                    content
## 1        1 chat                                  So how should we do this?
## 2        1 chat So I guess one of us should pick c and one should pick a?
## 3        1 chat                                                       Yes
## 4        1 chat                                                Ill pick a
## 5        1 chat                                               I'll take a
## 6        1 chat                                                    c then
```

```
#load the outcome data
outcomedata<-read.csv("~/Documents/NYU/Fall 2017/Text Analysis Project/cpsv_text_project/group_outcomes.csv")
#outcomedata<-read.csv("C:/Users/jyao/Documents/Text Analysis/group_outcomes.csv") #office comp
head(outcomedata)
```

```
##   X group_id         w       delta
## 1 1      -53 0.9255376  0.09307671
## 2 2      -52 0.4795482  0.40842891
## 3 3      -51 0.9904785  1.01937085
## 4 4      -50 0.9254073  0.66388004
## 5 5      -49 0.9865247  0.65585781
## 6 6      -48 0.9176420 -0.28448291
```

```
subset1<- outcomedata[outcomedata$group_id>0,] #Get rid of all negative group_id
summary(subset1$delta) #we have 110 groups in this dataset
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## -2.698000 -0.481000 -0.001394 -0.043190  0.405800  3.310000
```

```
performance<-ifelse(subset1$delta>0.4058,"high",ifelse(subset1$delta< -0.481,"low","in-b
etween"))
temp22<-cbind(subset1,performance) #label the performance for these groups
head(temp22)
```

```
##     X group_id           w       delta performance
## 52 52        1 0.005253214 -0.47311933  in-between
## 53 53        2 0.994202046  2.31519714        high
## 54 54        3 0.775524810 -0.38482482  in-between
## 55 55        8 0.917280801  0.32462698  in-between
## 56 56       10 0.969194802  0.18964875  in-between
## 57 57       12 0.409264249  0.01658728  in-between
```

```
#try to get rid of the missing rows(some group id are missing in the outcome data)
merged_data<- merge(x=chatdata,y=temp22,by="group_id")
head(merged_data) #only 110 groups are in this dataset
```

```
##   group_id type                           content  X           w      delta
## 1        1 chat            which one are you on? 52 0.005253214 -0.4731193
## 2        1 chat                ok just got there 52 0.005253214 -0.4731193
## 3        1 chat                    The 4x+5y one 52 0.005253214 -0.4731193
## 4        1 chat oh I got 3 for the graph one lol 52 0.005253214 -0.4731193
## 5        1 chat                     -4x, yeah 16 52 0.005253214 -0.4731193
## 6        1 chat                      I'll take a 52 0.005253214 -0.4731193
##   performance
## 1  in-between
## 2  in-between
## 3  in-between
## 4  in-between
## 5  in-between
## 6  in-between
```

Make the data

```
#split to two groups- High performance group and low performance group;
#now we only want to compare the performance between high outcomes and low outcomes grou
ps
high_group<- merged_data[which(merged_data$performance=="high"),]
low_group<- merged_data[which(merged_data$performance=="low"),]

#get the most common non-punctuation, non-stop word terms in the chat
Y<- term_stats(merged_data$content, drop=stopwords_en, drop_punct=TRUE) #the support is
 the number of texts containing the term.
# by using drop= stopwords_en, we can exclude these "functional" words

#We kept these functional words in the analysis
Y_high<- term_stats(high_group$content)
Y_low<- term_stats(low_group$content)

S<- subset(Y, Y$support>5)
S_high<-subset(Y_high,Y_high$support>5)
S_low<-subset(Y_low,Y_low$support>5)
head(S_high,10)
```

```
##    term count support
## 1  i      864     797
## 2  ?      777     752
## 3  the    816     638
## 4  .     1016     558
## 5  ,      694     555
## 6  is     534     489
## 7  you    368     356
## 8  to     401     352
## 9  a      334     303
## 10 and    327     303
```

```
head(S_low,10)
```

```
##    term count support
## 1  i      586     517
## 2  ?      529     498
## 3  the    562     454
## 4  .      694     404
## 5  is     417     384
## 6  ,      474     372
## 7  ok     255     254
## 8  you    251     240
## 9  to     257     232
## 10 it     238     223
```

```
#higher-order n-grams
term_stats(merged_data$content,ngrams = 3)
```

```
##    term               count support
## 1  . . .                416     356
## 2  the next one         135     135
## 3  i have no             89      89
## 4  the value of          86      81
## 5  on this one           76      76
## 6  have no idea          74      74
## 7  for the next          71      71
## 8  i don't know          67      67
## 9  total number of       77      63
## 10 the answer is         63      63
## 11 what is the           59      59
## 12 go with that          53      53
## 13 value of x            51      50
## 14 number of boxes       48      46
## 15 . . i                 46      46
## 16 what do you           44      44
## 17 number of students    44      43
## 18 in class a            46      42
## 19 i think the           42      42
## 20 next one is           42      42
## ⋮  (39129 rows total)
```

```
term_stats(merged_data$content,ngrams = 4)
```

```
##    term                     count support
## 1  i have no idea              64      64
## 2  for the next one            37      37
## 3  . . . .                    39      35
## 4  total number of students    32      31
## 5  . . . i                     29      29
## 6  total number of boxes       28      27
## 7  number of students in       27      26
## 8  sounds good to me           26      26
## 9  the total number of         31      24
## 10 i have no clue              24      24
## 11 i think the answer          24      24
## 12 what do you think           24      24
## 13 the next one ?              23      23
## 14 think the answer is         23      23
## 15 are you on the              22      22
## 16 sold in class a             21      21
## 17 i think it is               20      20
## 18 on the next one             20      20
## 19 number of boxes of          21      19
## 20 . what is the               19      19
## ⋮  (40400 rows total)
```

