DAT-5317 FMAN2
A3: Business Insight Report
Vivian Soo
Dec 3rd, 2021

Business Insight Report

In this report, I will be using chat transcripts exported from my two MBAN cohort groups on WhatsApp. Those are in the format of .txt files. One chat group is named “MBAN lost in code” while the other chat group is “Business Analytics SFR21”. Both groups were created around the same time at the beginning of this school year for the same purposes.

As a virtual student, I tried my best to stay up to date with the chat contents in the beginning especially since my method of communication with my peers are limited to online. However, as time went on, through quick previews of the messages from the notifications, a lot of the conversations outside of pure code sharing seemed either irrelevant to the course or mostly catered toward in-person students. Sometimes, a lot of information were duplicated and repeated since both groups consisted of people from the same cohort. In the end, there were at least two months’ worth of conversations from both groups that I have not carefully paid attention to.

Instead of revisiting the WhatsApp groups and read the chat line by line, I would like to use this opportunity to set up my business objective which is to first understand the gist of the chats, then decide if the contents suggest that I should invest more time and pay closer attention to them in the future.

Challenges

WhatsApp is not particularly user-friendly for text analytics purposes. Much noise was generated from WhatsApp’s frequent systemic message such as “[10/5/21, 12:15:11 PM] +1 (XXX) 451-XXXX joined using this group's invite link” in the txt file. Moreover, WhatsApp automatically converts any images that were attached in the original chat to “image omitted” in words. And any other media attached such as stickers and gifs also generated further noise for the data.

Example: "<attached: 00000809-STICKER-2021-11-10-17-34-57.webp>".

Although slang words were expected in a chat setting, this resulted in an immense amount of unique word tokens, such as "hahahaah", "haha", "yeahhhhh", "lmk" and more.

Addressing the Challenges

After separating the text into individual word tokens, I removed all numbers using the gsub() function. This is to reduce noise generated from WhatsApp's frequent systemic messages.

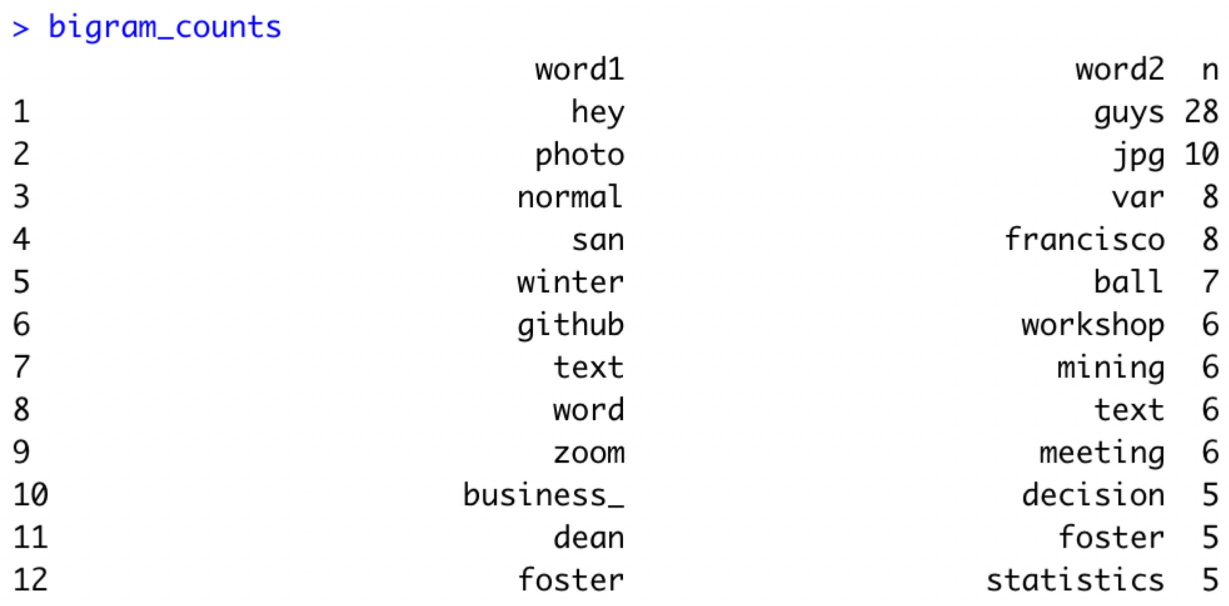
Next, after removing the default stop words, in the same filter function I also added a list of "junk words" that appeared frequently that delivered no business insights. Some of these words are "joined", "using", "this", "group's", "image", "omitted" and more.

Some technical terms such as codes and jargon are also removed as they are usually R scripts shared by peers.

Although not exhaustive, the list of "junk words" to be filtered grew quickly as the analyses continued.

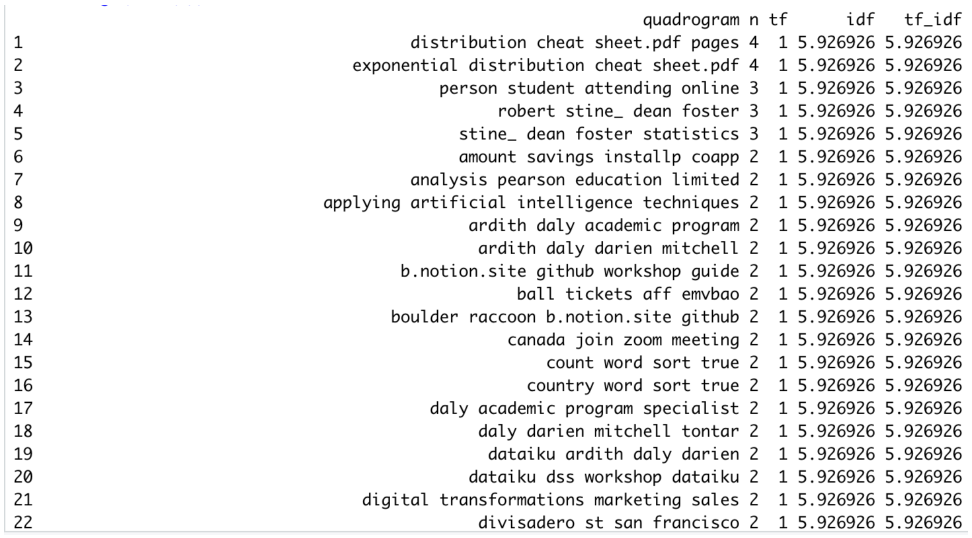
Frameworks

Bigram. By combining individual tokens into bigrams, I can understand the order of words that they come in. The most common two-word phrase is "hey guys" with 28 appearances followed by other common phrases such as "san francisco" and "winter ball". Although many stop words and junk words have already been filtered, there are still more than 1600 rows of bigrams in total.



```
> bigram_counts
                                    word1                         word2  n
1                                     hey                          guys 28
2                                   photo                           jpg 10
3                                  normal                           var  8
4                                     san                     francisco  8
5                                  winter                          ball  7
6                                  github                      workshop  6
7                                    text                        mining  6
8                                    word                          text  6
9                                    zoom                       meeting  6
10                              business_                      decision  5
11                                   dean                        foster  5
12                                 foster                    statistics  5
```

