# TP3 (Analyse de données textuelles)

**ÉNONCÉ GÉNÉRAL**

Dans ce TP, vous allez faire une analyse textuelle avec les bases de données de votre choix. Comme pour le TP2, vous devez trouver les données textuelles que vous allez utiliser dans votre travail, mais je reste disponible pour vous conseiller. L'objectif de ce TP est de vous introduire à l'analyse textuelle automatisée et d'écrire un rapport de recherche décrivant sommairement vos données, les méthodes mobilisées et vos résultats.

Comme pour le TP2, je veux voir votre code de nettoyage et d'analyse de données dans une annexe à la fin de votre document. Nous n'avons pas d'invité(e) dans la deuxième moitié des cours 5 et 6. Nous profiterons de ces périodes de temps pour vous introduire à l'analyse textuelle tranquillement et pour avancer dans votre TP3.

## Consignes et étapes:

**ANALYSE DEMMANDÉE:**

L'analyse de base demandée pour votre TP3 est une analyse du dictionnaire. Je vais introduire cette méthode en classe lors du cours 5. Si vous avez besoin de sources scientifiques pertinentes sur l'analyse par dictionnaires, je vous conseille de consulter l'article de Young et Soroka (2012) qui introduit et utilise le dictionnaire ***Lexicoder*** pour analyser des données textuelles tirées de sources médiatiques. Vous pouvez également consulter l'article de Duval et Pétry (2016). Ces derniers utilisent la version française de ce dictionnaire.

Vous pouvez utiliser un dictionnaire existant (comme le ***Lexicoder***) ou en développer un de votre côté. Vous devez justifier la raison pour laquelle vous avez utilisé ou créé le dictionnaire mobilisé dans votre recherche (*par rapport aux données utilisées dans votre travail*).

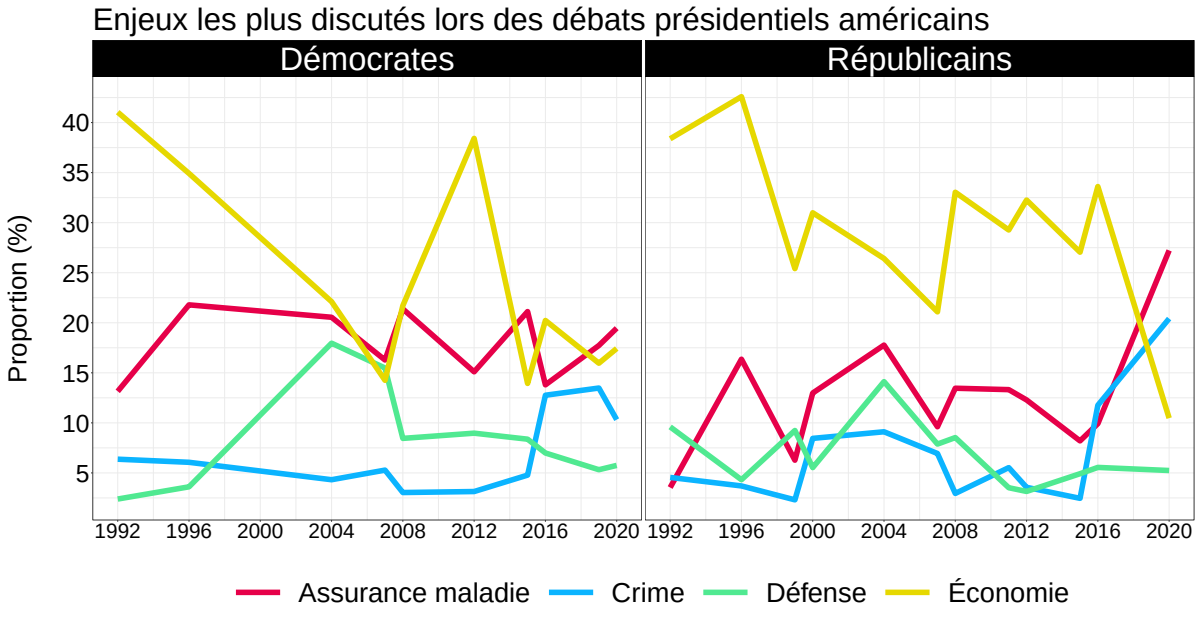
**Un 0.5 point BONUS** sera attribué aux étudiant-es qui utiliseront une méthode plus poussée que l'analyse du dictionnaire (comme des méthodes non supervisées d'analyse textuelle) pour leur TP3.