```
term_stats(merged_data$content,ngrams = 5)
```

```
##    term                         count support
## 1  i think the answer is           21      21
## 2  total number of students in     21      20
## 3  number of boxes of cookies      20      18
## 4  i don't know how to             17      17
## 5  of boxes of cookies sold        18      16
## 6  number of students in class     17      16
## 7  what do you think ?             16      16
## 8  boxes of cookies sold in        17      15
## 9  of cookies sold in class        17      15
## 10 of students in class a          15      15
## 11 for the next one ?              14      14
## 12 the total number of boxes       14      13
## 13 total number of boxes of        14      13
## 14 no idea on this one             13      13
## 15 have no idea how to             12      12
## 16 no clue on this one             12      12
## 17 what is the value of            12      12
## 18 - 3 , - 1                       11      11
## 19 cookies sold in class a         11      11
## 20 i have no idea how              11      11
## ⋮  (36130 rows total)
```

```
term_stats(high_group$content,ngrams = 4)
```

```
##    term                       count support
## 1  i think the answer            15      15
## 2  think the answer is           14      14
## 3  . . . i                       13      13
## 4  on this one .                 13      13
## 5  for the next one              12      12
## 6  go with that .                11      11
## 7  let's go with that            11      11
## 8  what do you think             11      11
## 9  total number of boxes         11      10
## 10 number of students in         10      10
## 11 the value of a                10      10
## 12 total number of students      10      10
## 13 i have no clue                 9       9
## 14 one . . .                      9       9
## 15 the next one ?                 9       9
## 16 nice working with you          8       8
## 17 sounds good to me              8       8
## 18 that's what i got              8       8
## 19 the total number of           10       7
## 20 + 18 = 0                       7       7
## ⋮  (13701 rows total)
```

```
term_stats(low_group$content,ngrams = 4)
```

```
##    term                    count support
## 1  i have no idea             12      12
## 2  . . . .                    11       9
## 3  ok . . .                    8       8
## 4  the next one is             8       8
## 5  lets go with that           7       7
## 6  boxes of cookies sold       6       6
## 7  cookies sold in class       6       6
## 8  don't know how to           6       6
## 9  how to do this              6       6
## 10 i am on the                 6       6
## 11 i don't know how            6       6
## 12 nice working with you       6       6
## 13 on the next one             6       6
## 14 on the next question        6       6
## 15 on to the next              6       6
## 16 . . . i                     5       5
## 17 . what is the               5       5
## 18 average number of boxes     5       5
## 19 have no idea how            5       5
## 20 i think it is               5       5
## ⋮  (9932 rows total)
```

```
Y_high<- Y_high[,-3] #drop the support column
Y_low<- Y_low[,-3] #drop the support column
names(Y_high)[2]<- paste("high")
names(Y_low)[2]<- paste("low")
dat<- merge(Y_high,Y_low, by="term",all = TRUE) #create the dataset for High and low gro
ups counts by terms
dat[is.na(dat)]<- 0
```

Don’t run

Emotion-lexicon

```
#Emotion-Lexicon
affect<- subset(affect_wordnet,emotion != "Neutral")
affect$emotion<- droplevels(affect$emotion) #drop the unused neutral level
affect$category<- droplevels(affect$category) #drop unused categories

term_stats(merged_data$content, subset = term %in% affect$term)
```

```
##    term      count support
## 1  good        284     281
## 2  sorry       160     160
## 3  like        139     139
## 4  submit       90      89
## 5  still        82      80
## 6  cool         78      77
## 7  move         76      76
## 8  great        64      63
## 9  easy         50      45
## 10 bad          37      37
## 11 hope         36      36
## 12 down         21      20
## 13 hopefully    18      18
## 14 close        16      16
## 15 hate         16      16
## 16 positive     16      16
## 17 trust        15      15
## 18 confused     14      14
## 19 confusing    14      14
## 20 care         12      12
## ⋮  (74 rows total)
```

```
term_stats(high_group$content, subset = term %in% affect$term)
```

```
##    term      count support
## 1  good        100      97
## 2  like         47      47
## 3  sorry        43      43
## 4  submit       31      31
## 5  cool         27      27
## 6  move         26      26
## 7  great        26      25
## 8  easy         25      21
## 9  bad          13      13
## 10 still        12      12
## 11 down         10       9
## 12 hate          9       9
## 13 hope          9       9
## 14 hopefully     6       6
## 15 care          5       5
## 16 close         5       5
## 17 dear          5       5
## 18 positive      5       5
## 19 trust         5       5
## 20 weight        5       5
## ⋮  (44 rows total)
```

```
term_stats(low_group$content, subset = term %in% affect$term)
```

```
##    term      count support
## 1  good         52      52
## 2  still        41      39
## 3  sorry        36      36
## 4  submit       30      30
## 5  like         23      23
## 6  move         19      19
## 7  cool         12      12
## 8  hope         10      10
## 9  hopefully    10      10
## 10 bad           9       9
## 11 great         7       7
## 12 easy          6       5
## 13 close         3       3
## 14 horrible      3       3
## 15 positive      3       3
## 16 terrible      3       3
## 17 confused      2       2
## 18 down          2       2
## 19 score         2       2
## 20 stupid        2       2
## ⋮  (45 rows total)
```

```
text_sample(high_group$content,"hard")
```

```
##   text            before             instance             after
## 1 686                                  hard    questions
## 2 372               next one looks     hard
## 3 600                                  hard    question
## 4 381               next one looks     hard
## 5 3718  individual part wasn't this    hard    for me
```

