AGENCE NATIONALE DE LA STATISTIQUE ET DE LA DEMOGRAPHIE

ECOLE NATIONALE DE LA STATISTIQUE ET DE L'ANALYSE ECONOMIQUE

PROJET_STATISTIQUE_R

Membres du groupe :

TANGOUO KUETE Ivana

&

AMADOU Moussa

&

GNING IBRAHIMA

Sous la supervision de :

HEMA Aboubacar

Analyst research

Notre travail consiste à faire le traitement de texte en utilisant les techniques statistiques. Le textmining permet d’extraire des connaissances significatives à partir d’un corpus ou d’un texte. Pour faire le textmining il est nécessaire de :

# Cas d’une base de donnée

1- pretraitement du texte
2- frequence des mots ou des ngrammes
3- matrice terme des mots
4- nuage de mots
5- le reseau des mots

# Cas de messages whatsapp ,facebook ,twitter ou livre

1- Chronologie des messages
2- classement des ID selon le nombre de messsage envoyé
3- les mots les plus fréquents utilisés par chaque ID
4- comparaison des mots les plus fréquents

Packages necessaires

1- pretraitement du texte
les packages necessaires pour faire le prétraitement sont :

```
library(tidytext)
library(dplyr)
library(tm)
```

tidytext et dplyr

Fonctions necessaires
tidytext::unnest_tokens(tbl,output,input,token,format,to_lower) tidytext:: stop_words()
dplyr:: anti_join()
dplyr:: filter()
dplyr:: count()
dplyr:: mutate()
dplyr:: filter()
dplyr:: anti_joint() dplyr:: tibble()

tm

```
tm:: tm_map(txte1, removePunctuation)
tm:: tm_map(txte1, removeNumbers)
tm:: tm_map(txte1, removeWords, stopwords("english"))
tm:: tm_map(txte1, stripWhitespace)
tm:: tm_map(txte1, tm_reduce)
tm:: VCorpus(VectorSource())
```

2-Après avoir faire le prétraitement il est intéressant de d'afficher la fréquence des mots ou des ngrammes.On utilise le package ggplot2. Ainsi on peut faire la matrice de term .

Il s'agit de transformer le texte en une matrice document-terme, où chaque ligne correspond à un document et chaque colonne correspond à un mot.

tm::DocumentTermMatrix.

L'étape suivante consiste à faire le nuage de mots.Pour cela nous verrons le package necessaire.

```
library(wordcloud)
```

Après avoir réalisé le nuage des mots, il est nécessaire de faire le réseau des mots.

```
library(ggraph)
```

Cas pratique

```r
library(readxl)
txte <- readLines('FormulaMilk_3months.txt' , encoding = "UTF-8")
txte <- txte[1:3]
```

Prétraitement

```r
txte <- removePunctuation(txte)
txte[1:2]

## [1] "text"
## [2] "I have 67 days worth of milk stocked up for my little baby  I can do
this I can beat this formula shortage Its soooo hard but I need to keep going
for now As much as its mentally challenging for me to keep going I would
rather do this than to hunt down formula"
```

Supprimer les nombres

```r
txte <- removeNumbers(txte)
txte[1:3]

## [1] "text"
## [2] "I have  days worth of milk stocked up for my little baby  I can do
this I can beat this formula shortage Its soooo hard but I need to keep going
for now As much as its mentally challenging for me to keep going I would
rather do this than to hunt down formula"
## [3] "Free Day amp Night Toddler Milk Samples  HAPPi are giving away free
samples of their Day amp Night toddler milk If you claim the free sample you
will get three samples of the Day Toddler Milk Drink Formula and three
samples of the Night httpstcowkZsAlf"
```

Convertir le texte en minuscule

```r
for(i in 1:length(txte))
   txte[i]=tolower(txte[i])
txte[1:3]

## [1] "text"
## [2] "i have  days worth of milk stocked up for my little baby  i can do
this i can beat this formula shortage its soooo hard but i need to keep going
for now as much as its mentally challenging for me to keep going i would
rather do this than to hunt down formula"
## [3] "free day amp night toddler milk samples  happi are giving away free
samples of their day amp night toddler milk if you claim the free sample you
will get three samples of the day toddler milk drink formula and three
samples of the night httpstcowkzsalf"
```

Supprimer votre propre liste de mots non désirés

```
txte <- removeWords(txte, c("for",
"his","a","of","is","hes","IN","I","but","i"))
txte[1:3]

## [1] "text"
## [2] " have  days worth  milk stocked up  my little baby   can do this  can
beat this formula shortage its soooo hard   need to keep going  now as much
as its mentally challenging  me to keep going  would rather do this than to
hunt down formula"
## [3] "free day amp night toddler milk samples  happi are giving away free
samples  their day amp night toddler milk if you claim the free sample you
will get three samples  the day toddler milk drink formula and three samples
the night httpstcowkzsalf"
```

transformation dans un format Corpus

```
txte <- Corpus(VectorSource(txte))
txte

## <<SimpleCorpus>>
## Metadata:  corpus specific: 1, document level (indexed): 0
## Content:  documents: 3
```

Transformation du Corpus en une matrice (longueur > 3)

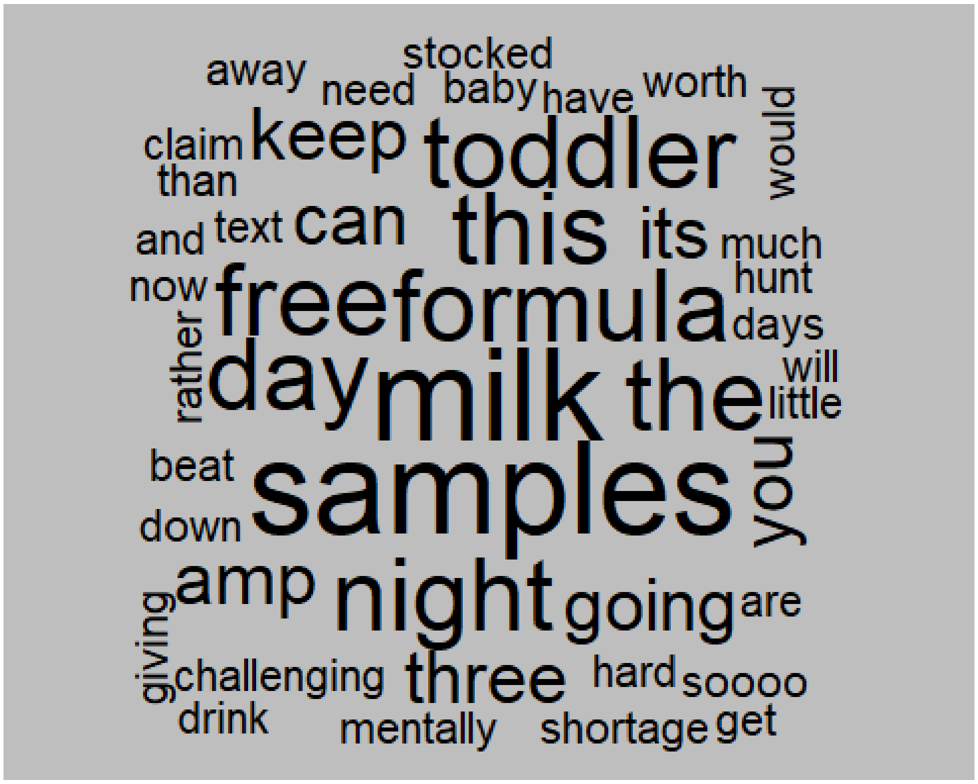
```
tdm <- TermDocumentMatrix(txte,control = list(minWordLength=3))
```

mot le plus fréquents dans le texte

```
m <- as.matrix(tdm)
freqWords=rowSums(m)
freqWords=sort(freqWords , d=T)
t(freqWords[1:6])

##      milk samples formula this day free
## [1,]    4       4       3    3   3    3
```