**1:** Écrire une courte introduction d'une demi-page à une page expliquant la question de recherche et les données utilisées pour répondre à cette dernière.

**2:** Écrire une section sur les données utilisées d'environ une page. Je veux également que vous décriviez votre processus de nettoyage de vos données textuelles dans cette section (exemple: division du corpus par phrases, par mots, retrait des *stopwords*, `lemmatization` ou `stemming`, etc.). Pour un refresh sur le `lemmatization` ou sur le `stemming`, ou sur les *stopwords* veuillez consulter le texte de Grimmer et Stewart (2013). Si vous avez besoin de plus de sources scientifiques pertinentes, je peux vous en suggérer. C'est aussi dans cette section que vous devez commenter le dictionnaire utilisé (pourquoi ce dictionnaire, pourquoi l'avoir développé dans le cas où vous l'avez développé, par rapport à vos données, etc.)

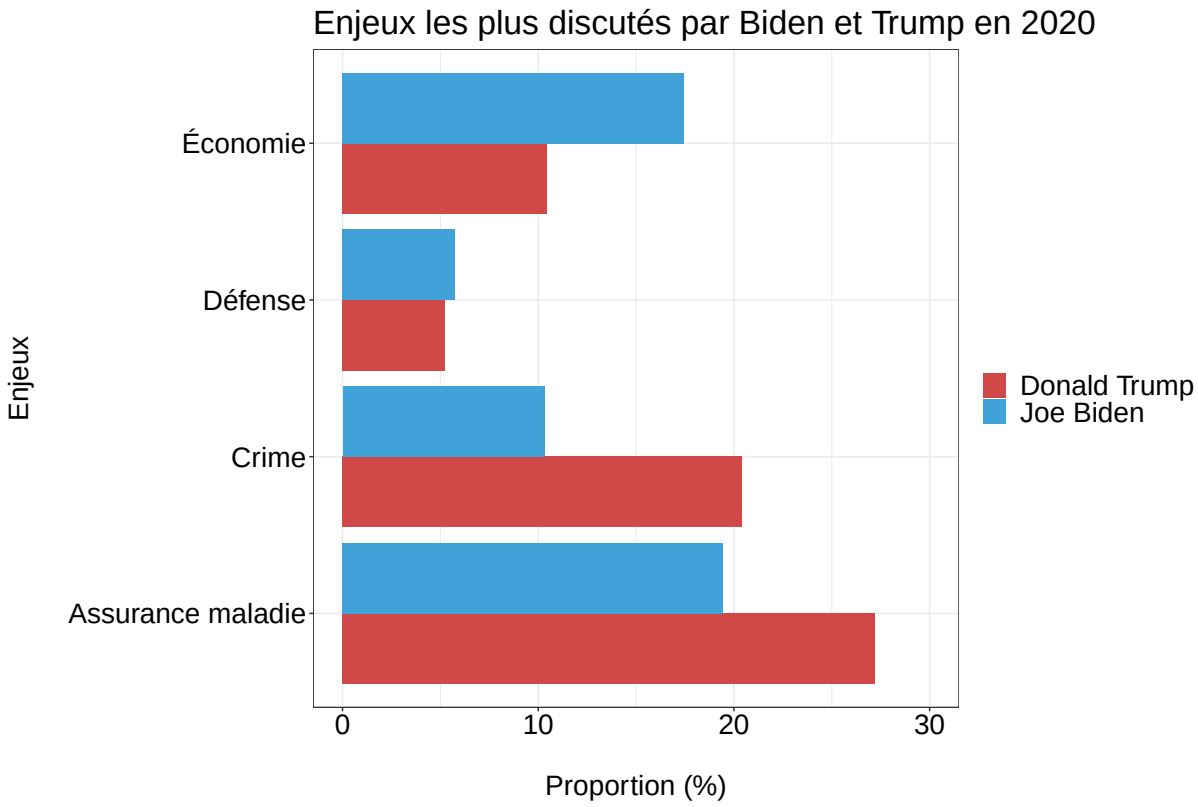
**3:** Faites une section qui décrit vos résultats. Dans cette section, présentez un graphique qui montre les résultats de votre analyse du dictionnaire. Cela peut être un graphique longitudinal.

**Exemple:**



Ou vous pouvez produire un graphique à barres.

**Exemple:**




Ces graphiques sont des exemples, utilisez le graphique qui vous permet le mieux de représenter vos données et analyses. Cette section devrait vous prendre 1 à 2 pages maximum.

**4:** Terminez votre TP3 par une petite conclusion d'une demi-page résumant vos résultats. Remettez vos fiches sur `GitHub` en suivant les procédures habituelles.