```
text_sample(low_group$content,"hard")
```

```
##   text            before             instance             after
## 1 396           Thanks for all your    hard    work.
## 2 475          Some of these are so    hard    bc I can't remember how to …
## 3 2948                    this is a    hard    one now
## 4 94   …guess the other because its    hard
## 5 198                      this one    hard
## 6 234                          also    hard    one
```

```
#term emotion matrix
#segment the text into smaller chunks and then compute the emotion occurence rates in ea
ch chunk, broken down by category ("positive","negative","ambiguous")

term_score<- with(affect, unclass(table(term,emotion)))
head(term_score) #while not very informative
```

```
##             emotion
## term         Positive Negative Ambiguous
##   abase             0        2         0
##   abash             0        1         0
##   abashed           0        1         0
##   abashment         0        1         0
##   abhor             0        1         0
##   abhorrence        0        1         0
```

create 2 by 2 tables for each term in the chat

The outcome X are 2666 *22 matrix. Each matrix is a 2* 2 table for each term.

```
#create 2 * 2 tables for each term
aux<- 1:length(dat$term)
x<- rep(list(diag(2)), 2677)
for (i in 1:length(aux)){
  x[[i]][1,1]<-dat$high[[i]]
  x[[i]][2,1]<-dat$low[[i]]
  x[[i]][1,2]<-colSums(dat[,c(2,3)])[1]-dat$high[[i]]
  x[[i]][2,2]<-colSums(dat[,c(2,3)])[2]-dat$low[[i]]
  colnames(x[[i]])<- c(dat$term[i], paste0("\u00ac",dat$term[i]))
  rownames(x[[i]])<- c("high", "low")
}

#one example
x[[2010]]
```

```
##      right ¬right
## high   146  26350
## low     76  17911
```

The 2010th Matrix shows the frequency of “right” term is 146 in the high performance group, and not term “right” is 26366. In the low performance group, the frequency for term “right” is 76. The ratio below this term for between high performance and low performance groups is 146/76=1.92

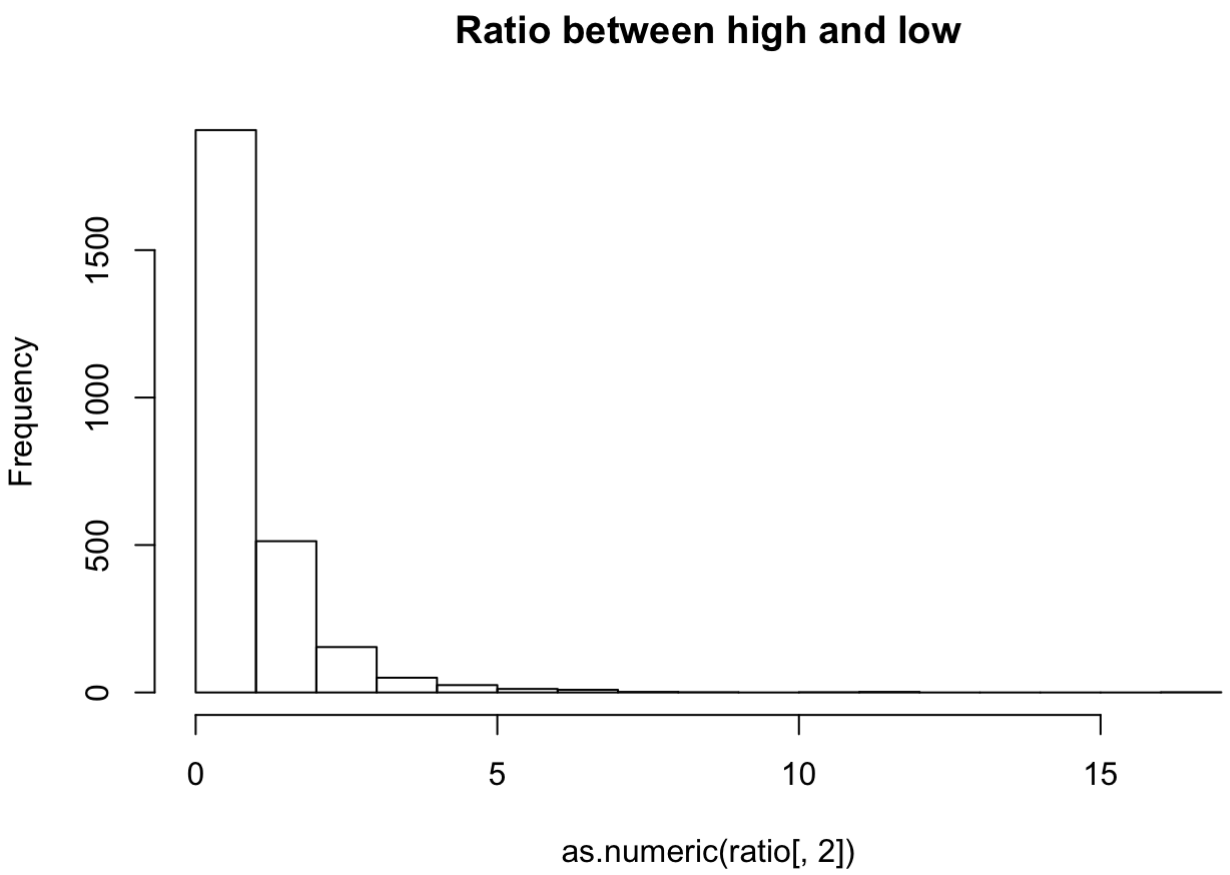
Now we would like to explore all terms ratio between high preformance groups and low preformance groups.