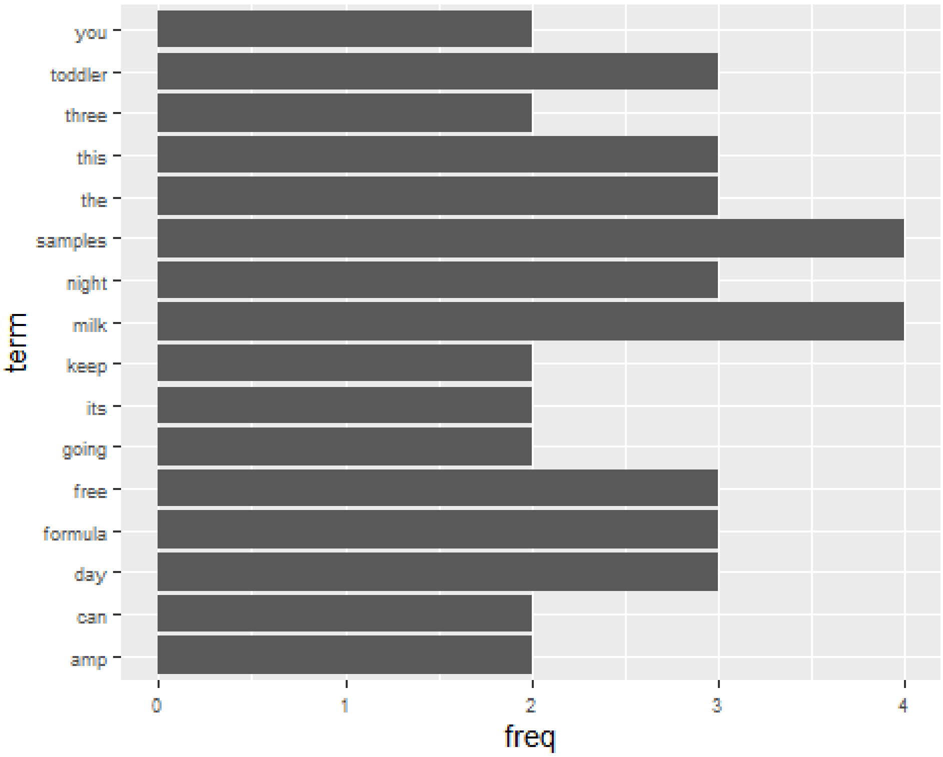
Traçons maintenant le word cloud.

```
freqWords=rowSums(m)
v=sort(freqWords,d=T)
dt=data.frame(word=names(v),freq=v)
head(dt)

##             word freq
## milk        milk    4
## samples samples    4
## formula formula    3
## this         this     3
## day           day     3
## free         free     3

par(bg="gray")
wordcloud(dt$word,dt$freq,min.freq =5 ,stack=T,random.order = F)
```



# Trouver les termes les plus fréquents

```
freq.terms <- findFreqTerms(tdm, lowfreq = 20)
freq.terms

term.freq <- rowSums(as.matrix(tdm))
term.freq <- subset(term.freq, term.freq >= 2)
df <- data.frame(term = names(term.freq), freq = term.freq)

library(ggplot2)

ggplot(df, aes(x=term, y=freq)) + geom_bar(stat="identity") +
    coord_flip() + theme(axis.text=element_text(size=7))
```



Recherche d’associations

```
findAssocs(tdm, "milk", 0.2)

## $milk
##             amp             and             are            away
claim
##            0.94            0.94            0.94            0.94
0.94
##             day           drink            free             get
giving
##            0.94            0.94            0.94            0.94
0.94
##           happi httpstcowkzsalf           night          sample
samples
##            0.94            0.94            0.94            0.94
0.94
##             the           their           three         toddler
will
##            0.94            0.94            0.94            0.94
0.94
##             you         formula
##            0.94            0.33
```

#Cas pratique 2

```
df<-read_xlsx("lait.xlsx")
base <- df %>% dplyr:: distinct(text, .keep_all = TRUE)
base <- df[1:100,] %>% dplyr::select(text)
```

Numerotation des observations

```
base1<-tibble(index_person=1:100, opinion=base$text)
head(base1$opinion,1)

## [1] "I have 67 days worth of milk stocked up for my little baby
<U+0001F62D> I can do this. I can beat this formula shortage. It's soooo hard
but I need to keep going for now. As much as it's mentally challenging for me
to keep going I would rather do this than to hunt down formula."
```

creation des tokens des mots indexés

```
base_mot<-base1 %>%
  tidytext::unnest_tokens(word, opinion)
```

Suppression des mots vides

```
base_mot<-base_mot%>%anti_join(stop_words)

## Joining with `by = join_by(word)`
```

creations du dictionnaire des mots vides

```
dictionnaire<-data.frame(word
=c("t.co","to","i","are","https","the","their","his","to","for","i","I","l","
we","my","of","from","a","at","he","that","so","if","our"))

texte_filtre<-anti_join(base_mot,dictionnaire,by="word")
ivi<-base_mot %>%
  dplyr::filter(!word %in%dictionnaire )
```

