Data and Climate

Session 5 - Extracting and analyzing textual data using R

Jean-Baptiste Guiffard

2024-01-11

Course structure

Sessions

Sessions	Topics
Session 1	The Basics of R / Manipulating dataset with the DPLYR package
Session 2	Graphic representations with GGPLOT
Session 3	Making maps with R
Session 4	Web scraping with R
Session 5	Extracting and analyzing textual data using R
Session 6	Produce documents with Rmarkdown. . .

La manipulation des données textuelles sur R

La production et l'analyse de données qualitatives

- De plus en plus important de maîtriser des méthodes d'analyse de données qualitatives (et particulièrement textuelles).
- Les packages et fonctions Rfacilitent le processus d'identification, de manipulation et d'analyse des textes.

Que pouvons-nous faire ?

- Analyse de la fréquence des mots
- Comparaison des textes
- Sentiment Analysis
- Des nuages de mots
- Des réseaux de co-occurence
- Analyse des thématiques et leur évolution en fonction du temps

Quelques références

- https://m-clark.github.io/text-analysis-with-R/string-theory.html#basic-text-functionality
- https://www.red-gate.com/simple-talk/databases/sql-server/bi-sql-server/text-mining-and-sentiment-analysis-with-r/
- https://www.tidytextmining.com/topicmodeling.html

Le traitement des chaînes de caractères sur R

Partons d'un exemple simple...

```
"L'influence humaine a réchauffé l'atmosphère, l'océan et les terres."
```

```
## [1] "L'influence humaine a réchauffé l'atmosphère, l'océan et les te
```

Nous pouvons en faire un objet sur R...

```
ma_phrase <- "L'influence humaine a réchauffé l'atmosphère, l'océan et
```

Nous pouvons avoir un vecteur de *characters* (des chaînes de caracters séparées par des virgules) qui peut aussi être utilisé dans le cadre d'une variable dans une *data.frame*.

Quelques fonctions basiques de manipulation des données "characters" sur R

Vérifier que le scalaire, le vecteur ou la variable étudié est constitué d'une ou de chaînes de caractères.

```
is.character(ma_phrase)
```

```
## [1] TRUE
```

```
nchar(ma_phrase) #nombre de caractères dans le string
```

```
## [1] 68
```

Le package **stringr** (qui se charge aussi avec le package **tidyverse**) fournit un ensemble cohérent de fonctions conçues pour rendre le travail avec les chaînes de caractères aussi facile que possible.

Quelques fonctions basiques (II)

Détecter un champ dans une chaîne de caractères...

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

```
## Warning: le package 'stringr' a été compilé avec la version R 4.1.3
```

```
str_detect(ma_phrase,"influence")
```

```
## [1] TRUE
```

```
str_detect(ma_phrase,'calamar')
```

```
## [1] FALSE
```

Remplacer une partie d'une chaîne de caractères...

```
ma_phrase <- str_replace(ma_phrase,'humaine', "de l'homme")
```

Quelques fonctions basiques (III)

Remplacer plusieurs éléments d'un seul coup (gsub en séparant les éléments avec |)... Nous allons l'utiliser régulièrement pour le nettoyage des variables "textuelles".

```
ma_phrase <- gsub("'|,", " ", ma_phrase)
print(ma_phrase)
```

```
## [1] "L influence de l homme a réchauffé l atmosphère  l'océan et les
```

Nos premières analyses de corpus

Le package tidytext

```
# install.packages('tidytext')
library(tidytext)
```

- Un package efficace d'analyse textuelle (qui associe des fonctionnalités déjà présentes dans plusieurs autres packages : dplyr, broom, tidyr and ggplot2).
- Le format tidy text est un tableau avec un token par ligne.
- Un token est un mot, ou un phrase ou un n-gram.