```
ratio<- matrix(NA,nrow=2677,ncol=2)
for (i in 1:length(x)){
  ratio[i,1]<- colnames(x[[i]])[1] #high/low for the rest of terms
  ratio[i,2]<- x[[i]][1,1]/(x[[i]][2,1]+1)#add 0.01 here to avoid infinite value
}
colnames(ratio)<- c("term","ratio")
head(ratio)
```

```
##      term ratio
## [1,] "_"  "0"
## [2,] "-"  "2.49152542372881"
## [3,] ","  "1.46105263157895"
## [4,] ";"  "0.294117647058824"
## [5,] ":"  "1.32075471698113"
## [6,] "!"  "1.88888888888889"
```

Look at the distribution of ratio

```
hist(as.numeric(ratio[,2]),main="Ratio between high and low") #very skewed distribution
```




```
Ordered_Ratio<- ratio[order(as.numeric(ratio[,2]), decreasing=TRUE),] #sort the order
head(Ordered_Ratio,50) #the biggest 10 terms ratio
```

```
##       term            ratio
##  [1,] "pi"            "17"
##  [2,] "4.71"          "12"
##  [3,] "aye"           "12"
##  [4,] "o"             "11"
##  [5,] "hundredth"     "9"
##  [6,] "circle"        "8"
##  [7,] "points"        "8"
##  [8,] "65"            "7"
##  [9,] "lowest"        "7"
## [10,] "meant"         "7"
## [11,] "mind"          "7"
## [12,] "n1"            "7"
## [13,] "order"         "7"
## [14,] "whatever"      "7"
## [15,] "squared"       "6.5"
## [16,] "ha"            "6.25"
## [17,] "243"           "6"
## [18,] "7.5"           "6"
## [19,] "9.6"           "6"
## [20,] "ef"            "6"
## [21,] "find"          "6"
## [22,] "gd"            "6"
## [23,] "greater"       "6"
## [24,] "hahaha"        "6"
## [25,] "plug"          "6"
## [26,] "shit"          "6"
## [27,] "taken"         "6"
## [28,] "looking"       "5.5"
## [29,] "8.5"           "5"
## [30,] "brain"         "5"
## [31,] "care"          "5"
## [32,] "circumference" "5"
## [33,] "closet"        "5"
## [34,] "crap"          "5"
## [35,] "formula"       "5"
## [36,] "freeze"        "5"
## [37,] "heads"         "5"
## [38,] "high"          "5"
## [39,] "hmmm"          "5"
## [40,] "job"           "5"
## [41,] "made"          "5"
## [42,] "messed"        "5"
## [43,] "needs"         "5"
## [44,] "sq"            "5"
## [45,] "tens"          "5"
## [46,] "thousands"     "5"
## [47,] "yikes"         "5"
## [48,] "we're"         "4.66666666666667"
## [49,] "hate"          "4.5"
## [50,] "shall"         "4.5"
```

```
#check the case "pi"
x[[1815]]
```

```
##      pi   ¬pi
## high 17 26479
## low   0 17987
```

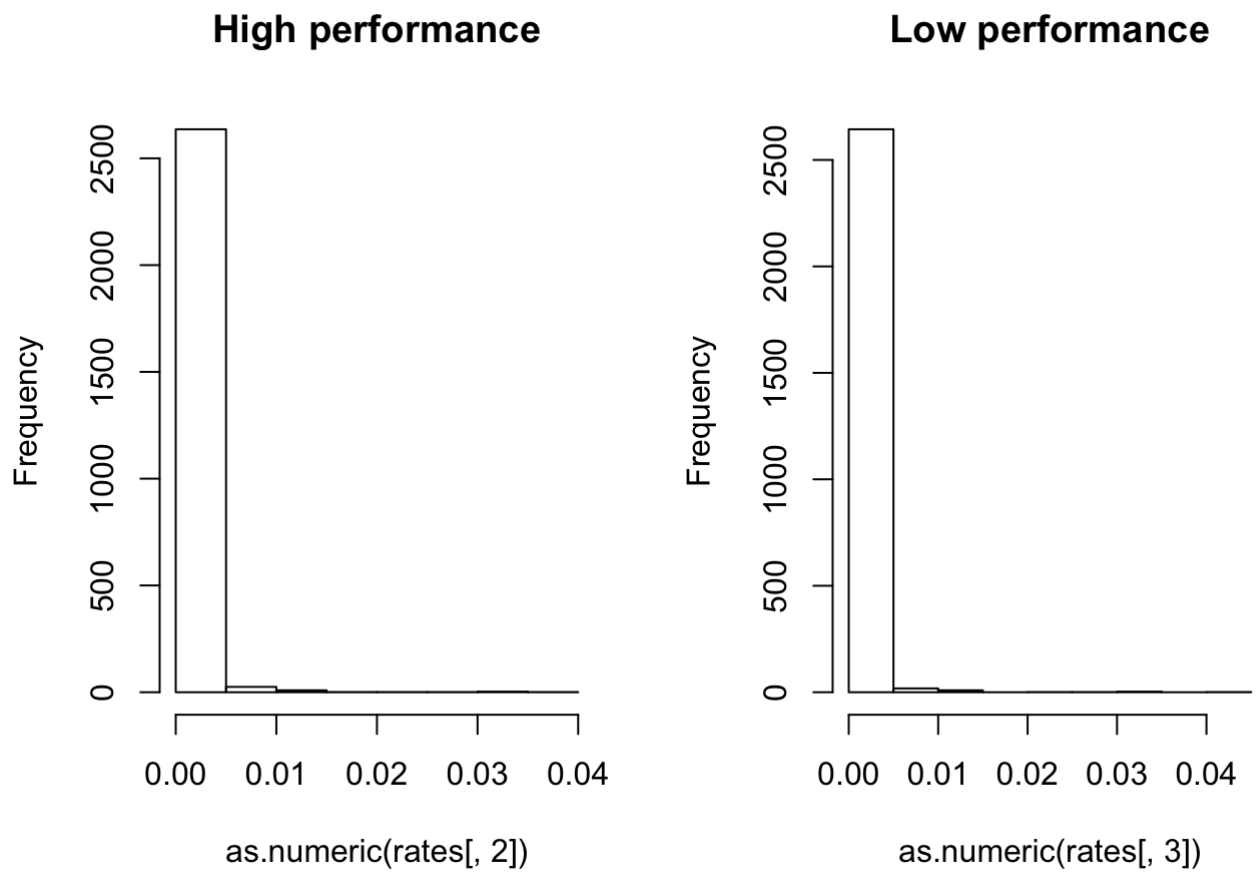
Here are the rates between the term and the rest of terms

Rates=term/non_term