Supprimer les chiffres

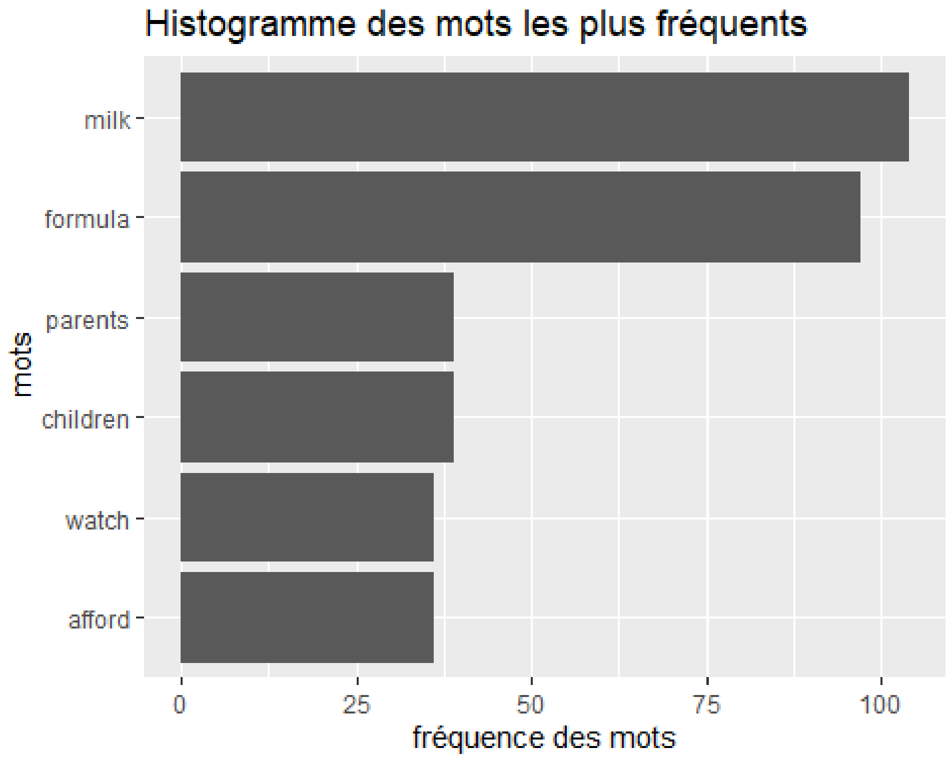
```
texte_filtre<-texte_filtre %>% filter(!grepl("\\d",word))
```

fréquence des mots

```
frequence_mot<-base_mot%>%dplyr::count(word,sort=TRUE)%>%
  filter(!word %in% dictionnaire$word) %>%
  filter(n > 20)
```

histogramme des mots

```
base_mot %>%
  dplyr::count(word, sort = TRUE) %>%filter(!word %in% dictionnaire$word) %>%
  filter(n > 35) %>%
  mutate(word = reorder(word, n)) %>%
  ggplot(aes(n, word)) +
  geom_col() +
  ylab(" mots")+
  xlab("fréquence des mots")+
  ggtitle("Histogramme des mots les plus fréquents")
```



# Mots frequents

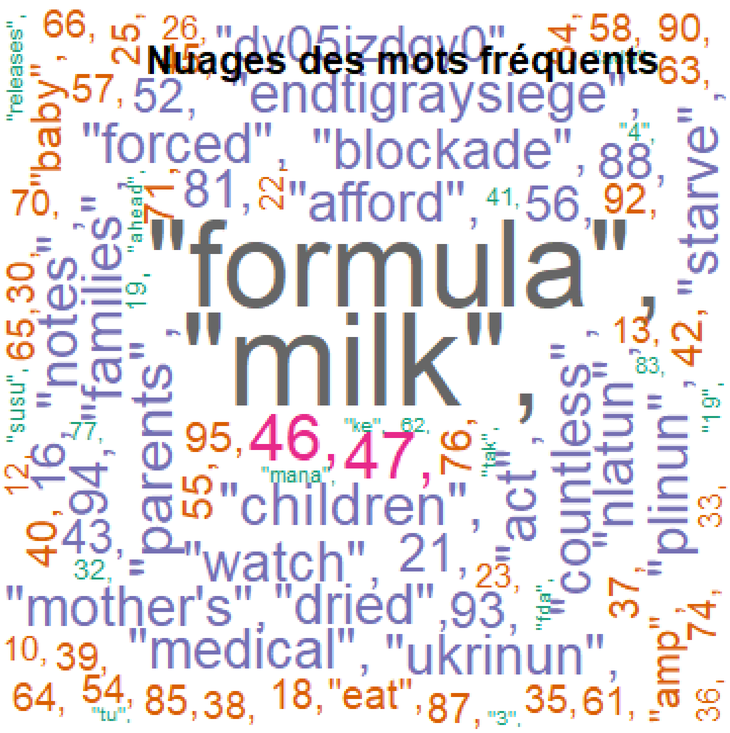
```
mot_inter<-frequence_mot[1:7,] %>% dplyr::select(word)
```

nuages de mots avec tm

```
dtm <-
TermDocumentMatrix(base_mot%>%dplyr::mutate(word=stringr::str_replace_all(.$w
ord, "â⍰⍰", " ")) %>% filter(!word %in% dictionnaire$word))
m <- as.matrix(dtm)
v <- sort(rowSums(m),decreasing=TRUE)
d <- data.frame(word = names(v),freq=v)
head(d, 10)
```

```
##                     word freq
## "milk",          "milk",  104
## "formula",    "formula",   97
## 46,                  46,   45
## 47,                  47,   45
## "children", "children",   39
## "parents",    "parents",   39
## "afford",      "afford",   36
## "watch",        "watch",   36
## "act",            "act",   35
## "blockade", "blockade",   35
```

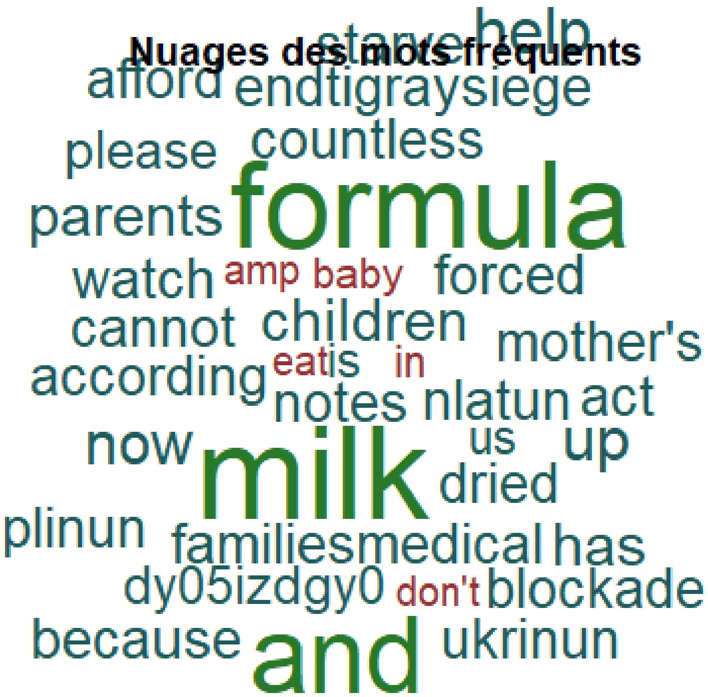
```
set.seed(1234)
wordcloud(words = d$word, freq = d$freq, min.freq = 10000,
          max.words=200, random.order=FALSE, rot.per=0.35,
          colors=brewer.pal(8, "Dark2"))
title(main = "Nuages des mots fréquents")
```



# nuages des mots

```
base2 <- base1 %>%
  dplyr::mutate(text = stringr::str_replace_all(.$opinion, "â⍰⍰", " "))  %>%
  tidytext::unnest_tokens(word, text) %>%
  filter(!word %in% dictionnaire$word) %>%
  dplyr::count(word, sort = TRUE)%>%
  filter(n > 20)
```

```
wordcloud(base2$word, base2$n, max.words = 200, rot.per = FALSE, colors =
c("#973232", "#1E5B5B", "#6D8D2F", "#287928"))
title(main = "Nuages des mots fréquents")
```



