```
suppressWarnings(source("~/GitHub/ajoutRep/ajoutRep/R/sources.R"))
suppressWarnings(library(tidyverse))
```

sommaire

accueil

haut

```
text <- data.frame(accueil())
text <- text[2]

tm <- function(text){
  # Load the data as a corpus
  TextDoc <- Corpus(VectorSource(text))

  #Replacing "/", "@" and "|" with space
  toSpace <- content_transformer(function (x , pattern ) gsub(pattern, " ", x))
  removeSpace <- content_transformer(function (x , pattern ) gsub(pattern, "", x))
  TextDoc <- tm_map(TextDoc, toSpace, "/")
  TextDoc <- tm_map(TextDoc, toSpace, "@")
  TextDoc <- tm_map(TextDoc, toSpace, "\\|")
  # Convert the text to lower case
  TextDoc <- tm_map(TextDoc, content_transformer(tolower))
  # Remove numbers
  # TextDoc <- tm_map(TextDoc, removeNumbers)
  # Remove english common stopwords
  # TextDoc <- tm_map(TextDoc, removeWords, stopwords("english"))
  # Remove your own stop word
  # specify your custom stopwords as a character vector
  TextDoc <- tm_map(TextDoc, removeWords, c("conclu", "conclubis", "eff","the"))
  # Remove punctuations
  TextDoc <- tm_map(TextDoc, removePunctuation)
  # Eliminate extra white spaces
  TextDoc <- tm_map(TextDoc, stripWhitespace)
  # Eliminate spaces
```

```
  # TextDoc <- gsub("[[:blank:]]", "", TextDoc)
  # Text stemming - which reduces words to their root form
  # TextDoc <- tm_map(TextDoc, stemDocument)

  # Build a term-document matrix
  TextDoc_dtm <- TermDocumentMatrix(TextDoc)
  dtm_m <- as.matrix(TextDoc_dtm)
  # Sort by descearing value of frequency
  dtm_v <- sort(rowSums(dtm_m),decreasing=TRUE)
  dtm_d <- data.frame(word = names(dtm_v),freq=dtm_v)
  # Display the top 20 most frequent words
  head(dtm_d, 30)

  # Plot the most frequent words
  barplot(dtm_d[1:15,]$freq, las = 2, names.arg = dtm_d[1:15,]$word,
          col ="lightgreen", main ="Top 15 most frequent words",
          ylab = "Word frequencies")

  #generate word cloud
  # set.seed(1234)
  wordcloud(words = dtm_d$word, freq = dtm_d$freq, min.freq = 1,
            max.words=150, random.order=FALSE, rot.per=0.35,
            colors=brewer.pal(8, "Dark2"))
  dev.print(device = png, file = "accueil.png", width = 800)
}

suppressWarnings(tm(text))
```

page copy

haut

```


main <- function(){
  text <- data.frame(copy())
  # Read the text file from local machine , choose file interactively
  # text2 <- readLines(file.choose())
  # Load the data as a corpus
  TextDoc <- Corpus(VectorSource(text))

  #Replacing "/", "@" and "|" with space
  toSpace <- content_transformer(function (x , pattern ) gsub(pattern, " ", x))
  removeSpace <- content_transformer(function (x , pattern ) gsub(pattern, "", x))
  TextDoc <- tm_map(TextDoc, toSpace, "/")
  TextDoc <- tm_map(TextDoc, toSpace, "@")
  TextDoc <- tm_map(TextDoc, toSpace, "\\|")
  # Convert the text to lower case
  TextDoc <- tm_map(TextDoc, content_transformer(tolower))
  # Remove numbers
  # TextDoc <- tm_map(TextDoc, removeNumbers)
  # Remove english common stopwords
```

```
  # TextDoc <- tm_map(TextDoc, removeWords, stopwords("english"))
  # Remove your own stop word
  # specify your custom stopwords as a character vector
  TextDoc <- tm_map(TextDoc, removeWords, c("conclu", "conclubis", "eff","the"))
  # Remove punctuations
  TextDoc <- tm_map(TextDoc, removePunctuation)
  # Eliminate extra white spaces
  TextDoc <- tm_map(TextDoc, stripWhitespace)
  # Eliminate spaces
  # TextDoc <- gsub("[[:blank:]]", "", TextDoc)
  # Text stemming - which reduces words to their root form
  # TextDoc <- tm_map(TextDoc, stemDocument)

  # Build a term-document matrix
  TextDoc_dtm <- TermDocumentMatrix(TextDoc)
  dtm_m <- as.matrix(TextDoc_dtm)
  # Sort by descearing value of frequency
  dtm_v <- sort(rowSums(dtm_m),decreasing=TRUE)
  dtm_d <- data.frame(word = names(dtm_v) ,freq=dtm_v)
  # Display the top 20 most frequent words
  head(dtm_d, 30)

  # Plot the most frequent words
  barplot(dtm_d[1:15,]$freq, las = 2, names.arg = dtm_d[1:15,]$word,
          col ="lightgreen", main ="Top 15 most frequent words",
          ylab = "Word frequencies")

  #generate word cloud
  # set.seed(1234)
  wordcloud(words = dtm_d$word, freq = dtm_d$freq, min.freq = 1,
            max.words=150, random.order=FALSE, rot.per=0.35,
            colors=brewer.pal(8, "Dark2"))
  dev.print(device = png, file = "copy.png", width = 600)

}

suppressWarnings(main())
```