```
rates<- matrix(NA, nrow = 2677, ncol = 3)
for (i in 1:length(x)){
  rates[i,1]<-colnames(x[[i]])[1]
  rates[i,2]<- x[[i]][1,1]/(x[[i]][1,2]) #high performance group
  rates[i,3]<- x[[i]][2,1]/(x[[i]][2,2]) #low performance group
}
colnames(rates)<- c("term","high","low")
```

look at the rates distribution

```
par(mfrow=c(1,2))
hist(as.numeric(rates[,2]), main="High performance")
hist(as.numeric(rates[,3]), main="Low performance")
```



```
Ordered_Rates_high<- rates[order(as.numeric(rates[,2]), decreasing=TRUE),]
Ordered_Rates_low<- rates[order(as.numeric(rates[,3]), decreasing=TRUE),]


head(Ordered_Rates_high,20)
```

```
##       term    high                   low
##  [1,] "."     "0.0398744113029827"  "0.0401318452553056"
##  [2,] "i"     "0.0337078651685393"  "0.0336762255042814"
##  [3,] "the"   "0.0317757009345794"  "0.0322525107604017"
##  [4,] "?"     "0.030211127959874"   "0.0303012945354565"
##  [5,] ","     "0.026897139756608"   "0.0270656084051847"
##  [6,] "is"    "0.0205685232262538"  "0.0237336368810472"
##  [7,] "to"    "0.0153669285303698"  "0.0144952058657642"
##  [8,] "you"   "0.0140845070422535"  "0.0141520072169599"
##  [9,] "a"     "0.012766608057488"   "0.0125534789461833"
## [10,] "of"    "0.0126504872921842"  "0.0111873172925568"
## [11,] "and"   "0.0124957010202912"  "0.011130473888358"
## [12,] "for"   "0.0117997479665483"  "0.0131238030866284"
## [13,] "it"    "0.0114907425081122"  "0.0134092061524593"
## [14,] "so"    "0.0112591122476241"  "0.00993823694553621"
## [15,] "ok"    "0.010989010989011"   "0.0143807805098128"
## [16,] "="     "0.0105648575460544"  "0.00457972633342642"
## [17,] "that"  "0.00979458058615039" "0.00937149270482604"
## [18,] "one"   "0.00956372642408078" "0.0109031641656831"
## [19,] "think" "0.00703127969290411" "0.00570310315907185"
## [20,] "be"    "0.00672517952809757" "0.00362682736301752"
```

