Travail individuel 4

Zouheyr AYAS

# Importation les librairies

# Installer les packages  
#install.packages("tm") # for text mining  
#install.packages("SnowballC") # for text stemming  
#install.packages("wordcloud") # word-cloud generator  
#install.packages("RColorBrewer") # color palettes  
#install.packages("syuzhet") # for sentiment analysis  
#install.packages("ggplot2") # for plotting graphs  
# Charger les libraries  
library("tm")

## Loading required package: NLP

library("SnowballC")  
library("wordcloud")

## Loading required package: RColorBrewer

library("RColorBrewer")  
library("syuzhet")  
library("ggplot2")

##   
## Attaching package: 'ggplot2'

## The following object is masked from 'package:NLP':  
##   
## annotate

# Lecture de texte

text <- readLines(file.choose())

# Chargement des données sous forme de corpus

TextDoc <- Corpus(VectorSource(text))

# Netoyage

toSpace <- content\_transformer(function (x , pattern ) gsub(pattern, " ", x))  
TextDoc <- tm\_map(TextDoc, toSpace, "/")

## Warning in tm\_map.SimpleCorpus(TextDoc, toSpace, "/"): transformation drops  
## documents

TextDoc <- tm\_map(TextDoc, toSpace, "200")

## Warning in tm\_map.SimpleCorpus(TextDoc, toSpace, "200"): transformation drops  
## documents

TextDoc <- tm\_map(TextDoc, toSpace, "\\|")

## Warning in tm\_map.SimpleCorpus(TextDoc, toSpace, "\\|"): transformation drops  
## documents

TextDoc <- tm\_map(TextDoc, toSpace,"™")

## Warning in tm\_map.SimpleCorpus(TextDoc, toSpace, "\231"): transformation drops  
## documents

TextDoc <- tm\_map(TextDoc, toSpace,"€")

## Warning in tm\_map.SimpleCorpus(TextDoc, toSpace, "\200"): transformation drops  
## documents

TextDoc <- tm\_map(TextDoc, toSpace,"â")

## Warning in tm\_map.SimpleCorpus(TextDoc, toSpace, "â"): transformation drops  
## documents

TextDoc <- tm\_map(TextDoc, toSpace,"“")

## Warning in tm\_map.SimpleCorpus(TextDoc, toSpace, "“"): transformation drops  
## documents

TextDoc <- tm\_map(TextDoc, toSpace,"”")

## Warning in tm\_map.SimpleCorpus(TextDoc, toSpace, "”"): transformation drops  
## documents

TextDoc <- tm\_map(TextDoc, toSpace,"€”")

## Warning in tm\_map.SimpleCorpus(TextDoc, toSpace, "\200”"): transformation drops  
## documents

# Transformation en miniscule, élimination des chiffres, et autres

TextDoc <- tm\_map(TextDoc, content\_transformer(tolower))

## Warning in tm\_map.SimpleCorpus(TextDoc, content\_transformer(tolower)):  
## transformation drops documents

TextDoc <- tm\_map(TextDoc, removeNumbers)

## Warning in tm\_map.SimpleCorpus(TextDoc, removeNumbers): transformation drops  
## documents

TextDoc <- tm\_map(TextDoc, removeWords, stopwords("english"))

## Warning in tm\_map.SimpleCorpus(TextDoc, removeWords, stopwords("english")):  
## transformation drops documents

TextDoc <- tm\_map(TextDoc, removeWords, c("s", "company","team"))

## Warning in tm\_map.SimpleCorpus(TextDoc, removeWords, c("s", "company", "team")):  
## transformation drops documents

TextDoc <- tm\_map(TextDoc, removePunctuation)

## Warning in tm\_map.SimpleCorpus(TextDoc, removePunctuation): transformation drops  
## documents

TextDoc <- tm\_map(TextDoc, stripWhitespace)

## Warning in tm\_map.SimpleCorpus(TextDoc, stripWhitespace): transformation drops  
## documents

TextDoc <- tm\_map(TextDoc, stemDocument)

## Warning in tm\_map.SimpleCorpus(TextDoc, stemDocument): transformation drops  
## documents

class(TextDoc)

## [1] "SimpleCorpus" "Corpus"

inspect(TextDoc[[7]])