# Gestion des bigrams

```
bigram <- base1 %>%
  unnest_tokens(bigram, opinion, token = "ngrams", n = 2) %>%
  filter(!is.na(bigram))
bigrams_separated <-bigram %>%
  tidyr::separate(bigram, c("word1", "word2"), sep = " ")
```

Filtrage du bigramme

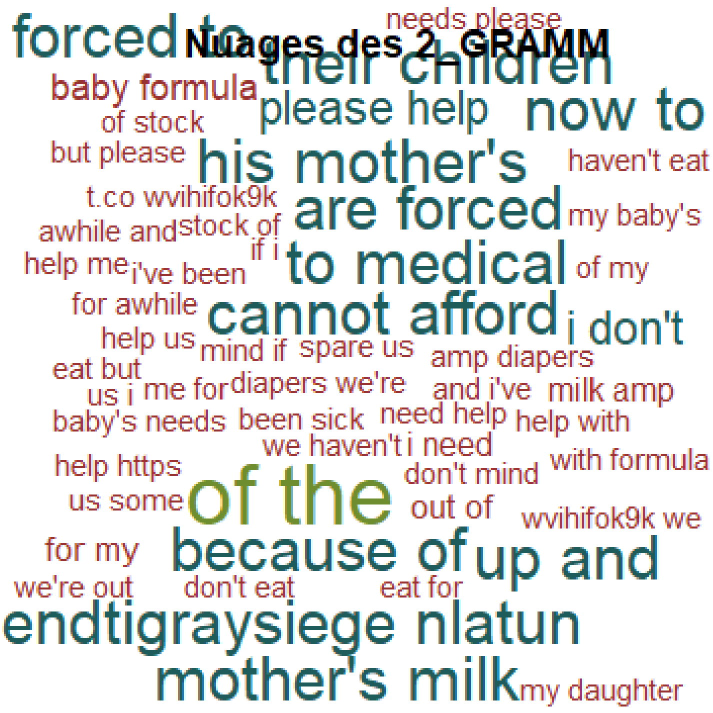
```
bigrams_filtered <- bigrams_separated %>%
  filter(!word1 %in% dictionnaire$word) %>%
  filter(!word2 %in% dictionnaire$word)
```

compter les bigrammes

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

Nuage des bigrams

```
graphe_bigram <- base1 %>%
  dplyr::mutate(text = stringr::str_replace_all(.$opinion, "â⍰⍰", " "))  %>%
  tidytext::unnest_tokens(bigram, text, token = "ngrams", n = 2) %>%
  filter(!is.na(bigram)) %>%
  dplyr::count(bigram, sort = TRUE)%>%
  filter(n > 10)
wordcloud(graphe_bigram$bigram, graphe_bigram$n, max.bigram = 200, rot.per =
FALSE, colors = c("#973232", "#1E5B5B", "#6D8D2F", "#287928"))
title(main = "Nuages des 2_GRAMM")
```



Cas pratique 3 : messages whatsapp

la base de l'ensae

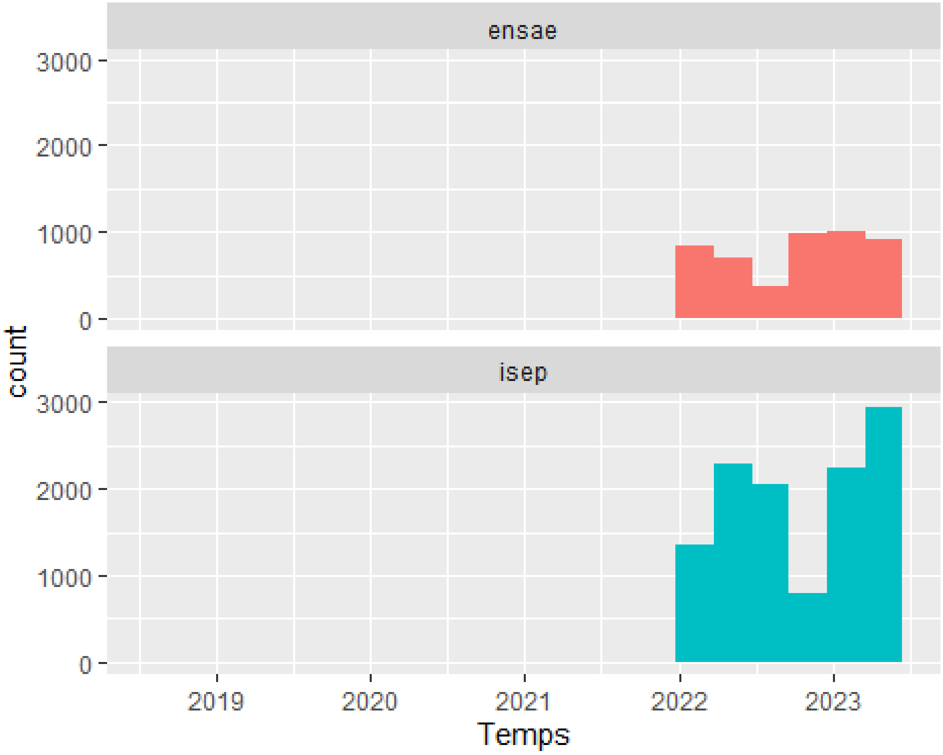
```
library(dplyr)
library(rwhatsapp)

ensae <- rwa_read("ensae.txt")%>% filter(!is.na(author))
colnames(ensae)

## [1] "time"       "author"     "text"       "source"     "emoji"
## [6] "emoji_name"
```

la base des isep

```
isep <- rwa_read("isep.txt")%>% filter(!is.na(author))
colnames(ensae)

## [1] "time"         "author"       "text"         "source"       "emoji"
## [6] "emoji_name"
```