```
head(Ordered_Rates_low,20)
```

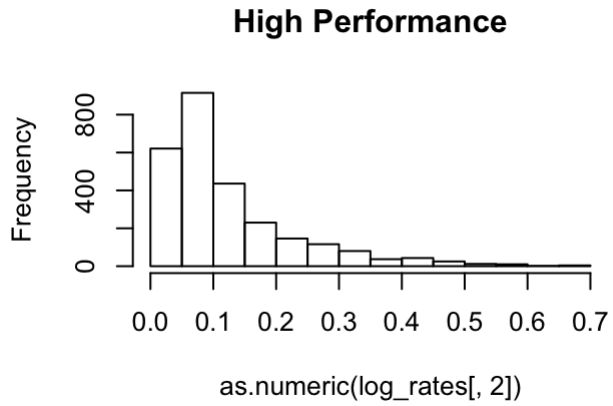
```
##       term   high                   low
##  [1,] "."    "0.0398744113029827"  "0.0401318452553056"
##  [2,] "i"    "0.0337078651685393"  "0.0336762255042814"
##  [3,] "the"  "0.0317757009345794"  "0.0322525107604017"
##  [4,] "?"    "0.030211127959874"   "0.0303012945354565"
##  [5,] ","    "0.026897139756608"   "0.0270656084051847"
##  [6,] "is"   "0.0205685232262538"  "0.0237336368810472"
##  [7,] "to"   "0.0153669285303698"  "0.0144952058657642"
##  [8,] "ok"   "0.010989010989011"   "0.0143807805098128"
##  [9,] "you"  "0.0140845070422535"  "0.0141520072169599"
## [10,] "it"   "0.0114907425081122"  "0.0134092061524593"
## [11,] "for"  "0.0117997479665483"  "0.0131238030866284"
## [12,] "a"    "0.012766608057488"   "0.0125534789461833"
## [13,] "of"   "0.0126504872921842"  "0.0111873172925568"
## [14,] "and"  "0.0124957010202912"  "0.011130473888358"
## [15,] "one"  "0.00956372642408078" "0.0109031641656831"
## [16,] "so"   "0.0112591122476241"  "0.00993823694553621"
## [17,] "that" "0.00979458058615039" "0.00937149270482604"
## [18,] "next" "0.00618995177154141" "0.00914497307001795"
## [19,] "we"   "0.00672517952809757" "0.00897515005328995"
## [20,] "on"   "0.00657219921741443" "0.0077315255756625"
```

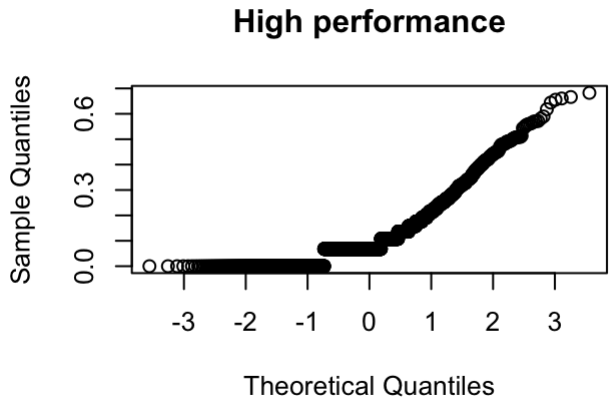
Try the log ratio per the gender lesson

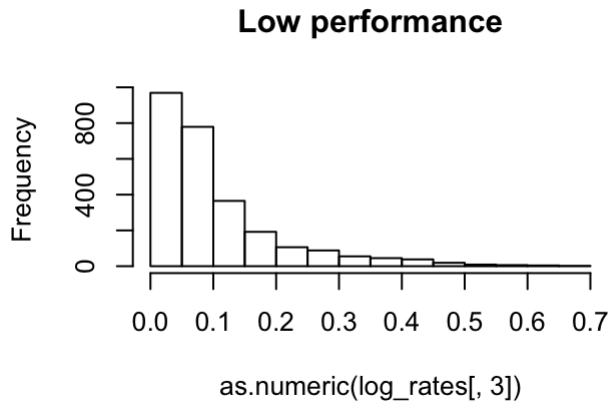
```
log_rates<- matrix(NA, nrow = 2677, ncol = 3)
for (i in 1:length(x)){
  log_rates[i,1]<-colnames(x[[i]])[1]
  log_rates[i,2]<- log2(x[[i]][1,1]+1)/log2(x[[i]][1,2]) #high
  log_rates[i,3]<- log2(x[[i]][2,1]+1)/log2(x[[i]][2,2]) #low
}
```

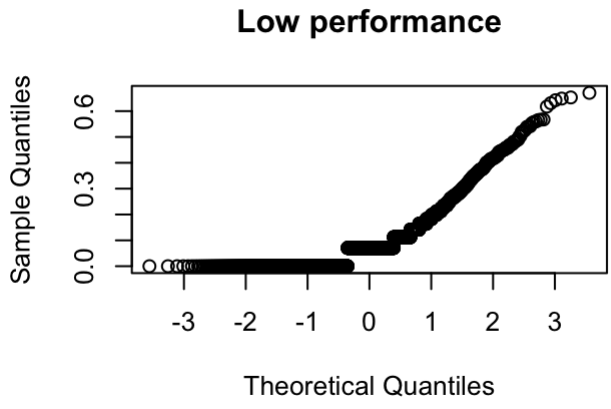
Here are a histogram and normal probability plot of the estimates for both high performance groups and low performance groups

```
par(mfrow=c(2,2))
hist(as.numeric(log_rates[,2]),main="High Performance")
qqnorm(as.numeric(log_rates[,2]),main = "High performance")
hist(as.numeric(log_rates[,3]),main="Low performance")
qqnorm(as.numeric(log_rates[,3]), main="Low performance")
```







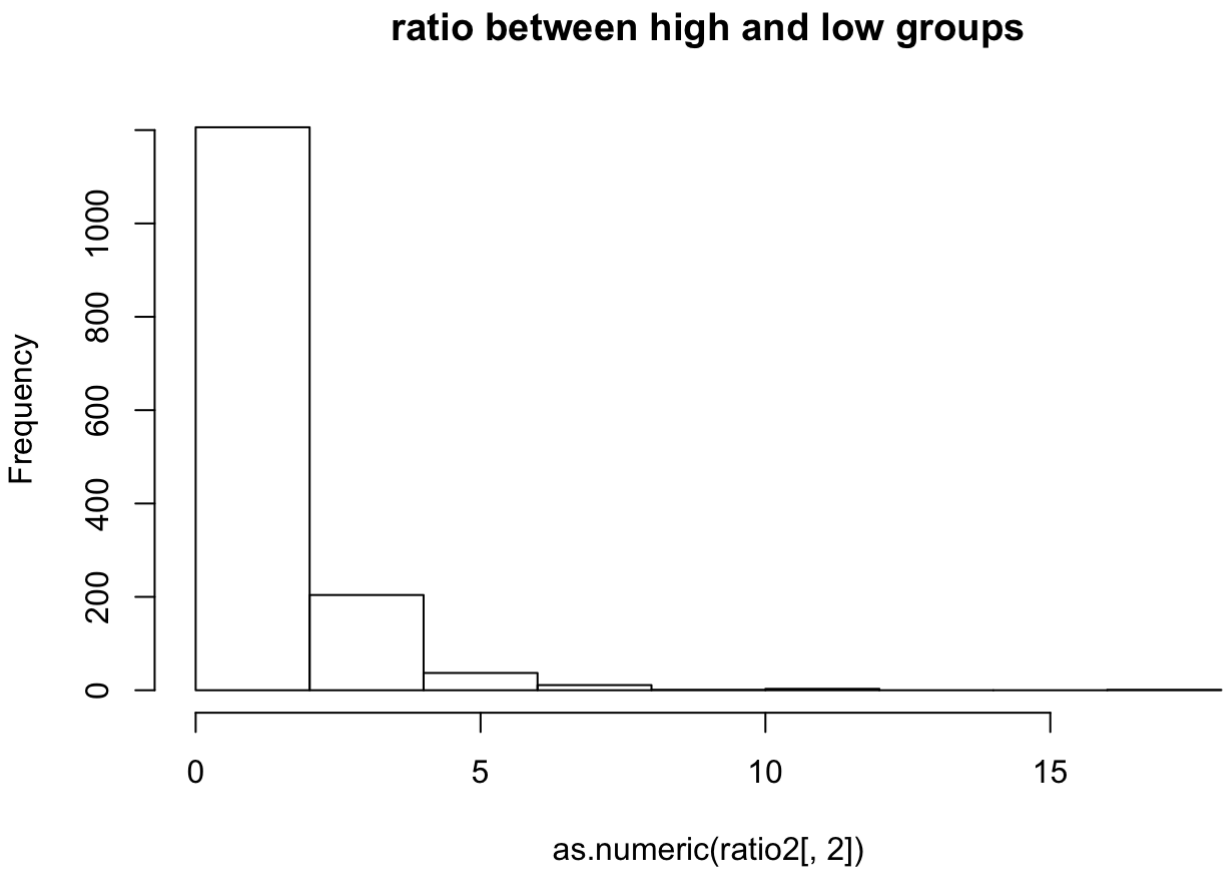


Some terms only apprear once in the dataset. This could be unreliable and not very informative. So we discard them