## <<PlainTextDocument>>  
## Metadata: 7  
## Content: chars: 99  
##   
## thank pioneer leadership lowest violent crime rate quarter centuri cleanest environ quarter centuri

# La Matrice “Termes par Document”

TextDoc\_dtm <- TermDocumentMatrix(TextDoc)  
inspect(TextDoc\_dtm)

## <<TermDocumentMatrix (terms: 1263, documents: 156)>>  
## Non-/sparse entries: 3552/193476  
## Sparsity : 98%  
## Maximal term length: 16  
## Weighting : term frequency (tf)  
## Sample :  
## Docs  
## Terms 132 137 145 22 32 67 74 76 82 88  
## america 1 0 1 0 0 0 0 1 0 0  
## american 0 1 0 0 2 0 1 2 0 0  
## centuri 0 1 1 0 0 0 0 0 2 1  
## must 0 0 0 1 1 0 0 1 1 2  
## new 0 0 0 0 2 0 1 2 0 0  
## now 1 0 1 1 1 2 0 0 1 1  
## secur 0 0 0 3 0 0 0 0 1 0  
## will 0 0 0 0 2 0 1 0 2 0  
## work 0 0 0 0 0 1 1 0 2 1  
## year 0 1 1 4 2 1 3 0 0 0

dtm\_m <- as.matrix(TextDoc\_dtm)

La Matrice “Termes par Document” (suite)

dtm\_v <- sort(rowSums(dtm\_m),decreasing=TRUE)  
dtm\_d <- data.frame(word = names(dtm\_v),freq=dtm\_v)

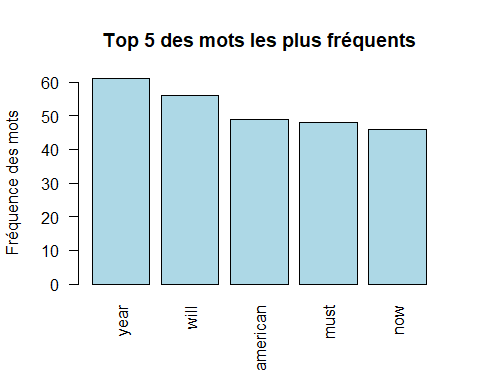
Afficher les mot les plus fréquents , ici on a choisit 15

head(dtm\_d, 15)

## word freq  
## year year 61  
## will will 56  
## american american 49  
## must must 48  
## now now 46  
## work work 42  
## america america 37  
## centuri centuri 36  
## secur secur 33  
## new new 32  
## school school 30  
## nation nation 28  
## support support 28  
## help help 27  
## congress congress 26

Tracer ces mots par un barplot()

barplot(dtm\_d[1:5,]$freq,  
 las = 2,  
 names.arg = dtm\_d[1:5,]$word,  
 col ="lightblue",  
 main ="Top 5 des mots les plus fréquents",  
 ylab = "Fréquence des mots"  
 )



Génération d’un nuage de mots (word cloud)

set.seed(1234)  
wordcloud(words = dtm\_d$word,  
 freq = dtm\_d$freq,  
 min.freq = 5,  
 max.words=50,  
 random.order=FALSE,  
 rot.per=0.40,  
 colors=brewer.pal(8, "Dark2")  
 )

## Warning in wordcloud(words = dtm\_d$word, freq = dtm\_d$freq, min.freq = 5, :  
## today could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = dtm\_d$word, freq = dtm\_d$freq, min.freq = 5, : need  
## could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = dtm\_d$word, freq = dtm\_d$freq, min.freq = 5, : give  
## could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = dtm\_d$word, freq = dtm\_d$freq, min.freq = 5, :  
## propos could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = dtm\_d$word, freq = dtm\_d$freq, min.freq = 5, : care  
## could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = dtm\_d$word, freq = dtm\_d$freq, min.freq = 5, :  
## invest could not be fit on page. It will not be plotted.