chronologie des messages entre les deux groupes

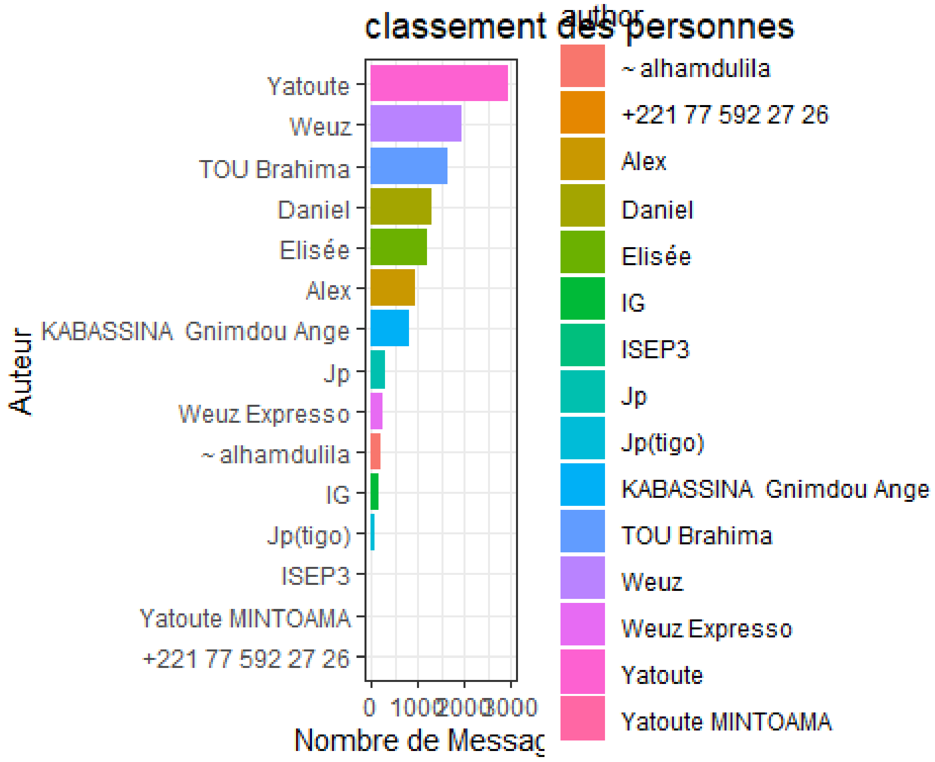
```
library(lubridate)

library(ggplot2)

ecole <- bind_rows(ensae %>%
                     mutate(source = "ensae"),
                   isep %>%
                     mutate(source = "isep")) %>%
  mutate(Temps = ymd_hms(time))

ggplot(ecole, aes(x = Temps, fill = source)) +
  geom_histogram(position = "identity", bins = 20, show.legend = FALSE) +
  facet_wrap(~source, ncol = 1)
```



```
# classement des personnes selon les messages
```

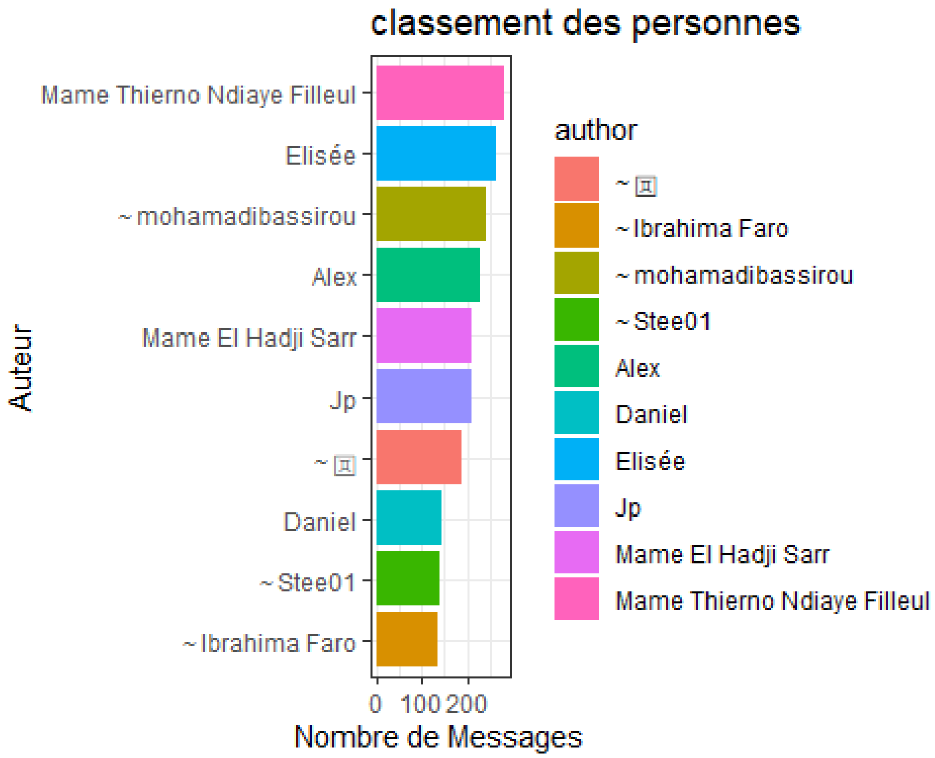
isep

```
isep %>%
  count(author) %>%
  ggplot(aes(x = reorder(author, n), y = n, fill=author)) +
  geom_bar(stat = "identity") + xlab("Auteur") + ylab("Nombre de Messages") +
  coord_flip() + theme_bw() +
  ggtitle("classement des personnes")
```



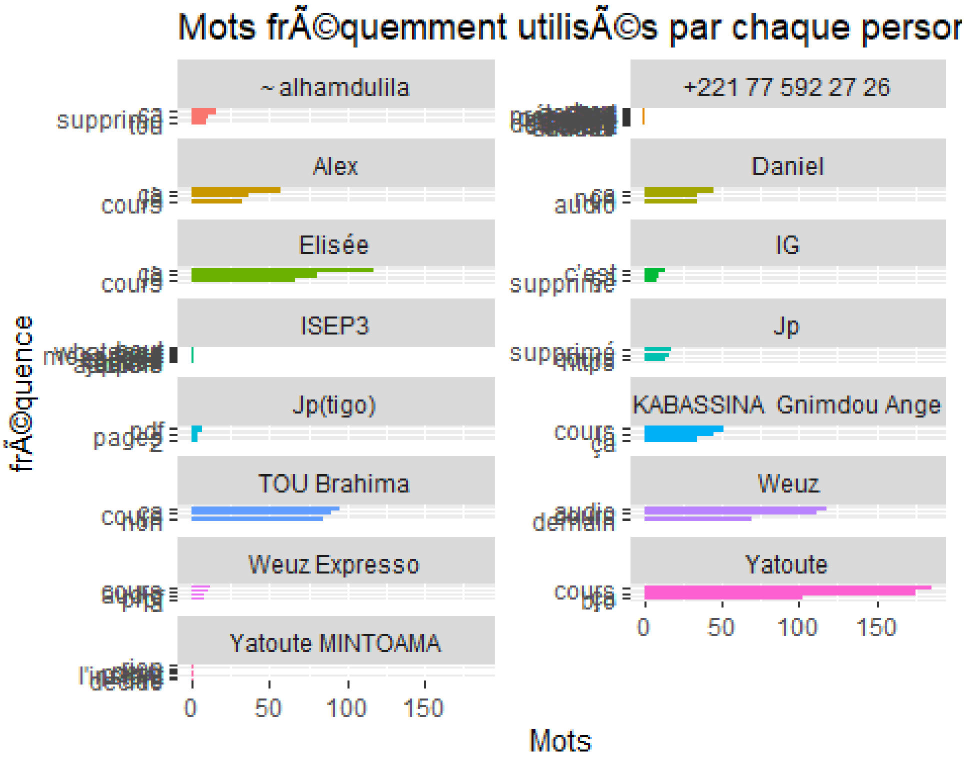
ensae

```
ensae %>%
  count(author) %>%
  top_n(10, wt = n) %>%
  ggplot(aes(x = reorder(author, n), y = n, fill = author)) +
  geom_bar(stat = "identity") +
  xlab("Auteur") + ylab("Nombre de Messages") + coord_flip() + theme_bw() +
  ggtitle("classement des personnes")
```



```
# les mots fréquements utilisés par ces personnes isep
```

```
library(stopwords)

library(tidytext)
vide <- c(stopwords(language = 'fr'),"sticker","omis","absente","image",
          "document","message","supprimÃ©","manquant","c'est","oui","a","ok",
          "pa","Ã§a","hein","va","lÃ ")
isepnew <- isep%>% unnest_tokens(input = text,output = word) %>%
  filter(!word %in% vide)
isepnew %>% count(author, word, sort = TRUE) %>%
  group_by(author) %>%
  top_n(n = 3, n) %>%
  ggplot(aes(x = reorder_within(word, n, author), y = n, fill = author)) +
  geom_col(show.legend = FALSE) +
  ylab("Mots") +
  xlab("frÃ©quence") +
  coord_flip() +
  facet_wrap(~author, ncol = 2, scales = "free_y") +
  scale_x_reordered() +
  ggtitle("Mots frÃ©quemment utilisÃ©s par chaque personne")
```



# Etudions maintenant la base école .

Fréquences des mots.