```
dat$tot<- rowSums(dat[,2:3])
dat2<- dat[which(dat$tot>1),]

aux2<- 1:length(dat2$term)
xx<- rep(list(diag(2)), 1463)
for (i in 1:length(aux2)){
  xx[[i]][1,1]<-dat2$high[[i]]
  xx[[i]][2,1]<-dat2$low[[i]]
  xx[[i]][1,2]<-colSums(dat2[,c(2,3)])[1]-dat2$high[[i]]
  xx[[i]][2,2]<-colSums(dat2[,c(2,3)])[2]-dat2$low[[i]]
  colnames(xx[[i]])<- c(dat2$term[i], paste0("\u00ac",dat2$term[i]))
  rownames(xx[[i]])<- c("high", "low")
}

#ratio between the high and low

ratio2<- matrix(NA,nrow=1463,ncol=2)
for (i in 1:length(xx)){
  ratio2[i,1]<- colnames(xx[[i]])[1] #high/low for the rest of terms
  ratio2[i,2]<- xx[[i]][1,1]/(xx[[i]][2,1]+1)#add 0.01 here to avoid infinite value
}
colnames(ratio2)<- c("term","ratio")
ordered_ratio2<-ratio2[order(ratio2[,2],decreasing=TRUE),]
head(ordered_ratio2,50)
```

```
##       term            ratio
##  [1,] "hundredth"     "9"
##  [2,] "circle"        "8"
##  [3,] "points"        "8"
##  [4,] "65"            "7"
##  [5,] "lowest"        "7"
##  [6,] "meant"         "7"
##  [7,] "mind"          "7"
##  [8,] "n1"            "7"
##  [9,] "order"         "7"
## [10,] "whatever"      "7"
## [11,] "squared"       "6.5"
## [12,] "ha"            "6.25"
## [13,] "243"           "6"
## [14,] "7.5"           "6"
## [15,] "9.6"           "6"
## [16,] "ef"            "6"
## [17,] "find"          "6"
## [18,] "gd"            "6"
## [19,] "greater"       "6"
## [20,] "hahaha"        "6"
## [21,] "plug"          "6"
## [22,] "shit"          "6"
## [23,] "taken"         "6"
## [24,] "looking"       "5.5"
## [25,] "8.5"           "5"
## [26,] "brain"         "5"
## [27,] "care"          "5"
## [28,] "circumference" "5"
## [29,] "closet"        "5"
## [30,] "crap"          "5"
## [31,] "formula"       "5"
## [32,] "freeze"        "5"
## [33,] "heads"         "5"
## [34,] "high"          "5"
## [35,] "hmmm"          "5"
## [36,] "job"           "5"
## [37,] "made"          "5"
## [38,] "messed"        "5"
## [39,] "needs"         "5"
## [40,] "sq"            "5"
## [41,] "tens"          "5"
## [42,] "thousands"     "5"
## [43,] "yikes"         "5"
## [44,] "we're"         "4.66666666666667"
## [45,] "hate"          "4.5"
## [46,] "shall"         "4.5"
## [47,] "looks"         "4.4"
## [48,] "85.9"          "4.33333333333333"
## [49,] "line"          "4.33333333333333"
## [50,] "13.453"        "4"
```

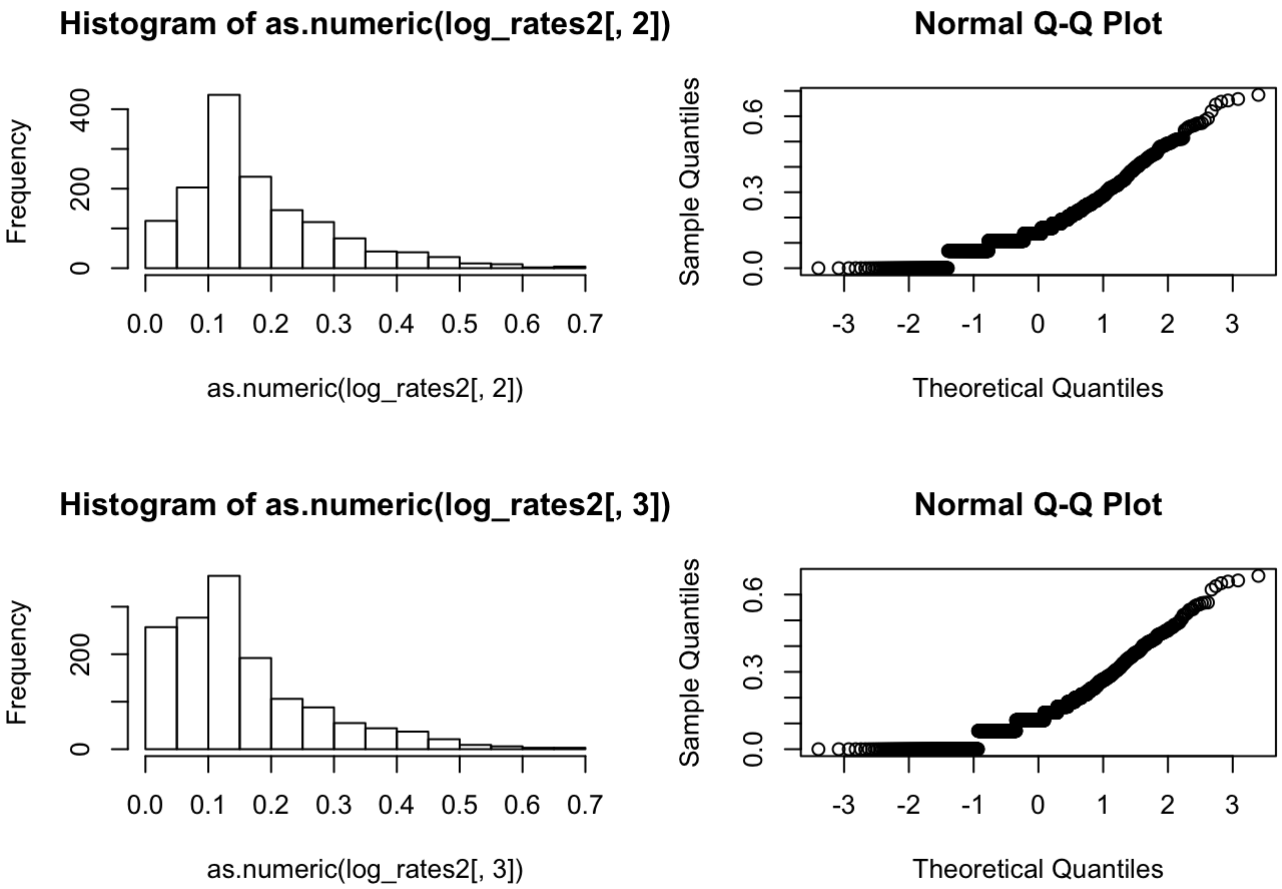
```
hist(as.numeric(ratio2[,2]), main="ratio between high and low groups")
```