Quadrogram. Next, to gain more contextual information, the number of tokens together increased to four. With some scrutinization, there are some notable quadrograms that raise interest such as “daly academic program specialist”, “digital transformations marketing sales”, “dress fancy dress bougie”, “dynamic thinking embraces change”, etc. Having been somewhat familiar with the past conversations, these phrases suggest something about internship opportunities, winter ball event, and occasional peers’ sharing of opinions. However, there still existed some R-related jargon that added noise.



```
                                           quadrogram n tf      idf   tf_idf
1                  distribution cheat sheet.pdf pages 4  1 5.926926 5.926926
2               exponential distribution cheat sheet.pdf 4  1 5.926926 5.926926
3                        person student attending online 3  1 5.926926 5.926926
4                              robert stine_ dean foster 3  1 5.926926 5.926926
5                          stine_ dean foster statistics 3  1 5.926926 5.926926
6                          amount savings installp coapp 2  1 5.926926 5.926926
7                    analysis pearson education limited 2  1 5.926926 5.926926
8          applying artificial intelligence techniques 2  1 5.926926 5.926926
9                           ardith daly academic program 2  1 5.926926 5.926926
10                           ardith daly darien mitchell 2  1 5.926926 5.926926
11                 b.notion.site github workshop guide 2  1 5.926926 5.926926
12                              ball tickets aff emvbao 2  1 5.926926 5.926926
13               boulder raccoon b.notion.site github 2  1 5.926926 5.926926
14                             canada join zoom meeting 2  1 5.926926 5.926926
15                                count word sort true 2  1 5.926926 5.926926
16                              country word sort true 2  1 5.926926 5.926926
17                    daly academic program specialist 2  1 5.926926 5.926926
18                        daly darien mitchell tontar 2  1 5.926926 5.926926
19                          dataiku ardith daly darien 2  1 5.926926 5.926926
20                        dataiku dss workshop dataiku 2  1 5.926926 5.926926
21              digital transformations marketing sales 2  1 5.926926 5.926926
22                          divisadero st san francisco 2  1 5.926926 5.926926
```

In addition, I used the igraph and ggraph function to create two visualizations on the relationship between words. Ggraph1 in appendix limits to n greater than 1 whereas Ggraph2 is set to equal to 1 for the more unique quadrograms. Ggraph1 is more readable but has relatively fewer information compared to Ggraph2.

LDA. Finally, an LDA graph is created to see the probability of a certain word that belongs to either topic 1 or 2. LDA graph from Appendix suggests that words such as “exam”, “tomorrow”, “feel”, “email” and more belong to one topic, while words such as “gif”, “zoom”, “join”, “lol” belong to the other. This makes some sense as the first group seems to have a more serious tone whereas the second group is more casual.

Result and Conclusion

With some of the technical words removed, more conversational rather than code-related technical information was revealed. However, the important tradeoff is that the more information removed, the less data granularity we have for the analysis.

Second, although less term frequency usually derives more business value, in this dataset, however, it was impossible to read through thousands of unique tokens as a result of prominent use of slang and typos in casual chats. Instead, this analysis would be more suitable in a group that has a more professional atmosphere with proper use of word choices. These analyses suggest that even though not reading through every text may result in missed opportunities such as internship opportunities and important messages shared by peers, the frequencies of such incidence are comparably low.

Getting up-to-date information from the current WhatsApp group chats would be less crucial than if it were from live Zoom chat, Announcements in MyCourses, or school emails. As a graduate student, we not only have limited time with different obligations in a day but must manage online communications across multiple platforms. Therefore, I believe I am not the only one who rarely reads through every single message from the group chats. Therefore, if I wish to gather my peers' attention and hope for quicker responses such as a time-limited survey for example, I should not rely on the group chat as my only channel for communication.

