# Comptage et estimation

Guillaume Wisniewski
guillaume.wisniewski@u-paris.fr

septembre 2020

L'objectif de ce TP est de mettre en évidence deux lois fondamentales de la linguistique de corpus, la loi de Zipf et la loi de Heaps, à l'aide de commandes shell. Vous devrez m'envoyer par mail un compte-rendu de votre travail avant le **30 septembre à 8h**. Ce compte-rendu, au format pdf, comportera la réponse aux 16 questions de ce sujet.

## 1 Préparation des données

Vous trouverez sur le site du cours une archive contenant plusieurs ouvrages écrits par Émile Zola que nous utiliserons pour *estimer* la probabilité d'apparition d'une lettre dans un texte en français. La première étape de cette estimation consiste à « préparer » ces documents en :

— supprimant les espaces, retours à la ligne et tous les autres caractères de « mise en page » (on pourra par contre garder les ponctuations)

— supprimant les « entêtes » et notamment les méta-données telles le nom de la personne ayant numérisé le livre, le copyright, ...

— passant tous les caractères en minuscules.

① Écrivez une fonction prenant en paramètre le nom du répertoire dans lequel sont stockés les documents et retournant une chaine de caractère contenant les textes « nettoyés » (c.-à-d. sans espaces et sans entêtes).

② Combien de caractères différents sont utilisés ?

## 2 Comptage

③ Déterminez la fréquence d'apparition de chaque caractère dans le corpus que vous avez constitué à la question précédente. Quels sont les 5 caractères les plus fréquents ? les moins fréquents ?

④ Quel est l'impact de la suppression des entêtes sur le calcul de ces fréquences ?

⑤ Déterminez la fréquence d'apparition des lettres `a` et `t`, lorsque seuls les $i$ premiers caractères de votre corpus sont pris en compte. On fera varier $i$ entre 100 et 100 000 par pas de 100. Représentez graphiquement, pour chaque caractère, la fréquence en fonction de $i$. Qu'en concluez-vous ?

⑥ Reprenez la question précédente lorsque les $i$ caractères sont choisis aléatoirement dans votre corpus. On pourra utiliser la fonction `sample` du module `random`.

⑦ Que se passe-t-il se vous exécutez plusieurs fois le code vous ayant permis de répondre à la question précédente ? Pourquoi ?