**ATTENTION**

**RAPPEL DE LA DATE DE REMISE: lundi 19 février avant minuit**.

## Évaluation (sur 5 points):

- **Justification des bases de données utilisées (cas d'étude, pourquoi ces bases de données aident-elles à répondre à la question de recherche): 1.5 point**.
- **Explication du dictionnaire utilisé ou créé, du processus de nettoyage des données textuelles (traitement du texte, `lemmatization`, `stemming`, stopwords, etc.): 2 points**
- **Description des résultats: 1.5 point**
- **BONUS #1 - Utilisation d'une méthode non supervisée pour les analyses: 0.5 point**.
- **BONUS #2 - Visualisation des mots les plus utilisés dans le corpus dans la section 2 du texte (exemple: graphique à barres montrant la fréquence des mots dans le texte, nuage de mots ou *wordcloud*): 0.5 point**

## Bibliographie

Duval, Dominic, et François Pétry. 2016. « L'analyse Automatisée du Ton Médiatique: Construction et Utilisation de la Version Française du Lexicoder Sentiment Dictionary ». *Canadian Journal of Political Science/Revue Canadienne de Science Politique* 49 (2): 197-220.

Grimmer, Justin, et Brandon M Stewart. 2013. « Text as Data: The Promise and Pitfalls of Automatic Content Analysis Methods for Political Texts ». *Political Analysis* 21 (3): 267-97.

Young, Lori, et Stuart Soroka. 2012. « Affective News: The Automated Coding of Sentiment in Political Texts ». *Political Communication* 29 (2): 205-31.