Appendix

```
library(textreadr)
library(tm)
library(dplyr)
library(magrittr)
library(stringi)
library(tidyverse)
library(tidytext)

setwd("/Users/tsztinviviansoo/Desktop/Text analytics/whatsapp")
nm <- list.files(path="/Users/tsztinviviansoo/Desktop/Text analytics/
whatsapp")
my_data <- read_document(file=nm[1])
my_data_together <- paste(my_data, collapse = " ")

View(my_data_together)
my_txt_text <- do.call(rbind, lapply(nm, function(x)
paste(read_document(file=x), collapse = " ")))
my_txt_df <- as.data.frame(my_txt_text)
colnames(my_txt_df)[1] <- "text"

#################
### txt_token ###
#################
txt_token <- my_txt_df %>%
  mutate(my_txt_df, text = gsub(x = text, pattern = "[0-9]", replacement
= "")) %>%
  unnest_tokens(word, text) %>%
  filter(!word %in% c(stop_words$word, "pm","time","free","attached",
                      "message","document","class","hult", "image",
                      "omitted", "deleted", "sticker", "age_standard",
                       "invite","joined","it’s","i’m","link","group's","https",
                       "library","tidytuesday","netflix","checking","data",
                       "binary","checking_norm","duration_norm","age_norm",
                       "germ_train","good_bad","my_logit","my_logit_norm",
                       "unnest_tokens","geom_jitter","duration",
                       "german_credit_card","anti_join","ggplot")) %>%
  #removing more noise in addition to stop words
  count(word, sort=TRUE)
#filter is the alternative to anti_join according to class
View(txt_token)

txt_token%>%
  filter(word=="happy")
# the frequency of word "happy" is 8
txt_token%>%
  filter(word=="love")

##############
### BIGRAM ###
##############

txt_bigrams <- my_txt_df %>%
  mutate(my_txt_df, text = gsub(x = text, pattern = "[0-9]", replacement
= "")) %>%
  #ommitted numbers because random phone numbers popped up too frequently
  unnest_tokens(bigram, text, token = "ngrams", n=2) %>%
  count(bigram, sort = TRUE)
View(txt_bigrams)
```

```
library(tidyr)

bigrams_separated <- txt_bigrams %>%
  separate(bigram, c("word1", "word2"), sep = " ")

bigrams_filtered <- bigrams_separated %>%
  filter(!word1 %in% c(stop_words$word, "pm","time","free","attached",
                       "message","document","class","hult", "image",
                       "omitted", "deleted", "sticker", "age_standard",
                       "invite","joined","it’s","i’m","link","group's","https",
                       "library","tidytuesday","netflix","checking","data",
                       "binary","checking_norm","duration_norm","age_norm",
                       "germ_train","good_bad","my_logit","my_logit_norm",
                       "unnest_tokens","geom_jitter","duration",
                       "german_credit_card","anti_join","ggplot")) %>%
  filter(!word2 %in% c(stop_words$word, "pm","time","free","attached",
                       "message","document","class","hult", "image",
                       "omitted", "deleted", "sticker", "age_standard",
                       "invite","joined","it’s","i’m","link","group's","https",
                       "library","tidytuesday","netflix","checking","data",
                       "binary","checking_norm","duration_norm","age_norm",
                       "germ_train","good_bad","my_logit","my_logit_norm",
                       "unnest_tokens","geom_jitter","duration",
                       "german_credit_card","anti_join","ggplot"))



bigram_counts <- bigrams_filtered %>%
  count(word1, word2, sort = TRUE)

View(bigram_counts)


##################
### QUADROGRAM ###
##################
quadrogram <- my_txt_df %>%
  mutate(my_txt_df, text = gsub(x = text, pattern = "[0-9]", replacement
= "")) %>%
  unnest_tokens(quadrogram, text, token = "ngrams", n=4) %>%
  separate(quadrogram, c("word1", "word2", "word3", "word4"), sep=" ")
%>%
  filter(!word1 %in% c(stop_words$word, "pm","time","free","attached",
                      "message","document","class","hult", "image",
                      "omitted", "deleted", "sticker", "age_standard",
                      "invite","joined","it’s","i’m","link","group's","https",
                      "library","tidytuesday","netflix","checking","data",
                      "binary","checking_norm","duration_norm","age_norm",
                      "germ_train","good_bad","my_logit","my_logit_norm",
                      "unnest_tokens","geom_jitter","duration",
                      "german_credit_card","anti_join","ggplot")) %>%
  filter(!word2 %in% c(stop_words$word, "pm","time","free","attached",
                       "message","document","class","hult", "image",
                       "omitted", "deleted", "sticker", "age_standard",
                       "invite","joined","it’s","i’m","link","group's","https",
                       "library","tidytuesday","netflix","checking","data",
                       "binary","checking_norm","duration_norm","age_norm",
                       "germ_train","good_bad","my_logit","my_logit_norm",
                       "unnest_tokens","geom_jitter","duration",
                       "german_credit_card","anti_join","ggplot")) %>%
  filter(!word3 %in% c(stop_words$word, "pm","time","free","attached",
                       "message","document","class","hult", "image",
                       "omitted", "deleted", "sticker", "age_standard",
                       "invite","joined","it’s","i’m","link","group's","https",
                       "library","tidytuesday","netflix","checking","data",
                       "binary","checking_norm","duration_norm","age_norm",
                       "germ_train","good_bad","my_logit","my_logit_norm",
                       "unnest_tokens","geom_jitter","duration",
                       "german_credit_card","anti_join","ggplot")) %>%
  filter(!word4 %in% c(stop_words$word, "pm","time","free","attached",
                       "message","document","class","hult", "image",
                       "omitted", "deleted", "sticker", "age_standard",
                       "invite","joined","it’s","i’m","link","group's","https",
                       "library","tidytuesday","netflix","checking","data",
                       "binary","checking_norm","duration_norm","age_norm",
                       "germ_train","good_bad","my_logit","my_logit_norm",
                       "unnest_tokens","geom_jitter","duration",
                       "german_credit_card","anti_join","ggplot"))
quadrogram

quadrogram_united <- quadrogram %>%
  unite(quadrogram, word1, word2, word3, word4, sep=" ")
View(quadrogram_united)
quadrogram_tf_idf <- quadrogram_united %>%
  count(quadrogram) %>%
  bind_tf_idf(quadrogram, quadrogram, n) %>%
  arrange(desc(n))#here i changed desc(tf_idf) to desc(n) since for some
reason,
#all the idf and tf_idf were the same....


quad_counts <- quadrogram %>%
  count(word1, word2, word3, word4, sort = TRUE)

View(quad_counts)
##################

word_cors %>%
  filter(item1 == "love")
##################
##################################
### BIGRAM AND QUADGRAM GRAPHS ###
##################################
library(igraph)
bigram_graph <- bigram_counts %>%
  filter(n>3) %>%
  graph_from_data_frame()

bigram_graph


library(ggraph)
ggraph(bigram_graph, layout = "fr") +
  geom_edge_link()+
  geom_node_point()+
  geom_node_text(aes(label=name), vjust =1, hjust=1)

quad_graph <- quad_counts %>%
  filter(n>1) %>%
  graph_from_data_frame()


ggraph(quad_graph, layout = "fr") +
  geom_edge_link()+
  geom_node_point()+
  geom_node_text(aes(label=name), vjust =1, hjust=1)

quad_graph2 <- quad_counts %>%
  filter(n==1) %>%
  graph_from_data_frame()

ggraph(quad_graph2, layout = "fr") +
  geom_edge_link()+
  geom_node_point()+
  geom_node_text(aes(label=name), vjust =1, hjust=1)
###################
### CORRELATION ###
###################
word_cors <- txt_token %>%
  group_by(word) %>%
  filter(n>=30) %>%
  pairwise_cor(word,txt_token, sort=TRUE)
word_cors
View(word_cors)

###
my_txt_dtm <- my_txt_df %>%
  mutate(my_txt_df, text = gsub(x = text, pattern = "[0-9]", replacement
= "")) %>%
  unnest_tokens(word, text) %>%
  filter(!word %in% c(stop_words$word, "pm","time","free","attached",
                      "message","document","class","hult", "image",
                      "omitted", "deleted", "sticker", "age_standard",
                      "invite","joined","it’s","i’m","link","group's","https",
                      "library","tidytuesday","netflix","checking","data",
                      "binary","checking_norm","duration_norm","age_norm",
                      "germ_train","good_bad","my_logit","my_logit_norm",
                      "unnest_tokens","geom_jitter","duration",
                      "german_credit_card","anti_join","ggplot")) %>%
  count(word, word) %>%
  cast_dtm(word, word, n)

my_txt_dtm

###########
### LDA ###
###########
#now that we have a dtm, we can use LDA to find topics
library(topicmodels)
txt_lda <- LDA(my_txt_dtm, k=2, control=list(seed=123))
```

```
library(tidytext)
txt_topics <- tidy(txt_lda, matrix="beta")
txt_topics

top_terms <- txt_topics %>%
  group_by(topic) %>%
  top_n(10, beta) %>%
  ungroup() %>%
  arrange(topic, -beta)
View(top_terms)

top_terms %>%
  mutate(term=reorder(term, beta)) %>%
  ggplot(aes(term, beta, fill = factor(topic))) +
  geom_col(show.legend=FALSE) +
  facet_wrap(~topic, scales = "free") +
  coord_flip()
#topic on the left seems more casual whereas topic on the right more
serious

beta_spread <- txt_topics %>%
  mutate(topic=paste0("topic", topic)) %>%
  spread(topic, beta) %>%
  filter(topic1>0.001 | topic2 >0.001) %>%
  mutate(log_rate = log2(topic2/topic1))

View(beta_spread)
#########GAMMA
chapters_gamma <- tidy(txt_lda, matrix="gamma") #we created ap_lda in our
LDA scripts
View(chapters_gamma)
str(chapters_gamma)
gamma_df<- as.data.frame(chapters_gamma) %>%
  arrange(desc(gamma))

write.csv(gamma_df, file = '/Users/tsztinviviansoo/Desktop/Text
analytics/gammadf.csv', row.names = FALSE)
#I exported this table to a csv file
########


mutate(my_txt_df, text = gsub(x = text, pattern = "[0-9]", replacement =
"")) %>%
  unnest_tokens(word, text) %>%
  filter(!word %in% c(stop_words$word, "pm","time","free","attached",
                      "message","document","class","hult", "image",
                      "omitted", "deleted", "sticker", "age_standard",
                      "invite","joined","it’s","i’m","link","group's","https",
                      "library","tidytuesday","netflix","checking","data",
                      "binary","checking_norm","duration_norm","age_norm",
                      "germ_train","good_bad","my_logit","my_logit_norm",
                      "unnest_tokens","geom_jitter","duration",
                      "german_credit_card","anti_join","ggplot")) %>%
#removing pm in addition to stop words
  count(word, sort=TRUE)

############
```