Nettoyons d'abord la base école

```
ecolenew <- ecole%>% unnest_tokens(input = text,output = word) %>%
  filter(!word %in% vide)
```

fréquences de mots pour selon la source

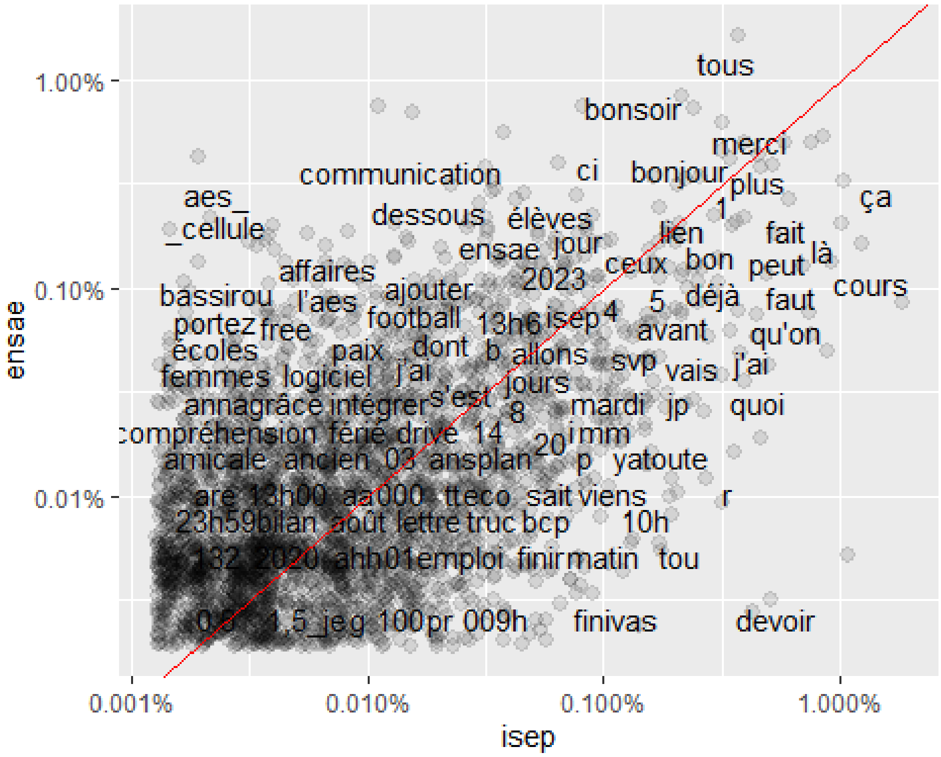
```
fréquence <- ecolenew %>%
  count(source, word, sort = TRUE) %>%
  left_join(ecolenew %>%
              count(source, name = "total")) %>%
  mutate(freq = n/total)

fréquence

## # A tibble: 15,822 x 5
##    source word        n total    freq
##    <chr>  <chr>   <int> <int>   <dbl>
##  1 isep   ça        618 43685 0.0141
##  2 isep   cours     595 43685 0.0136
##  3 ensae  tous      466 29620 0.0157
##  4 isep   non       379 43685 0.00868
##  5 isep   là        368 43685 0.00842
##  6 isep   faire     306 43685 0.00700
##  7 isep   si        292 43685 0.00668
##  8 ensae  bonsoir   290 29620 0.00979
##  9 isep   demain    274 43685 0.00627
## 10 isep   audio     271 43685 0.00620
## # ... with 15,812 more rows
```

cadre de données de forme différente

```
library(tidyr)

fréquence <- fréquence %>%
  select(source, word, freq) %>%
  pivot_wider(names_from = source, values_from = freq) %>%
  arrange(isep, ensae)

fréquence

## # A tibble: 12,717 x 3
##    word        isep     ensae
##    <chr>      <dbl>     <dbl>
##  1 0.6    0.0000229 0.0000338
##  2 1.5    0.0000229 0.0000338
##  3 10000f 0.0000229 0.0000338
##  4 10min  0.0000229 0.0000338
##  5 11e    0.0000229 0.0000338
##  6 12.0   0.0000229 0.0000338
##  7 12h20  0.0000229 0.0000338
##  8 13.0   0.0000229 0.0000338
##  9 14.0   0.0000229 0.0000338
## 10 15.0   0.0000229 0.0000338
## # ... with 12,707 more rows
```

compararaison des fréquences des mots entre les deux groupes

```
library(scales)

ggplot(fréquence, aes(isep, ensae)) +
  geom_jitter(alpha = 0.1, size = 2.5, width = 0.25, height = 0.25) +
  geom_text(aes(label = word), check_overlap = TRUE, vjust = 1.5) +
  scale_x_log10(labels = percent_format()) +
  scale_y_log10(labels = percent_format()) +
  geom_abline(color = "red")
```