Le vocabulaire

- Un **string** (ou un texte ou un document) : peut être stocké sous forme de chaîne de caractères, ou bien sous forme d'un vecteur de chaînes de caractères.
- Un **corpus** (ou une collection) : C'est un objet qui contient des strings (ou textes) avec des détails et métadonnées supplémentaires.
- La **matrice Document-term** : C'est une matrice décrivant un corpus de documents avec une ligne pour chaque document et une colonne pour chaque terme. Les valeurs à l'intérieur de la matrice sont soit un comptage de mots ou bien tf-idf.

Créer le corpus

```
#install.packages("tidyft") #Pour le recodage des variables textuelles
library(tidyft)
bdd_speech <- read.csv2('seance_5/bdd_discours_2007_2022.csv') %>%
  as.data.table() %>%
  utf8_encoding(titles) %>%
  utf8_encoding(dates) %>%
  as.data.frame()
head(bdd_speech$titles, n=6)
```

```
## [1] "Conseil des ministres du 4 janvier 2023. Mesures d'ordre indivi
## [2] "Conseil des ministres du 4 janvier 2023. Dispositions relatives
## [3] "Conseil des ministres du 4 janvier 2023. Droits sociaux des per
## [4] "Communiqué de la Présidence de la République, en date du 4 janv
## [5] "Déclaration de M. Sébastien Lecornu, ministre des armées, sur l
## [6] "Interview de M. Bruno Le Maire, ministre de l'économie, des fin
```

Nettoyage des données

Nous pouvons utiliser la fonction gsub pour nettoyer des variables "textuelles":

```
bdd_speech$titles <- gsub("[[:punct:]]", " ", bdd_speech$titles)
bdd_speech$titles <- gsub('[[:digit:]]+', '', bdd_speech$titles)
bdd_speech$titles <- gsub("[\r\n]", "", bdd_speech$titles)
head(bdd_speech$titles, n=4)
```

```
## [1] "Conseil des ministres du  janvier   Mesures d ordre individuel
## [2] "Conseil des ministres du  janvier   Dispositions relatives au c
## [3] "Conseil des ministres du  janvier   Droits sociaux des personne
## [4] "Communiqué de la Présidence de la République  en date du  janvi
```

Tokenisation

La fonction unnest() extrait les tokens, ou mots particuliers d'un ensemble de données, de la colonne "texte" et les distribue dans des lignes individuelles avec les métadonnées correspondantes.

```
tidy_text <- tibble(bdd_speech) %>%
  unnest_tokens("word", titles)

head(tidy_text, n=4)
```

```
## # A tibble: 4 x 3
##   dates          links
##   <chr>          <chr>
## 1 4 janvier 2023 /discours/287710-conseil-des-ministres-04012023-mes
## 2 4 janvier 2023 /discours/287710-conseil-des-ministres-04012023-mes
## 3 4 janvier 2023 /discours/287710-conseil-des-ministres-04012023-mes
## 4 4 janvier 2023 /discours/287710-conseil-des-ministres-04012023-mes
```

Retirer les stopwords

Les **stopwords** → sont un ensemble de mots couramment utilisés dans une langue. Lorsqu'on traite le langage naturel, nous voulons filtrer ces mots de nos données.

- Chargement des stopwords français
- On peut utiliser anti_join() pour trouver les stop words qui apparaissent parmi nos tokens et les supprimer.

```
#install.packages("lsa") # packages pour le chargement de stopwords (po
library(lsa)
data(stopwords_fr)
df_stopwords_fr <- data.frame(word=stopwords_fr,
                              lexicon = "?")
tidy_text <- tidy_text %>% dplyr::anti_join(df_stopwords_fr)
head(tidy_text, n=4)
```

```
## # A tibble: 4 x 3
##   dates          links
##   <chr>          <chr>
## 1 4 janvier 2023 /discours/287710-conseil-des-ministres-04012023-mes
```

Préparer l'analyse de l'occurence des mots

Avec cette nouvelle base de données, nous pouvons construire une matrice qui contient d'une part, les mots qui apparaissent dans le corpus et d'autre part, leur fréquence d'apparition.

```
head(tidy_text %>% dplyr::count(word, sort = TRUE))
```

```
## # A tibble: 6 x 2
##   word            n
##   <chr>       <int>
## 1 ministre    26881
## 2 déclaration 23294
## 3 paris       11685
## 4 mme         11671
## 5 interview    9845
## 6 secrétaire   8516
```