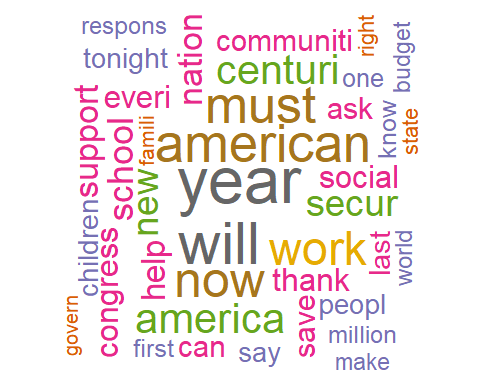
## Warning in wordcloud(words = dtm\_d$word, freq = dtm\_d$freq, min.freq = 5, :  
## medicar could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = dtm\_d$word, freq = dtm\_d$freq, min.freq = 5, : next  
## could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = dtm\_d$word, freq = dtm\_d$freq, min.freq = 5, : meet  
## could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = dtm\_d$word, freq = dtm\_d$freq, min.freq = 5, : keep  
## could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = dtm\_d$word, freq = dtm\_d$freq, min.freq = 5, :  
## trade could not be fit on page. It will not be plotted.

 Génération des associations

# Trouver des associations  
findAssocs(TextDoc\_dtm, terms = c("war","peace","peopl"), corlimit = 0.25)

## $war  
## cold answer depress dispossess largest overcom prejudic   
## 0.75 0.71 0.71 0.71 0.71 0.71 0.71   
## struggl twilight win racial class barrier long   
## 0.71 0.71 0.71 0.60 0.49 0.49 0.46   
## middl lift generat percent point arsenal framework   
## 0.39 0.39 0.38 0.36 0.35 0.35 0.35   
## height iii attack bomber captain desert execut   
## 0.35 0.35 0.35 0.35 0.35 0.35 0.35   
## flawless flew fox jeff machin oper superb   
## 0.35 0.35 0.35 0.35 0.35 0.35 0.35   
## taliaferro advisori alabama board bus goe journey   
## 0.35 0.35 0.35 0.35 0.35 0.35 0.35   
## other rosa sens sought throughout sinc start   
## 0.35 0.35 0.35 0.35 0.35 0.31 0.30   
## end great   
## 0.29 0.29   
##   
## $peace  
## numeric(0)  
##   
## $peopl  
## news podium pride ago welfar hire   
## 0.52 0.52 0.52 0.49 0.48 0.40   
## tonight roll choos lose digniti move   
## 0.35 0.35 0.35 0.35 0.35 0.35   
## partnership republ real access beyond coverag   
## 0.35 0.35 0.35 0.34 0.34 0.34   
## jefford kennedi moynihan offer roth bought   
## 0.34 0.34 0.34 0.34 0.34 0.34   
## expens advisori alabama board bus goe   
## 0.34 0.34 0.34 0.34 0.34 0.34   
## journey other rosa sens sought throughout   
## 0.34 0.34 0.34 0.34 0.34 0.34   
## china good thousand anoth five bring   
## 0.32 0.31 0.31 0.31 0.31 0.30   
## health insur longer get hundr realli   
## 0.29 0.29 0.27 0.27 0.27 0.27   
## past liberti   
## 0.27 0.27

# Trouver des associations pour des mots qui se produisent au moins 50 fois  
findAssocs(TextDoc\_dtm,  
 terms = findFreqTerms(TextDoc\_dtm, lowfreq = 50),  
 corlimit = 0.25)

## $year  
## six last fulfil reserv wise next knew sound   
## 0.48 0.43 0.39 0.39 0.39 0.37 0.37 0.35   
## anoth surplus improv spend five grant enact joint   
## 0.35 0.34 0.33 0.33 0.30 0.29 0.28 0.28   
## patient fifthgrad fiveyear hurt less literaci mount rapid   
## 0.28 0.28 0.28 0.28 0.28 0.28 0.28 0.28   
## skill team train corpor opic untap felon fugit   
## 0.28 0.28 0.28 0.28 0.28 0.28 0.28 0.28   
## murder schedul stalker straight now pass bill oversea   
## 0.28 0.28 0.28 0.28 0.26 0.26 0.26 0.26   
##   
## $will  
## asid divis heal love anoth hope time look reach hundr   
## 0.44 0.42 0.42 0.42 0.39 0.39 0.37 0.37 0.33 0.33   
## exhaust full older suffici unabl educ said cover payment decis   
## 0.31 0.31 0.31 0.31 0.31 0.28 0.28 0.27 0.27 0.27   
## listen five hour shape found ideal   
## 0.27 0.27 0.27 0.27 0.27 0.27