## Annexe

**Code des analyses:**

```
# 0.1 - Libraries ----

#install.packages("crayon")
library(crayon) # couleur du code
#install.packages("quanteda")
library(quanteda) # Package pour faire des analyses textuelles
#install.packages("tidyverse")
suppressMessages(library(tidyverse)) # Data wrangling

# 0.2 - Data (Données de débats américains) ----

Data <- read_csv("_data/debate_documents_dataset_clean.csv")

# 0.3 - Fonctions pour rouler des dictionnaires ----

# Ça :

run_dictionary <- function(data, text, dictionary) {
  tictoc::tic()
  if ( is.data.frame(data) != "TRUE") {
    stop(crayon::yellow('the argument "data" needs to be a dataframe'))
  }
  data <- data %>% dplyr::mutate(text = {{text}})
  if ( is.character(data$text) != "TRUE") {
    stop(crayon::yellow('The variable "text" needs to be a character vector'))
  }
  corpus <- quanteda::tokens(data$text)
  if ( quanteda::is.dictionary(dictionary) != "TRUE") {
    stop(crayon::yellow('Your "dictionary" needs to be in a dictionary format\n For more i
  }
  dfm    <- quanteda::dfm(quanteda::tokens_lookup(corpus, dictionary, nested_scope = "dict
  message(crayon::green("100% expressions/words found"))
  dataFinal   <- quanteda::convert(dfm, to = "data.frame")
  tictoc::toc()
  return(dataFinal)
}

# Ou ça...:
```

```r
#devtools::install_github("clessn/clessnverse")

# 0.4 - Dictionnaire Lexicoder ----

# Transformation d'une base de données en dictionnaire

topic_dictionary_en <- read_csv("_dictionnaire/lexicoder_merged.csv") |>
  rename(language = motAnglais) |>
  select(language, categorie) |>
  na.omit() |>
  filter(language != "x") |>
  unstack(., form=language~categorie) |>
  dictionary()

# Exemple 1

Data_candidats <- Data |>
  # Garder les variables les plus importantes #
  select(speaker, year, text, party) |>
  # Garder juste les textes des candidats à la présidence Après 1992 #
  filter(year %in% c(1992:2020),
         speaker %in% c("George H. Bush", "Bob Dole", "George W. Bush",
                        "John McCain", "Mitt Romney", "Donald Trump",
                        "William (Bill) Clinton", " Al Gore", "John Kerry",
                        "Barack Obama", "Hillary Clinton", "Joe Biden")) |>
  # Regroupement des textes par année et parti #
  group_by(year, party) |>
  # On colle les textes par parti par années + minuscules #
  mutate(text = tolower(paste0(text, collapse = " "))) |>
  distinct() |>
  na.omit()

# Exemple 2

Data_biden_trump <- Data |>
  # Garder les variables les plus importantes #
  select(speaker, year, text) |>
  # On garde Trump et Bident #
  filter(year == 2020,
         speaker %in% c("Joe Biden", "Donald Trump")) |>
  # Regroupement des textes par speaker #
```

```r
  group_by(speaker) |>
  # On colle les textes par parti par années + minuscules #
  mutate(text = tolower(paste0(text, collapse = " "))) |>
  distinct() |>
  na.omit()

# 1 - Analyse du dictionnaire pour l'exemple 1 ----

Data_candidats_clean <- run_dictionary(data = Data_candidats, text = text,
                                       dictionary = topic_dictionary_en) |>
  # On reprend les mêmes colonnes pour l'analyse #
  bind_cols(Data_candidats) |>
  select(-c(doc_id,text,speaker)) |>
  # Pivote la base de données pour voir la proportion par sujet #
  pivot_longer(!c(year, party), names_to = "categorie", values_to="n") |>
  ungroup() |>
  group_by(year, party, categorie) |>
  summarise(n=sum(n)) |>
  mutate(prop = round(n/sum(n),4)*100,
         party = case_when(party == "Republican" ~ "Républicains",
                           party == "Democratic" ~ "Démocrates"),
         categorie = case_when(categorie == "macroeconomics" ~ "Économie",
                               categorie == "crime" ~ "Crime",
                               categorie == "healthcare" ~ "Assurance maladie",
                               categorie == "defence" ~ "Défense",
                               T ~ as.character(categorie))) |>
  # À des fins d'exemple on garde seulement certains thèmes #
  filter(categorie %in% c("Économie", "Crime", "Assurance maladie", "Défense"))

# 1.2 - Graph ----

ggplot(Data_candidats_clean, aes(x = year, y = prop, color = categorie)) +
  expand_limits(x =1992:2020) +
  geom_line(linewidth=4) +
  facet_wrap(~party) +
  scale_y_continuous(breaks = seq(0, 40, 5)) +
  scale_x_continuous(breaks = seq(1992, 2020, 4)) +
  scale_color_manual("", values = c("#e60049", "#0bb4ff", "#50e991", "#e6d800")) +
  labs(x = "",
       y = "Proportion (%)\n",
       title = "Enjeux les plus discutés lors des débats présidentiels américains") +
```

```
  theme_bw() +
  ## theme() en fonction des dimensions dans Quarto ##
  theme(title = element_text(size = 40),
        legend.text = element_text(size = 42),
        axis.title = element_text(size = 40, color = "black"),
        axis.text.x = element_text(size = 32, color = "black"),
        axis.text.y = element_text(size = 38, color = "black"),
        legend.key.width = unit(3,"cm"),
        legend.position = "bottom",
        strip.text = element_text(size = 50, color = "white"),
        strip.background = element_rect(color="black",
                                        fill = "black"))


# 2 - Analyse du dictionnaire pour l'exemple 2 ----

Data_biden_trump_clean <- run_dictionary(data = Data_biden_trump, text = text,
                                         dictionary = topic_dictionary_en) |>
  # On reprend les mêmes colonnes pour l'analyse #
  bind_cols(Data_biden_trump) |>
  select(-c(doc_id,year,text)) |>
  # Pivote la base de données pour voir la proportion par speaker #
  pivot_longer(!speaker, names_to = "categorie", values_to="n") |>
  ungroup() |>
  group_by(speaker, categorie) |>
  summarise(n=sum(n)) |>
  mutate(prop = round(n/sum(n),4)*100,
         categorie = case_when(categorie == "macroeconomics" ~ "Économie",
                               categorie == "crime" ~ "Crime",
                               categorie == "healthcare" ~ "Assurance maladie",
                               categorie == "defence" ~ "Défense",
                               T ~ as.character(categorie))) |>
  # À des fins d'exemple on garde seulement certains thèmes #
  filter(categorie %in% c("Économie", "Crime", "Assurance maladie", "Défense"))

# 2.1 - Graph ----

ggplot(Data_biden_trump_clean, aes(x = categorie, y = prop, fill = speaker)) +
  expand_limits(y=0:30) +
  geom_bar(stat = "identity", position = "dodge") +
  scale_fill_manual("", values = c("#D04848", "#40A2D8")) +
```

```
coord_flip() +
labs(x = "Enjeux\n",
     y = "\nProportion (%)",
     title = "Enjeux les plus discutés par Biden et Trump en 2020") +
theme_bw() +
## theme() en fonction des dimensions dans Quarto ##
theme(title = element_text(size = 20),
      legend.text = element_text(size = 20),
      axis.text = element_text(size = 20, color = "black"))
```