Compléter la liste des stopwords (I)

Des mots spécifiques à notre corpus peuvent revenir de manière répétées sans être forcément très informatif.

```
new_stop_words_fr <- data.frame("word" = c("mme", "janvier", "février",
tidy_text <- tidy_text %>% dplyr::anti_join(new_stop_words_fr)
```

```
## Joining, by = "word"
```

```
head(tidy_text %>% dplyr::count(word, sort = FALSE))
```

```
## # A tibble: 6 x 2
##   word            n
##   <chr>       <int>
## 1 aaa             1
## 2 aah             5
## 3 aai             1
## 4 abad            1
## 5 abadi           1
## 6 abaissement     8
```

Compléter la liste des stopwords (II)

```
head(tidy_text %>% dplyr::count(word, sort = TRUE))
```

```
## # A tibble: 6 x 2
##   word            n
##   <chr>       <int>
## 1 ministre    26881
## 2 déclaration 23294
## 3 paris       11685
## 4 interview    9845
## 5 secrétaire   8516
## 6 affaires     8427
```

Retrouver la racine de chaque mot

Les tokens doivent être "stemmed" → réduction des mots à leur racine, leur base ou leur forme.

```
tidy_text_stems <- tidy_text %>%
  mutate_at("word", funs(wordStem((.), language="fr")))

head(tidy_text_stems)
```

```
## # A tibble: 6 x 3
##   dates          links
##   <chr>          <chr>
## 1 4 janvier 2023 /discours/287710-conseil-des-ministres-04012023-mes
## 2 4 janvier 2023 /discours/287710-conseil-des-ministres-04012023-mes
## 3 4 janvier 2023 /discours/287710-conseil-des-ministres-04012023-mes
## 4 4 janvier 2023 /discours/287710-conseil-des-ministres-04012023-mes
## 5 4 janvier 2023 /discours/287710-conseil-des-ministres-04012023-mes
## 6 4 janvier 2023 /discours/287708-conseil-des-ministres-04012023-dis
```

Notre premier nuage de mots

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

tidy_text %>%
  dplyr::count(word) %>%
  with(wordcloud(word, n, min.freq = 1000, colors = brewer.pal(8, "Dark
```

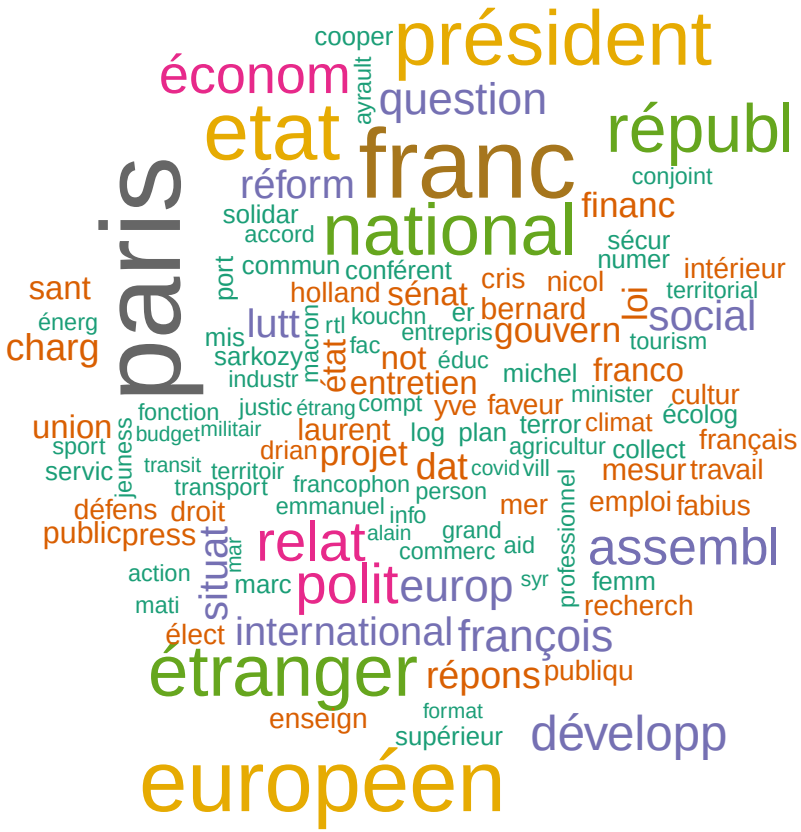
Améliorer les enseignements d'un nuage de mots