Score des sentiments

syuzhet\_vector <- get\_sentiment(text,method="syuzhet")  
head(syuzhet\_vector)

## [1] 0.9 1.0 3.1 1.0 -1.0 0.0

summary(syuzhet\_vector)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## -4.5000 0.3875 1.6250 1.5506 2.7125 7.4500

#par la methode bing  
bing\_vector <- get\_sentiment(text, method="bing")  
head(bing\_vector)

## [1] 0 1 1 2 0 1

summary(bing\_vector)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## -4.000 0.000 1.000 1.026 2.000 7.000

#par la metheode affin  
afinn\_vector <- get\_sentiment(text, method="afinn")  
head(afinn\_vector)

## [1] 2 2 4 -1 -3 -1

summary(afinn\_vector)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## -16.000 0.000 2.000 2.744 6.000 18.000

Extraction des émotions

d<-get\_nrc\_sentiment(text)  
head (d,10)

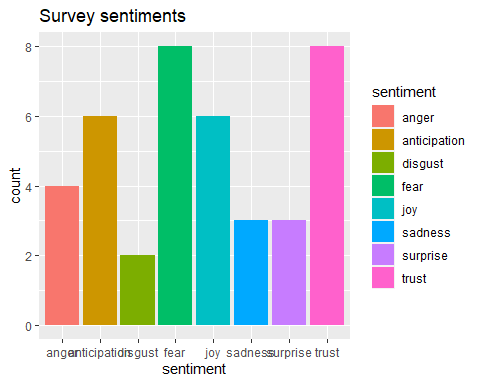
## anger anticipation disgust fear joy sadness surprise trust negative positive  
## 1 0 0 1 0 0 0 0 3 1 2  
## 2 0 0 0 0 0 0 0 1 0 1  
## 3 0 0 0 2 1 0 0 1 0 3  
## 4 0 0 0 0 0 0 0 0 0 2  
## 5 0 2 0 2 3 1 1 0 3 3  
## 6 0 1 0 0 0 0 0 1 1 1  
## 7 2 0 1 1 0 1 1 0 3 0  
## 8 1 1 0 1 1 0 0 1 1 1  
## 9 1 2 1 2 1 1 1 3 3 8  
## 10 0 0 0 0 0 0 0 1 0 1

Classification des émotions (suite)

td<-data.frame(t(d))  
td\_new <- data.frame(rowSums(td[2:10]))  
names(td\_new)[1] <- "count"  
td\_new <- cbind("sentiment" = rownames(td\_new), td\_new)  
rownames(td\_new) <- NULL  
td\_new2<-td\_new[1:8,]

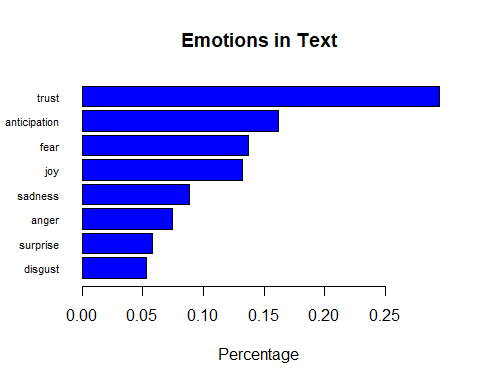
Classification des émotions - nombre de mots associés à chaque sentiment

quickplot(sentiment,  
 data=td\_new2,  
 weight=count,  
 geom="bar",  
 fill=sentiment,  
 ylab="count")+  
 ggtitle("Survey sentiments")



Classification des émotions (suite)

barplot(  
 sort(  
 colSums(prop.table(d[, 1:8]))  
 ),  
 horiz = TRUE,  
 cex.names = 0.7,  
 las = 1,  
 main = "Emotions in Text",  
 xlab="Percentage",  
 col="blue"  
 )

 # Commentaires

Cette analyse de text du président Bill Clinton, lors de son discours annuel tant que president des etats unis, nous montre plein de confiance avec esprit d’anticipation et un sentiement de peur. Il a cité,les mots ‘americans’, ‘must’ , ‘year’ , ‘will’ , qui peut signifier son intention de faire quelques chose cette année pour les americans.