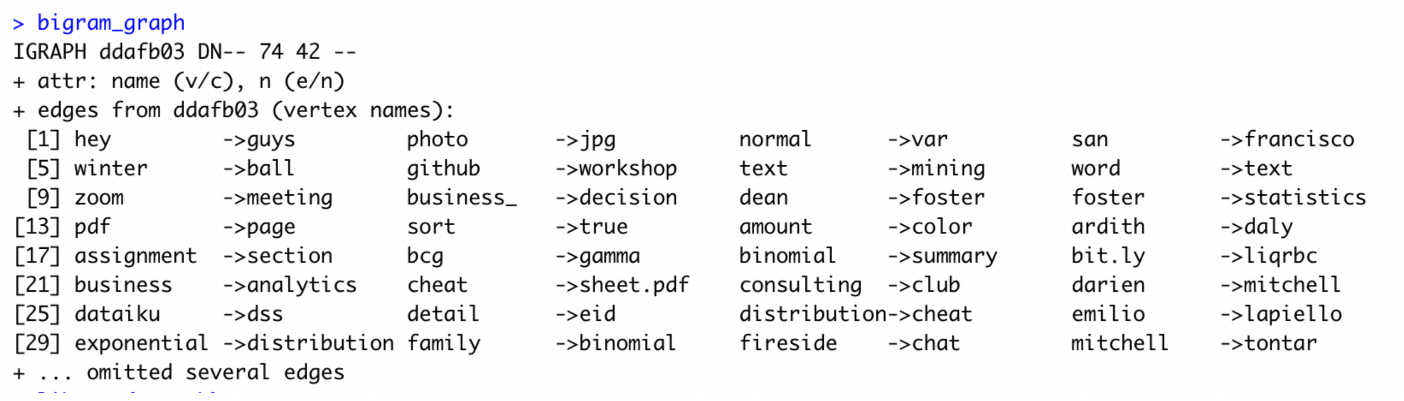
```
my_txt_df %>%
  mutate(my_txt_df, text = gsub(x = text, pattern = "[0-9]", replacement
= "")) %>%
  unnest_tokens(word, text) %>%
#################
### Frequency ###
#################
library(ggplot2)
freq_hist <- my_txt_df %>%
  mutate(my_txt_df, text = gsub(x = text, pattern = "[0-9]", replacement
= "")) %>%
  unnest_tokens(word,text) %>%
  filter(!word %in% c(stop_words$word, "pm","time","free","attached",
                      "message","document","class","hult", "image",
                      "omitted", "deleted", "sticker", "age_standard",
                      "invite","joined","it’s","i’m","link","group's","https",
                      "library","tidytuesday","netflix","checking","data",
                      "binary","checking_norm","duration_norm","age_norm",
                      "germ_train","good_bad","my_logit","my_logit_norm",
                      "unnest_tokens","geom_jitter","duration",
                      "german_credit_card","anti_join","ggplot")) %>%
  count(word, sort=TRUE) %>%
  filter(n>15) %>%
  mutate(word=reorder(word, n)) %>%
  ggplot(aes(word, n))+
  geom_col()+
  xlab(NULL)+
  coord_flip()
print(freq_hist)

#total words
total_words <- txt_token %>%
  summarize(total=sum(n))

txt_words <- c(txt_token, total_words)
txt_words <- as.data.frame(txt_words)

freq_by_rank <- txt_words %>%
  mutate(rank = row_number(),
         `term frequency`=n/total) %>%

View(freq_by_rank)


freq_by_rank %>%
  ggplot(aes(rank, `term frequency`))+
  #let's add a tangent line , the first derivative, and see what the slop
is
  geom_abline(intercept=-0.62, slope= -1.1, color='gray50', linetype=2)+
  geom_line(size= 1.1, alpha = 0.8, show.legend = FALSE)+
  scale_x_log10()+
  scale_y_log10()


txt_tfidf_words <- txt_words %>%
  bind_tf_idf(word, word, n) %>%
  arrange(desc(tf_idf))
```

```r
txt_tfidf_words %>%
  arrange(desc(tf_idf)) %>%
  mutate(word=factor(word, levels=rev(unique(word)))) %>%
  top_n(15) %>%
  ungroup %>%
  ggplot(aes(word, tf_idf))+
  geom_col(show.legend=FALSE)+
  labs(x=NULL, y="tf-idf")+
  facet_wrap(~word, ncol=2, scales="free")+
  coord_flip()
```