```
new_stop_words_fr2 <- data.frame("word" = c("affaires","ministre", "min

tidy_text <- tidy_text %>% dplyr::anti_join(new_stop_words_fr2)
```

```
## Joining, by = "word"
```

```
tidy_text_stems <- tidy_text %>%
  mutate_at("word", funs(wordStem((.), language="fr")))
```


cooper
président
économ
ayrault
question
etat
franc
républ
paris
réform
conjoint
financ
solidar
national
accord
sécur
numer
commun
conférent
intérieur
sant
holland
sénat
cris
nicol
territorial
énerg
lutt
kouchn
er
bernard
loi
social
charg
mis
rtl
entrepris
gouvern
tourism
sarkozy
fac
not
éduc
michel
franco
industr
entretien
minister
cultur
fonction
justic
étrang
compt
yve
faveur
écolog
union
budget
militair
laurent
log
plan
terror
climat
français
sport
drian
projet
agricultur
collect
servic
transit
territoir
dat
covid
vill
mesur
travail
transport
francophon
person
défens
droit
emmanuel
info
mer
emploi
fabius
publicpress
relat
alain
grand
commerc
aid
assembl
action
marc
polit
europ
syr
femm
mati
international
françois
recherch
élect
étranger
répons
publiqu
enseign
format
supérieur
développ
européen

Ajouter une dimension temporelle dans l'analyse

TF-IDF (*term frequency-inverse document frequency*)

- Mesure statistique → Importance d'un terme contenu dans un texte vis-à-vis d'un corpus de texte. Le poids du mot varie en fonction du nombre d'occurences du mot dans le texte et en fonction du nombre de la fréquence du mot dans le corpus de documents.

```
tidy_text_tfidf <- tidy_text %>%
  dplyr::count(word, dates) %>%
  bind_tf_idf(word, dates, n) %>%
  dplyr::arrange(desc(tf_idf))