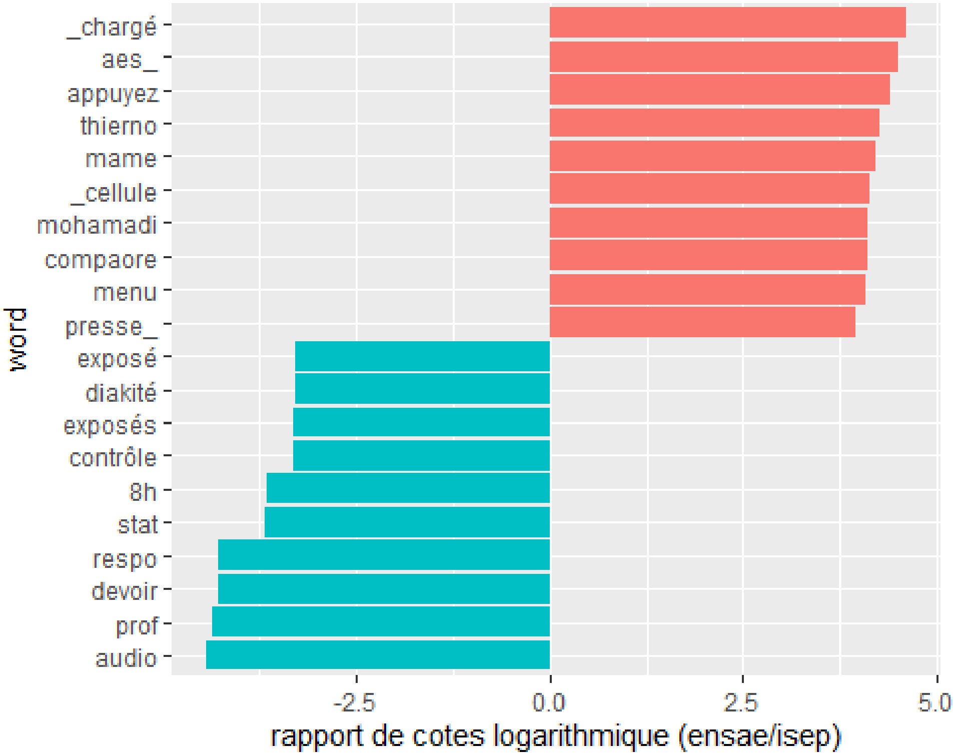
# compararaison de l’utilisation des mots

```
word_ratios <- ecolenew %>%
  count(word, source) %>%
  group_by(word) %>%
  filter(sum(n) >= 10) %>%
  ungroup() %>%
  pivot_wider(names_from = source, values_from = n, values_fill = 0) %>%
  mutate_if(is.numeric, list(~(. + 1) / (sum(.) + 1))) %>%
  mutate(logratio = log(ensae /isep)) %>%
  arrange(desc(logratio))

word_ratios %>% arrange(abs(logratio))
```

```
## # A tibble: 1,306 x 4
##    word        ensae     isep   logratio
##    <chr>       <dbl>    <dbl>      <dbl>
##  1 chacun   0.000993 0.000993  0.0000303
##  2 10       0.00141  0.00141   0.00123
##  3 avis     0.000575 0.000576 -0.00290
##  4 fois     0.00172  0.00173  -0.00290
##  5 ami      0.000418 0.000416  0.00407
##  6 niveau   0.000836 0.000833  0.00407
##  7 beaucoup 0.00193  0.00192   0.00615
##  8 semaine  0.00355  0.00349   0.0177
##  9 9        0.000261 0.000256  0.0196
## 10 comprend 0.000261 0.000256  0.0196
## # ... with 1,296 more rows
```

graphique

```
word_ratios %>%
  group_by(logratio < 0) %>%
  slice_max(abs(logratio), n = 10) %>%
  ungroup() %>%
  mutate(word = reorder(word, logratio)) %>%
  ggplot(aes(word, logratio, fill = logratio < 0)) +
  geom_col(show.legend = FALSE) +
  coord_flip() +
  ylab("rapport de cotes logarithmique (ensae/isep)") +
  scale_fill_discrete(name = "", labels = c("ensae", "isep"))
```



#

```
words_time <- ecolenew %>%
  mutate(time_floor = floor_date(time, unit = "1 month")) %>%
  count(time_floor, source, word) %>%
  group_by(source, time_floor) %>%
  mutate(time_total = sum(n)) %>%
  group_by(source, word) %>%
  mutate(word_total = sum(n)) %>%
  ungroup() %>%
  rename(count = n) %>%
  filter(word_total > 30)
```

```
words_time

## # A tibble: 5,233 x 6
##    time_floor          source word    count time_total word_total
##    <dttm>              <chr>  <chr>   <int>      <int>      <int>
##  1 2018-10-01 00:00:00 ensae  groupe      2         17         50
##  2 2018-10-01 00:00:00 ensae  peut        1         17         51
##  3 2021-03-01 00:00:00 isep   groupe      1         15        112
##  4 2021-03-01 00:00:00 isep   lire        1         15         32
##  5 2021-03-01 00:00:00 isep   peut        1         15        235
##  6 2021-12-01 00:00:00 ensae  1           1        181         95
##  7 2021-12-01 00:00:00 ensae  2           1        181         84
##  8 2021-12-01 00:00:00 ensae  2022        1        181         55
##  9 2021-12-01 00:00:00 ensae  _chargé     1        181         60
## 10 2021-12-01 00:00:00 ensae  adiouma     2        181         38
## # ... with 5,223 more rows
```

```
nested_data <- words_time %>%
  nest(data = c(-word, -source))

nested_data

## # A tibble: 373 x 3
##    source word    data
##    <chr>  <chr>   <list>
##  1 ensae  groupe  <tibble [16 x 4]>
##  2 ensae  peut    <tibble [19 x 4]>
##  3 isep   groupe  <tibble [15 x 4]>
##  4 isep   lire    <tibble [14 x 4]>
##  5 isep   peut    <tibble [19 x 4]>
##  6 ensae  1       <tibble [18 x 4]>
##  7 ensae  2       <tibble [18 x 4]>
##  8 ensae  2022    <tibble [12 x 4]>
##  9 ensae  _chargé <tibble [17 x 4]>
## 10 ensae  adiouma <tibble [10 x 4]>
## # ... with 363 more rows
```

```
library(purrr)

nested_models <- nested_data %>%
  mutate(models = map(data, ~ glm(cbind(count, time_total) ~ time_floor, .,
                                  family = "binomial")))
nested_models
```

```
## # A tibble: 373 x 4
##    source word    data              models
##    <chr>  <chr>   <list>            <list>
##  1 ensae  groupe  <tibble [16 x 4]> <glm>
##  2 ensae  peut    <tibble [19 x 4]> <glm>
##  3 isep   groupe  <tibble [15 x 4]> <glm>
##  4 isep   lire    <tibble [14 x 4]> <glm>
##  5 isep   peut    <tibble [19 x 4]> <glm>
##  6 ensae  1       <tibble [18 x 4]> <glm>
##  7 ensae  2       <tibble [18 x 4]> <glm>
##  8 ensae  2022    <tibble [12 x 4]> <glm>
##  9 ensae  _chargé <tibble [17 x 4]> <glm>
## 10 ensae  adiouma <tibble [10 x 4]> <glm>
## # ... with 363 more rows
```