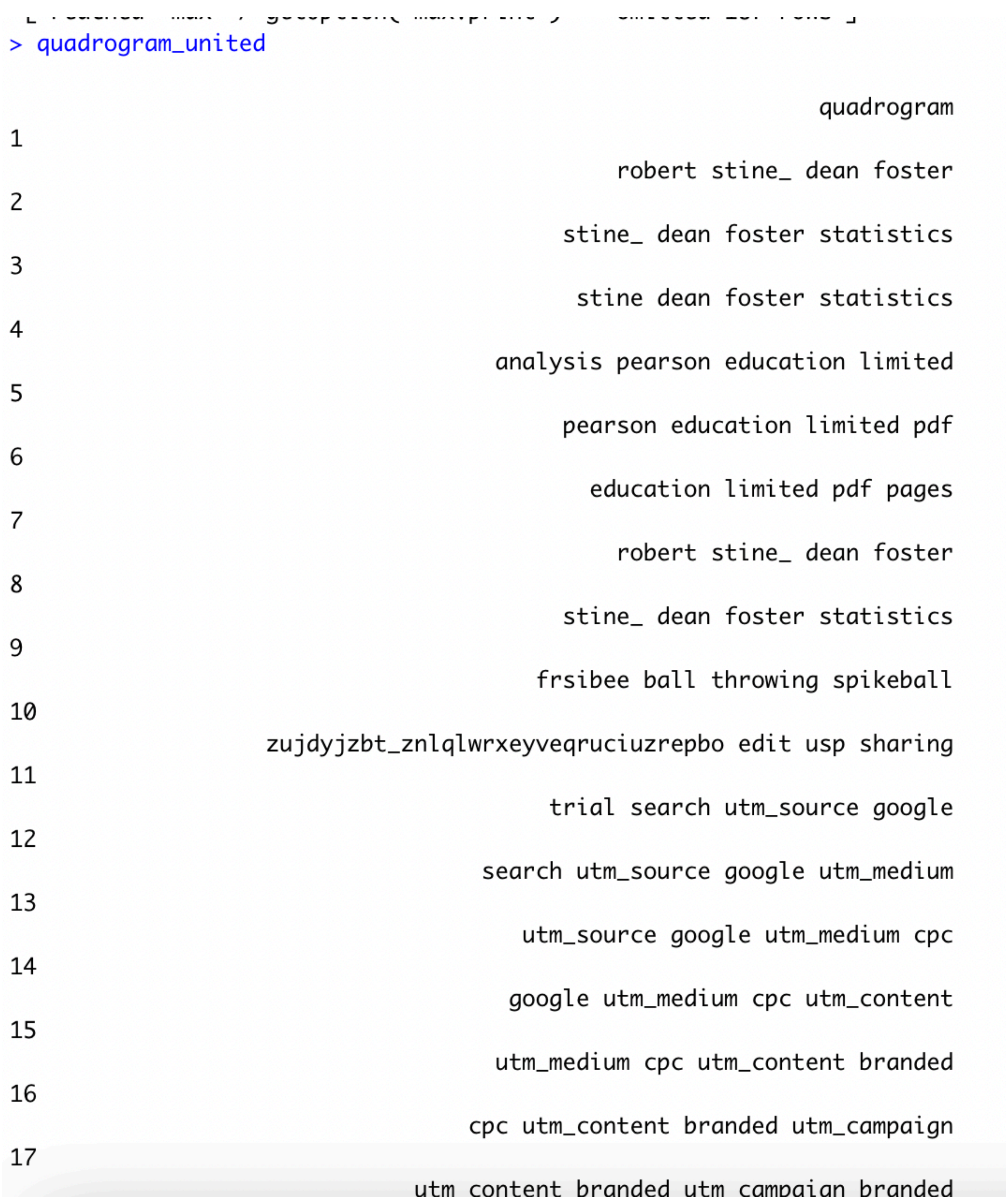
```r
word_cors <- my_txt_df %>%
  mutate(my_txt_df, text = gsub(x = text, pattern = "[0-9]", replacement
= "")) %>%
  unnest_tokens(word, text) %>%
  filter(!word %in% c(stop_words$word, "pm")) %>%
  group_by(word) %>%
  filter(n() >= 8) %>%
  pairwise_cor(word, word)

word_cors %>%
  filter(item1 == "love")
```

```r
word_cors %>%
  filter(correlation <1) %>%
  graph_from_data_frame() %>%
  ggraph(layout = "fr")+
  geom_edge_link(aes(edge_alpha=correlation), show.legend=F)+
  geom_node_point(color = "lightgreen", size=6)+
  geom_node_text(aes(label=name), repel=T)+
  theme_void()


###sentiments
library(textdata)
library(tidytext)
afinn <- get_sentiments("afinn")
nrc <- get_sentiments("nrc")
bing <- get_sentiments("bing")
View(sentiments)
sentiments <- bind_rows(mutate(afinn, lexicon="afinn"),
                        mutate(nrc, lexicon= "nrc"),
                        mutate(bing, lexicon="bing")
)
```