head(tidy_text_tfidf)
```

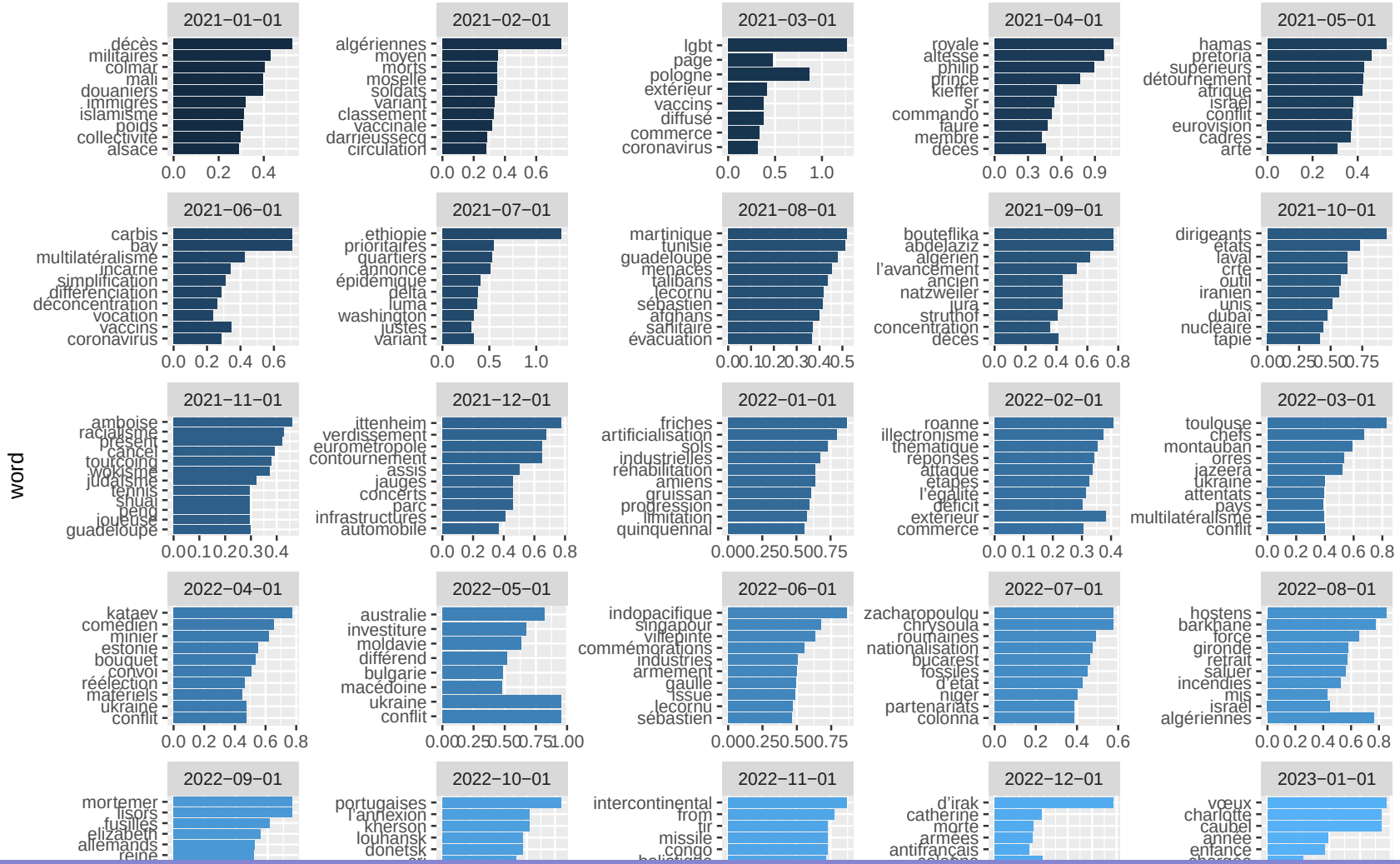
```
## # A tibble: 6 x 6
##   word          dates                n    tf   idf tf_idf
##   <chr>         <chr>            <int> <dbl> <dbl>  <dbl>
## 1 guatemala     18 mai 2013          2 0.222  7.84   1.74
## 2 châteaubriant 23 octobre 2016      2 0.2    8.54   1.71
## 3 karabakh      17 octobre 2020      1 0.25   5.49   1.37
```

La matrice TF-IDF