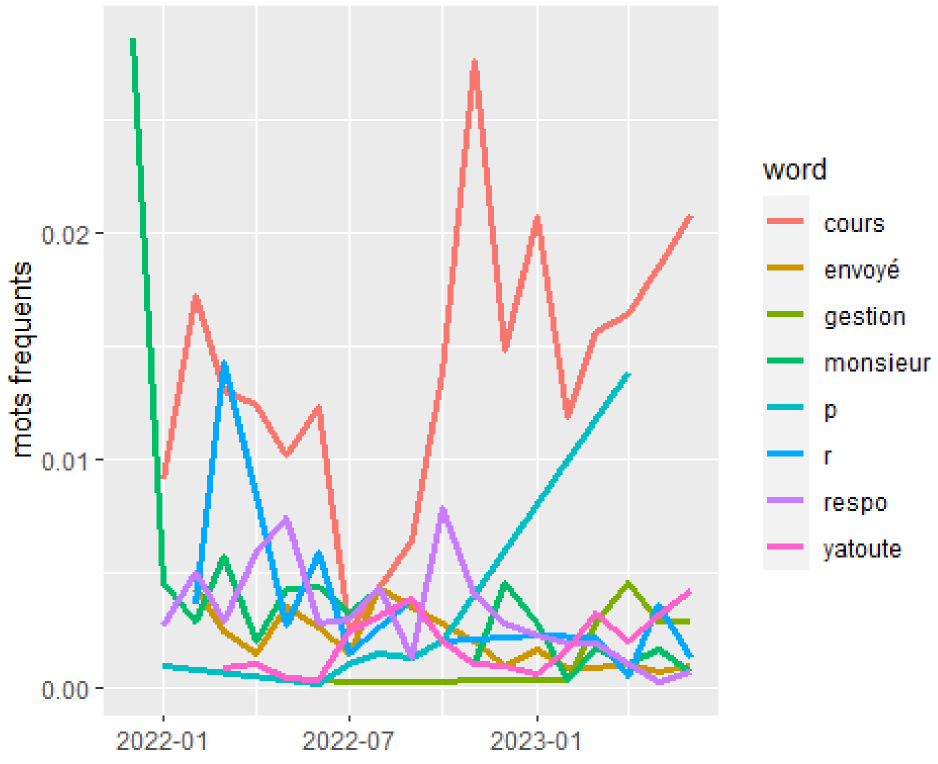
```
library(broom)

slopes <- nested_models %>%
  mutate(models = map(models, tidy)) %>%
  unnest(cols = c(models)) %>%
  filter(term == "time_floor") %>%
  mutate(adjusted.p.value = p.adjust(p.value))

top_slopes <- slopes %>%filter(adjusted.p.value < 0.05)

top_slopes
```

```
## # A tibble: 14 x 9
##    source word     data     term      estimate std.e~1 stati~2  p.value
adjust~3
##    <chr>  <chr>    <list>   <chr>        <dbl>   <dbl>   <dbl>    <dbl>
<dbl>
##  1 ensae  _chargé  <tibble> time_flo~ -3.63e-8 9.35e-9   -3.89 1.02e- 4
3.66e- 2
##  2 ensae  club     <tibble> time_flo~ -3.09e-8 6.23e-9   -4.96 7.16e- 7
2.64e- 4
##  3 ensae  jean     <tibble> time_flo~ -5.45e-8 1.22e-8   -4.45 8.41e- 6
3.07e- 3
##  4 ensae  ndiaye   <tibble> time_flo~ -3.42e-8 7.65e-9   -4.48 7.59e- 6
2.78e- 3
##  5 ensae  pierre   <tibble> time_flo~ -6.01e-8 1.36e-8   -4.41 1.05e- 5
3.82e- 3
##  6 isep   monsieur <tibble> time_flo~ -3.58e-8 7.09e-9   -5.05 4.32e- 7
1.59e- 4
##  7 isep   cours    <tibble> time_flo~  1.52e-8 2.97e-9    5.12 2.99e- 7
1.11e- 4
##  8 isep   p        <tibble> time_flo~  1.23e-7 1.62e-8    7.57 3.65e-14
1.36e-11
```

```
##   9 isep   respo    <tibble> time_flo~ -3.71e-8 7.34e-9   -5.06 4.27e- 7
1.58e- 4
## 10 isep   envoyé   <tibble> time_flo~ -3.68e-8 9.29e-9   -3.96 7.60e- 5
2.74e- 2
## 11 isep   r        <tibble> time_flo~ -3.78e-8 6.58e-9   -5.74 9.60e- 9
3.57e- 6
## 12 isep   yatoute  <tibble> time_flo~  4.07e-8 9.74e-9    4.18 2.91e- 5
1.05e- 2
## 13 isep   gestion  <tibble> time_flo~  8.31e-8 1.98e-8    4.19 2.75e- 5
9.99e- 3
## 14 ensae  anciens  <tibble> time_flo~ -2.15e-7 4.51e-8   -4.77 1.81e- 6
6.63e- 4
## # ... with abbreviated variable names 1: std.error, 2: statistic,
## #   3: adjusted.p.value
```

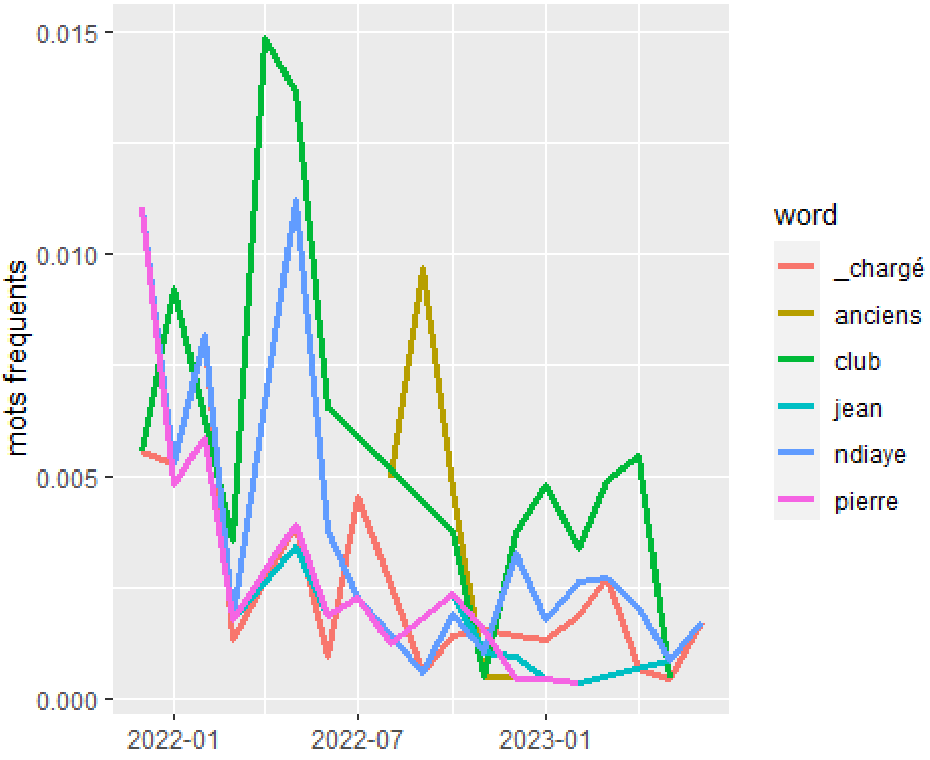
isep

```
words_time %>%
  inner_join(top_slopes, by = c("word", "source")) %>%
  filter(source == "isep") %>%
  ggplot(aes(time_floor, count/time_total, color = word)) +
  geom_line(size = 1.3) +
  labs(x = NULL, y = "mots frequents")
```



```
# ensae
```

```
words_time %>%
  inner_join(top_slopes, by = c("word", "source")) %>%
  filter(source == "ensae") %>%
  ggplot(aes(time_floor, count/time_total, color = word)) +
  geom_line(size = 1.3) +
  labs(x = NULL, y = "mots frequents")
```


mots frequents
word
_chargé
anciens
club
jean
ndiaye
pierre