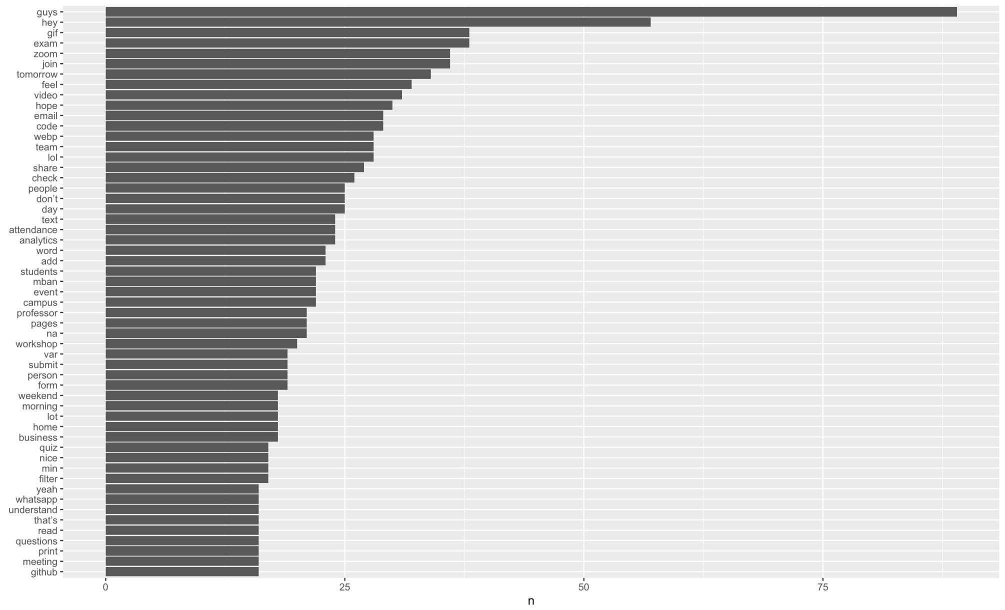
```r
nrcsurprise <- get_sentiments("nrc") %>%
  filter(sentiment == "surprise")
nrcanger <- get_sentiments("nrc") %>%
  filter(sentiment == "anger")
nrcdisgust <- get_sentiments("nrc") %>%
  filter(sentiment =="disgust")
nrcfear <- get_sentiments("nrc") %>%
  filter(sentiment =="fear")
nrctrust <- get_sentiments("nrc") %>%
  filter(sentiment =="trust")


txt_token %>%
  inner_join(nrctrust) %>%
  count(word, sort=T)

txt_token %>%
  inner_join(nrcsurprise) %>%
  count(word, sort=T)

txt_token %>%
  inner_join(nrcanger) %>%
  count(word, sort=T)

txt_token %>%
  inner_join(nrcdisgust) %>%
  count(word, sort=T)

txt_token %>%
  inner_join(nrcfear) %>%
  count(word, sort=T)



afinn <- txt_token %>%
  inner_join(get_sentiments("afinn"))%>%
  summarise(sentiment=sum(value)) %>%
  mutate(method="AFINN")

bing_and_nrc <- bind_rows(
  txt_token%>%
    inner_join(get_sentiments("bing"))%>%
    mutate(method = "Bing et al."),
  txt_token %>%
    inner_join(get_sentiments("nrc") %>%
                 filter(sentiment %in% c("positive", "negative"))) %>%
    mutate(method = "NRC")) %>%
  count(method,  sentiment) %>%
  spread(sentiment, n, fill=0) %>%
  mutate(sentiment = positive-negative)

bing_and_nrc

bind_rows(afinn, bing_and_nrc) %>%
  ggplot(aes(method, sentiment, fill=method))+
  geom_col(show.legend=FALSE)+
  facet_wrap(~method, ncol =1, scales= "free_y")

bing_counts <- txt_token %>%
  inner_join(get_sentiments("bing")) %>%
  top_n(10) %>%
  count(word, sentiment, sort=T) %>%
  ungroup()

bing_counts
summary(bing_counts)




bing_counts %>%
  group_by(sentiment) %>%
  top_n(10) %>%
  ungroup() %>%
  mutate(word=reorder(word, n)) %>%
  ggplot(aes(word, n, fill=sentiment)) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~sentiment, scales = "free_y")+
  labs(y="Contribution to sentiment", x=NULL)+
  coord_flip()
```

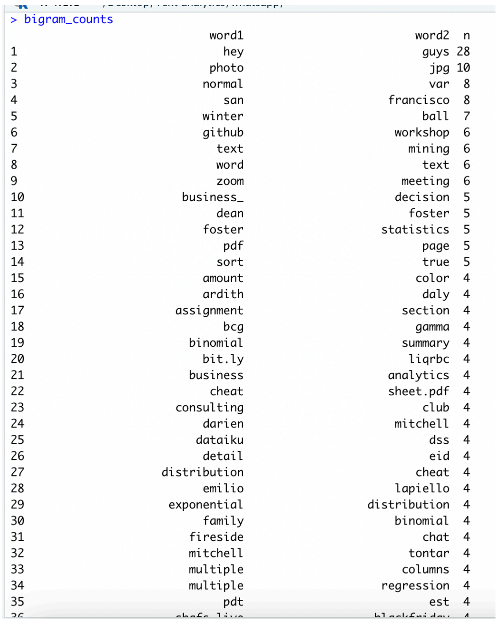


```
> bigram_graph
IGRAPH ddafb03 DN-- 74 42 --
+ attr: name (v/c), n (e/n)
+ edges from ddafb03 (vertex names):
 [1] hey         ->guys         photo       ->jpg          normal      ->var          san         ->francisco
 [5] winter      ->ball         github      ->workshop     text        ->mining       word        ->text
 [9] zoom        ->meeting      business_   ->decision     dean        ->foster       foster      ->statistics
[13] pdf         ->page         sort        ->true         amount      ->color        ardith      ->daly
[17] assignment  ->section      bcg         ->gamma        binomial    ->summary      bit.ly      ->liqrbc
[21] business    ->analytics    cheat       ->sheet.pdf    consulting  ->club         darien      ->mitchell
[25] dataiku     ->dss          detail      ->eid          distribution->cheat        emilio      ->lapiello
[29] exponential ->distribution family      ->binomial     fireside    ->chat         mitchell    ->tontar
+ ... omitted several edges
```