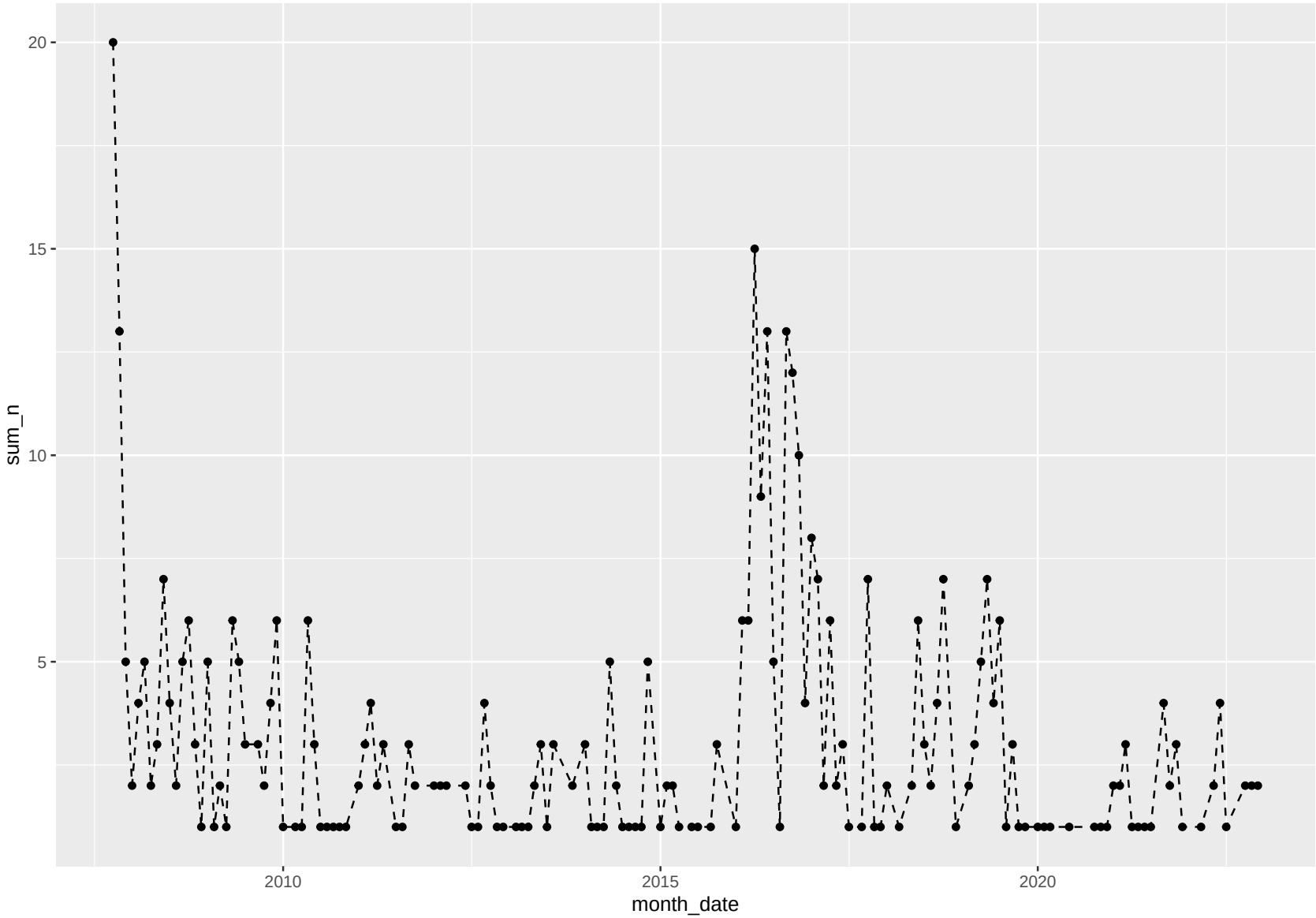
```
tidy_text_tfidf$dates <- as.Date(tidy_text_tfidf$dates, format = c("%d
tidy_text_tfidf$month <- paste("01/",format(tidy_text_tfidf$dates, "%m"
tidy_text_tfidf$month_date <- as.Date(tidy_text_tfidf$month, format=c('
```

Evolution des mots surprésentés par période

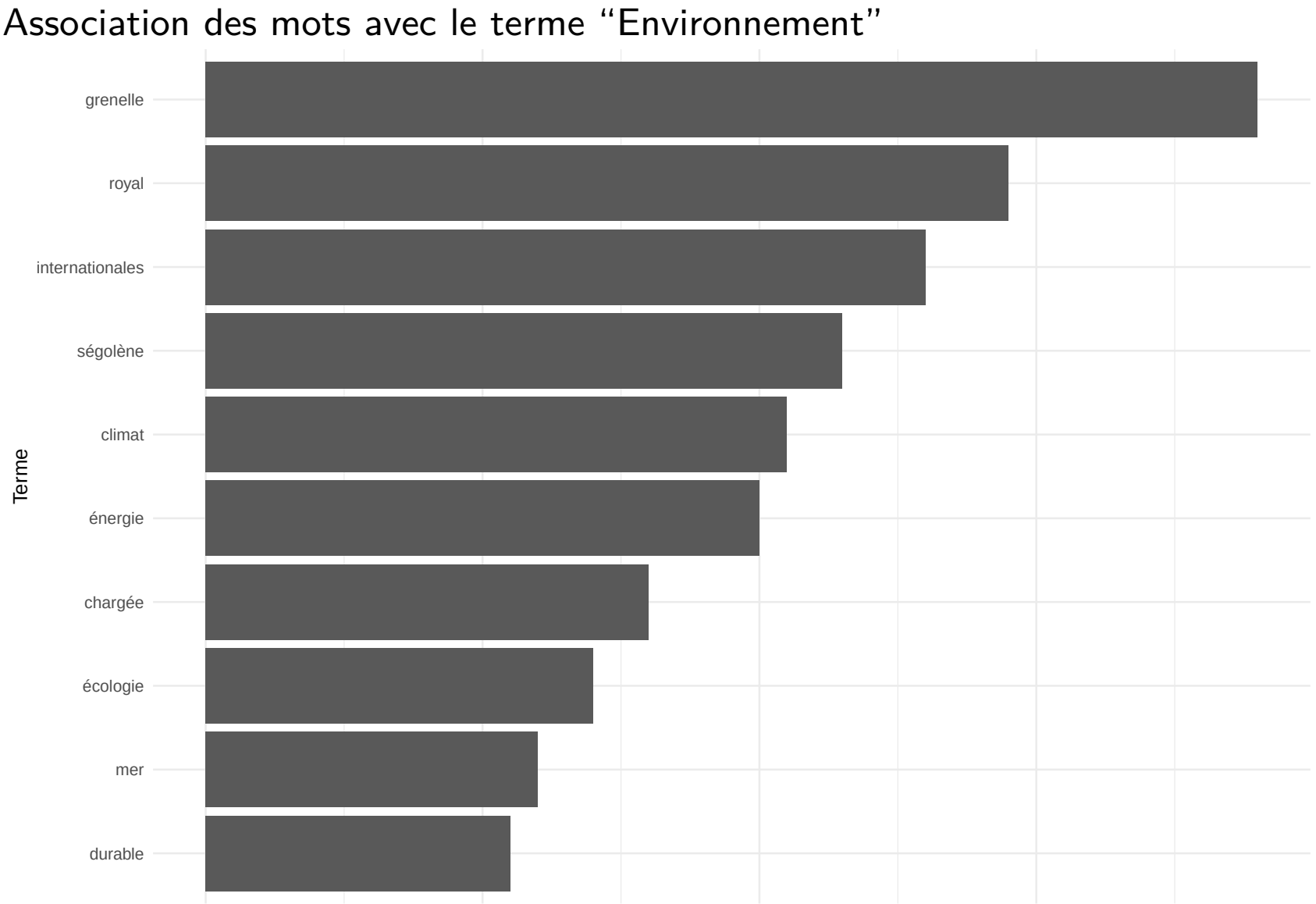
```
## Warning: le package 'ggplot2' a été compilé avec la version R 4.1.3
```