```
#ratio for the term and the rest of terms

rates2<- matrix(NA, nrow = 1463, ncol = 3)
for (i in 1:length(xx)){
  rates2[i,1]<-colnames(xx[[i]])[1]
  rates2[i,2]<- xx[[i]][1,1]/(xx[[i]][1,2])
  rates2[i,3]<- xx[[i]][2,1]/(xx[[i]][2,2])
}
colnames(rates2)<- c("term","high_rates","low_rates")

#log form
log_rates2<- matrix(NA, nrow = 1463, ncol = 3)
for (i in 1:length(xx)){
  log_rates2[i,1]<-colnames(xx[[i]])[1]
  log_rates2[i,2]<- log2(xx[[i]][1,1]+1)/log2(xx[[i]][1,2])
  log_rates2[i,3]<- log2(xx[[i]][2,1]+1)/log2(xx[[i]][2,2])
}
```

Now let's view the distribution for the log ratio

```
par(mfrow=c(2,2))
hist(as.numeric(log_rates2[,2]))
qqnorm(as.numeric(log_rates2[,2]))
hist(as.numeric(log_rates2[,3]))
qqnorm(as.numeric(log_rates2[,3]))
```




```
#a little more approached to Normal distributed shape.
```

Uncertainty quantification It's hard to know which of these differences are meaningful without quantifying the error associated with the estimates. Some words are common, and we have reliable estimates of the log ratios. Other words are rare, and the estimates are based on a small number of occurrences. In the rare case, the estimates of the log ratios will be unreliable.

Estimate the standard errors

```
rates_df1<- data.frame(as.numeric(rates2[,2])) #convert to data frame
rates_df2<- data.frame(as.numeric(rates2[,3]))
rates_df3<- data.frame(rates2[,1])
rates_df<- cbind(rates_df3,rates_df1,rates_df2)
colnames(rates_df)<-c("term","high_rates","low_rates")

high_se<- sqrt(rates_df$high_rates*(1-rates_df$high_rates)/ colSums(rates_df[,2:3])[1])
#a vector

low_se<- sqrt(rates_df$low_rates*(1-rates_df$low_rates)/ colSums(rates_df[,2:3])[2])
```

To find the standard errors for the logarithms of these quantities, we use the delta method. We multiply the standard error by the absolute value of the derivative of the logarithm function evaluated at the estimate:

```
log2_high_se<- abs(1/(log(2)*rates_df$high_rates))*high_se
log2_low_se<- abs(1/(log(2)*rates_df$low_rates))*low_se
```

now assume log2_high_se and log2_low_se are independent.

```
log2_ratio_se<- sqrt(log2_high_se^2+log2_low_se^2)
```

To produce a plot

```
r<- rank(log_rates2,ties.method = "first")

xlim<- xlim<- range(r)
ylim<- range(log_rates2[,2],log_rates2[,3])
```