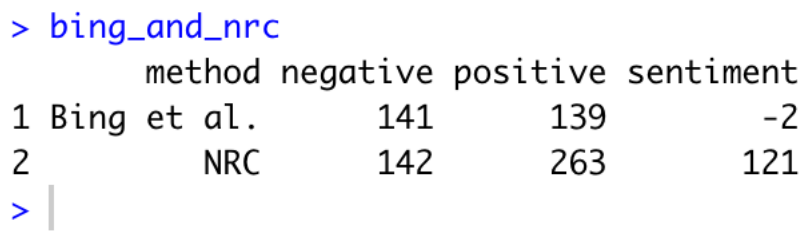


```
 [ reached 'max' / getOption("max.print") -- omitted 15 rows ]
> quadrogram_united
                                                                quadrogram
1
                                                   robert stine_ dean foster
2
                                               stine_ dean foster statistics
3
                                                stine dean foster statistics
4
                                         analysis pearson education limited
5
                                               pearson education limited pdf
6
                                                education limited pdf pages
7
                                                   robert stine_ dean foster
8
                                               stine_ dean foster statistics
9
                                             frsibee ball throwing spikeball
10
                     zujdyjzbt_znlqlwrxeyveqruciuzrepbo edit usp sharing
11
                                             trial search utm_source google
12
                                        search utm_source google utm_medium
13
                                           utm_source google utm_medium cpc
14
                                          google utm_medium cpc utm_content
15
                                        utm_medium cpc utm_content branded
16
                                      cpc utm_content branded utm_campaign
17
                                  utm content branded utm campaign branded
```

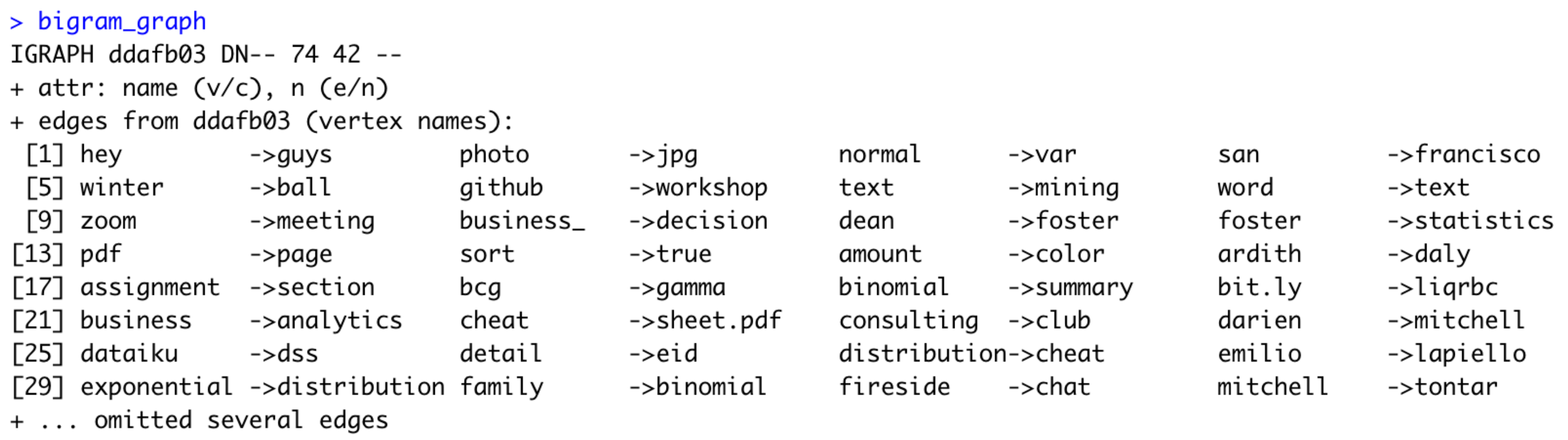

guys
hey
gif
exam
zoom
join
tomorrow
feel
video
hope
email
code
webp
team
lol
share
check
people
don't
day
text
attendance
analytics
word
add
students
mban
event
campus
professor
pages
na
workshop
var
submit
person
form
weekend
morning
lot
home
business
quiz
nice
min
filter
yeah
whatsapp
understand
that's
read
questions
print
meeting
github
0
25
50
75
n



```
> bigram_counts
                                   word1                          word2  n
1                                    hey                           guys 28
2                                  photo                            jpg 10
3                                 normal                            var  8
4                                    san                      francisco  8
5                                 winter                           ball  7
6                                 github                       workshop  6
7                                   text                         mining  6
8                                   word                           text  6
9                                   zoom                        meeting  6
10                             business_                       decision  5
11                                  dean                         foster  5
12                                foster                     statistics  5
13                                   pdf                           page  5
14                                  sort                           true  5
15                                amount                          color  4
16                                ardith                           daly  4
17                            assignment                        section  4
18                                   bcg                          gamma  4
19                              binomial                        summary  4
20                                bit.ly                         liqrbc  4
21                              business                      analytics  4
22                                 cheat                      sheet.pdf  4
23                            consulting                           club  4
24                                darien                       mitchell  4
25                               dataiku                            dss  4
26                                detail                            eid  4
27                          distribution                          cheat  4
28                                emilio                       lapiello  4
29                           exponential                   distribution  4
30                                family                       binomial  4
31                              fireside                           chat  4
32                              mitchell                         tontar  4
33                              multiple                        columns  4
34                              multiple                     regression  4
35                                   pdt                            est  4
36                            chefs_live                    blackfriday  4
```

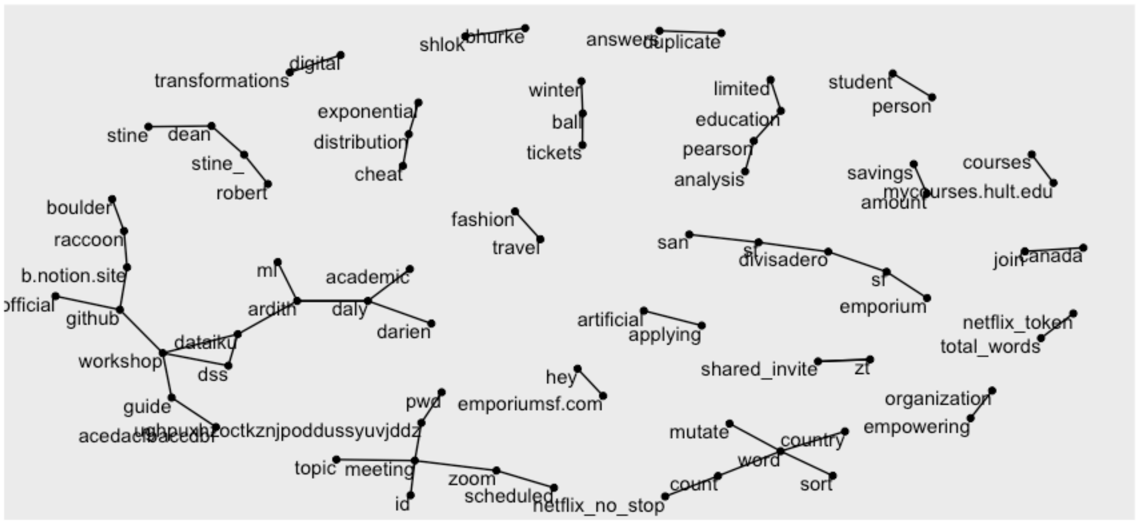


```
> bing_and_nrc
       method negative positive sentiment
1 Bing et al.      141      139        -2
2         NRC      142      263       121
> 
```