L'occurence des mots de l'écologie au travers du temps

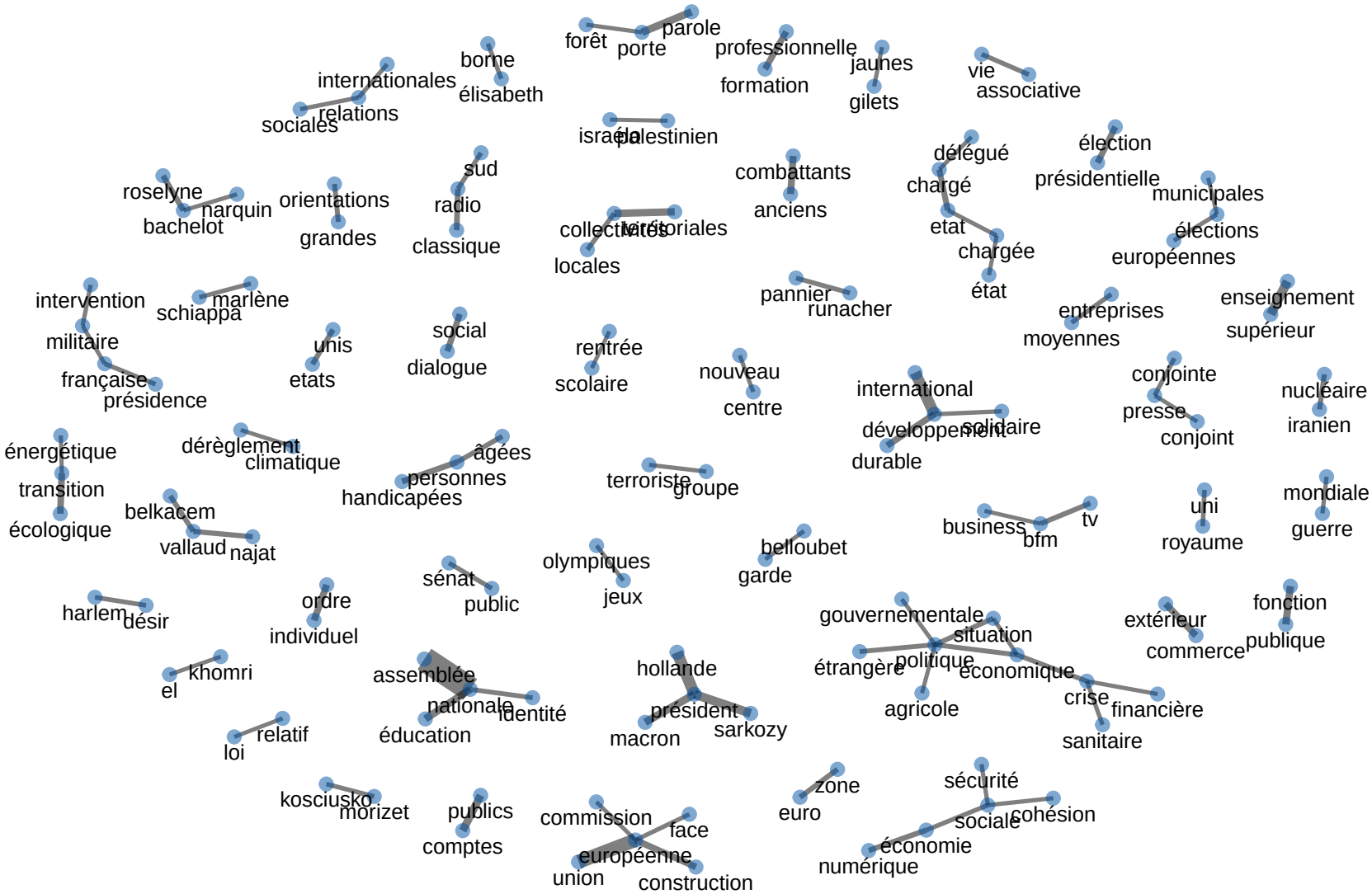


L'association entre les mots



Un graph d'association des mots

Réseau des mots − (2007−2023)



Exercice : Analyse des rapports du GIEC

Extraire de l'information textuelle depuis un pdf

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

```
## Warning: le package 'pdftools' a été compilé avec la version R 4.1.3
```

```
## Using poppler version 22.04.0
```

```
pdf.text_report_2001 <- pdftools::pdf_text("seance_5/2001_policy_report
pdf.text_report_2007 <- pdftools::pdf_text("seance_5/2007_policy_report
pdf.text_report_2014 <- pdftools::pdf_text("seance_5/2014_policy_report

# selection d'une page en particulier
# cat(pdf.text_report_2001[[8]])

pdf.text_report_2001<-unlist(pdf.text_report_2001)
pdf.text_report_2001<-tolower(pdf.text_report_2001)
```