copy max

haut

```
main <- function(){
  text2 <- copy.max()
  text2 <- text2[2]
  # Read the text file from local machine , choose file interactively
  # text2 <- readLines(file.choose())
  # Load the data as a corpus
  TextDoc <- Corpus(VectorSource(text2))
```

```
  #Replacing "/", "@" and "|" with space
  toSpace <- content_transformer(function (x , pattern ) gsub(pattern, " ", x))
  removeSpace <- content_transformer(function (x , pattern ) gsub(pattern, "", x))
  TextDoc <- tm_map(TextDoc, toSpace, "/")
  TextDoc <- tm_map(TextDoc, toSpace, "@")
  TextDoc <- tm_map(TextDoc, toSpace, "\\|")
  # Convert the text to lower case
  TextDoc <- tm_map(TextDoc, content_transformer(tolower))
  # Remove numbers
  # TextDoc <- tm_map(TextDoc, removeNumbers)
  # Remove english common stopwords
  # TextDoc <- tm_map(TextDoc, removeWords, stopwords("english"))
  # Remove your own stop word
  # specify your custom stopwords as a character vector
  TextDoc <- tm_map(TextDoc, removeWords, c("conclu", "conclubis",
"eff","na","the","conclucompi","conclucompibis"))
  # Remove punctuations
  TextDoc <- tm_map(TextDoc, removePunctuation)
  # Eliminate extra white spaces
  TextDoc <- tm_map(TextDoc, stripWhitespace)
  # Eliminate spaces
  # TextDoc <- gsub("[[:blank:]]", "", TextDoc)
  # Text stemming - which reduces words to their root form
  # TextDoc <- tm_map(TextDoc, stemDocument)

  # Build a term-document matrix
  TextDoc_dtm <- TermDocumentMatrix(TextDoc)
  dtm_m <- as.matrix(TextDoc_dtm)
  # Sort by descearing value of frequency
  dtm_v <- sort(rowSums(dtm_m),decreasing=TRUE)
  dtm_d <- data.frame(word = names(dtm_v),freq=dtm_v)
  # Display the top 20 most frequent words
  head(dtm_d, 30)

  # Plot the most frequent words
  barplot(
    dtm_d[1:15,]$freq,
    las = 2,
    names.arg = dtm_d[1:15,]$word,
    col ="lightgreen",
    main ="Top 15 most frequent words",
    ylab = "Word frequencies")

  #generate word cloud
  # set.seed(1234)
  wordcloud(words = dtm_d$word, freq = dtm_d$freq, min.freq = 1,
            max.words=200, random.order=FALSE, rot.per=0.35,
            colors=brewer.pal(8, "Dark2"))
  dev.print(device = png, file = "max.png", width = 600)

}

suppressWarnings(main())
```