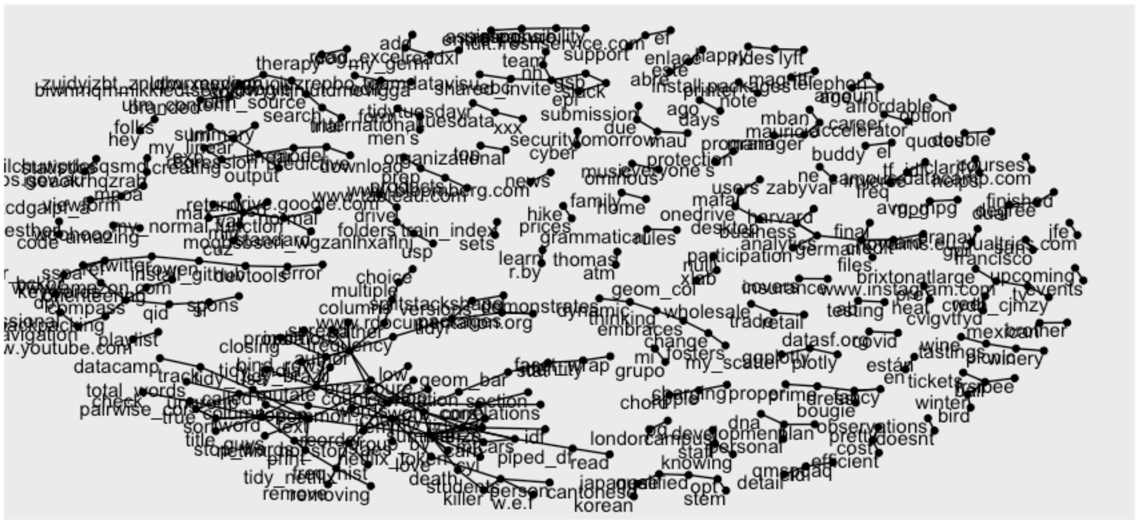


```
> bigram_graph
IGRAPH ddafb03 DN-- 74 42 --
+ attr: name (v/c), n (e/n)
+ edges from ddafb03 (vertex names):
 [1] hey         ->guys         photo       ->jpg          normal      ->var          san         ->francisco
 [5] winter      ->ball         github      ->workshop     text        ->mining       word        ->text
 [9] zoom        ->meeting      business_   ->decision     dean        ->foster       foster      ->statistics
[13] pdf         ->page         sort        ->true         amount      ->color        ardith      ->daly
[17] assignment  ->section      bcg         ->gamma        binomial    ->summary      bit.ly      ->liqrbc
[21] business    ->analytics    cheat       ->sheet.pdf    consulting  ->club         darien      ->mitchell
[25] dataiku     ->dss          detail      ->eid          distribution->cheat        emilio      ->lapiello
[29] exponential ->distribution family      ->binomial     fireside    ->chat         mitchell    ->tontar
+ ... omitted several edges
```

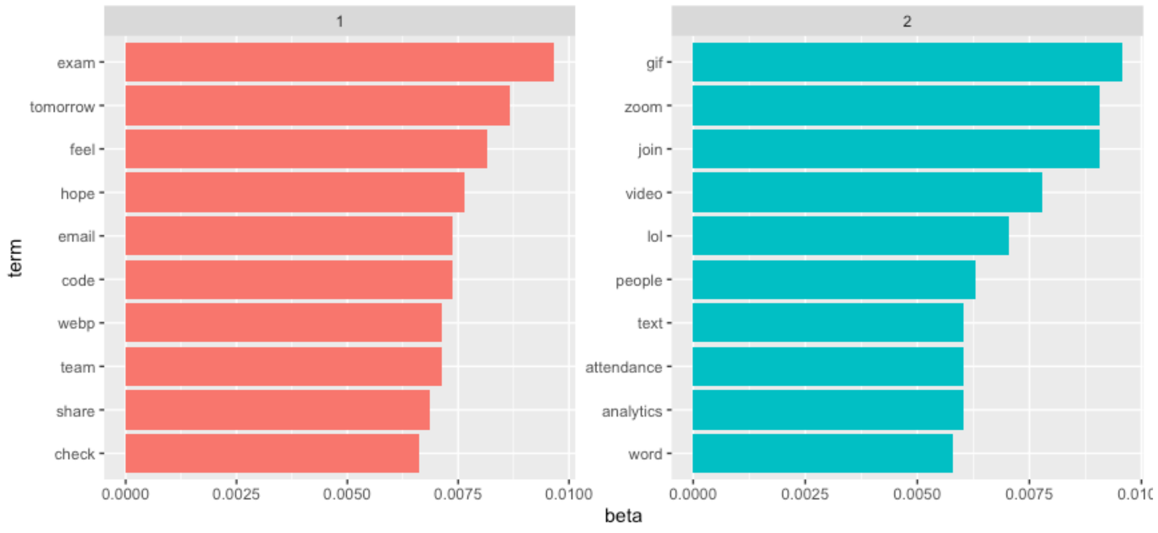
Ggraph1



Ggraph2



LDA Graph


1
2
term
exam
tomorrow
feel
hope
email
code
webp
team
share
check
gif
zoom
join
video
lol
people
text
attendance
analytics
word
0.0000
0.0025
0.0050
0.0075
0.0100
0.010
beta