test resume

haut

```
main <- function(){
  text2 <- data.frame(aTestResume())
  text2 <- text2[2]
  # Read the text file from local machine , choose file interactively
  # text2 <- readLines(file.choose())
  # Load the data as a corpus
  TextDoc <- Corpus(VectorSource(text2))

  #Replacing "/", "@" and "|" with space
  toSpace <- content_transformer(function (x , pattern ) gsub(pattern, " ", x))
  removeSpace <- content_transformer(function (x , pattern ) gsub(pattern, "", x))
  TextDoc <- tm_map(TextDoc, toSpace, "/")
  TextDoc <- tm_map(TextDoc, toSpace, "@")
  TextDoc <- tm_map(TextDoc, toSpace, "\\|")
  # Convert the text to lower case
  TextDoc <- tm_map(TextDoc, content_transformer(tolower))
  # Remove numbers
  # TextDoc <- tm_map(TextDoc, removeNumbers)
  # Remove english common stopwords
  # TextDoc <- tm_map(TextDoc, removeWords, stopwords("english"))
  # Remove your own stop word
  # specify your custom stopwords as a character vector
  TextDoc <- tm_map(TextDoc, removeWords, c("conclu", "conclubis",
"eff","na","conclucompi","conclucompibis"))
  # Remove punctuations
  TextDoc <- tm_map(TextDoc, removePunctuation)
  # Eliminate extra white spaces
  TextDoc <- tm_map(TextDoc, stripWhitespace)
  # Eliminate spaces
  # TextDoc <- gsub("[[:blank:]]", "", TextDoc)
  # Text stemming - which reduces words to their root form
  # TextDoc <- tm_map(TextDoc, stemDocument)

  # Build a term-document matrix
  TextDoc_dtm <- TermDocumentMatrix(TextDoc)
  dtm_m <- as.matrix(TextDoc_dtm)
  # Sort by descearing value of frequency
  dtm_v <- sort(rowSums(dtm_m),decreasing=TRUE)
  dtm_d <- data.frame(word = names(dtm_v),freq=dtm_v)
  # Display the top 20 most frequent words
  head(dtm_d, 30)

  # Plot the most frequent words
  barplot(dtm_d[1:15,]$freq, las = 2, names.arg = dtm_d[1:15,]$word,
          col ="lightgreen", main ="Top 15 most frequent words",
          ylab = "Word frequencies")
```

```
  #generate word cloud
  # set.seed(1234)
  wordcloud(words = dtm_d$word, freq = dtm_d$freq, min.freq = 1,
            max.words=200, random.order=FALSE, rot.per=0.35,
            colors=brewer.pal(8, "Dark2"))
  dev.print(device = png, file = "ec.png", width = 600)

}

suppressWarnings(main())
```

lettre

haut

```
data <- data.frame(listDesc.desc())
data %>% filter(episode == 1,status=="en cours") %>%
select(descr.img,tisaep,Horodateur)
```

liste EC : recherche alea

haut

```
data <- data.frame(listDesc.desc())
max <- 100
data %>% filter(status == "en cours") %>% sample_n(max) %>% group_by(tisa) %>%
summarise(pc = n()) %>% arrange(desc(pc)) %>% filter(pc >= 2)
```

liste EC : recherche par Titre

haut

```
suppressWarnings(library(tidyverse))
df <- data.frame(listDesc.ec())
data <- data.frame(listDesc.desc())
data <- distinct(data)
# titre = readline()
titre = df[1,2]
data %>% filter(Titre == titre) %>% select(tisaep,Horodateur)
if(data[2,2] == data[1,2]){
  titre = df[3,2]
} else {
```

```
  titre = df[2,2]
}
data %>% filter(Titre == titre) %>% select(tisaep,Horodateur)
```

liste EC

haut

```
suppressWarnings(library(tidyverse))
data <- data.frame(listDesc.desc())
data %>% filter(status == "en cours") %>% select(tisaep,Horodateur)
```

liste TER

haut

```
suppressWarnings(library(tidyverse))
data <- data.frame(listDesc.desc())
data %>% filter(status == "terminée") %>% select(tisaep,Horodateur)
```

liste film

haut

```
suppressWarnings(library(tidyverse))
data <- data.frame(listDesc.desc())
data %>% filter(status == "film") %>% select(tisaep,Horodateur)
```