SORBONNE UNIVERSITÉ

Chiffrements mono-alphabétiques en pratique

Version du 12 janvier 2023

TME

Le langage utilisé en TME est Python. Des squelettes de programmes dans ce langage vous seront fournis à l'occasion.

En cas de besoin nous vous proposons des rappels de Python sur le site de l'UE.

Exercice 1 – Fréquence des lettres dans une langue donnée

Le but de cet exercice est d'établir l'histogramme des fréquences des lettres dans un fichier passé en paramètre. Vous sont fournis deux scripts :

— `nettoie` qui prend un fichier composé d'une seule ligne d'une longueur quelconque en premier paramètre et écrit dans le fichier passé en second paramètre le texte en majuscules sans accents, ni espaces ni ponctuation. Vos programmes peuvent donc travailler sur des textes écrits dans ce format simplifié. Vous pouvez aussi lire la ligne à nettoyer sur l'entrée standard et afficher sur la sortie standard.

```
% ./nettoie --help
Usage   nettoie <fichier_à_nettoyer> <fichier_nettoyé>
ou      nettoie <<< "<chaîne à nettoyer>"
```

— `dessine_histogramme` affiche un histogramme à partir du résultat d'une commande sous la forme de lignes `<`*caractère*`>` `<`*fréquence*`>`.

Remarque :

Cette commande utilise ***gnuplot*** *(qui a besoin de* ***imagemagick*** *pour afficher le résultat dans une fenêtre). Si vous voulez vous en servir sur votre machine personnelle il faut installer ces deux packages.*

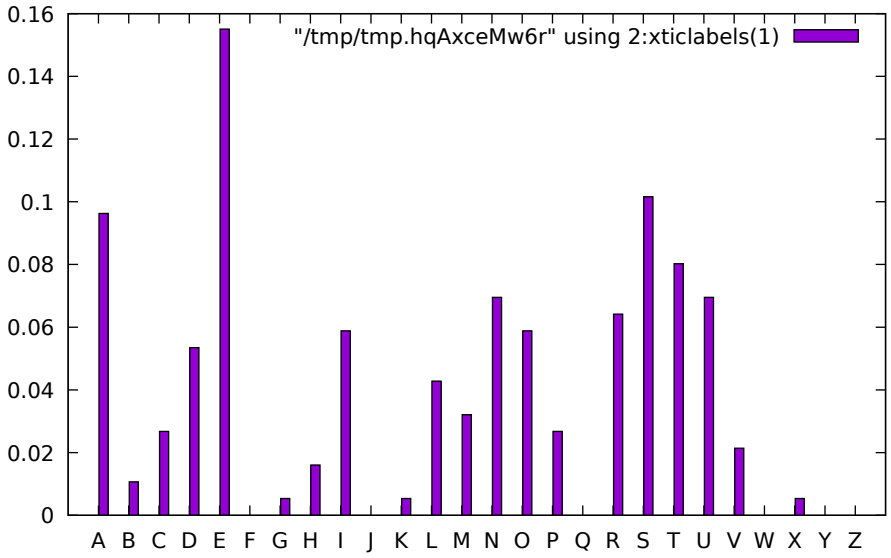
```
% dessine_histogramme -h
Usage:
    ./dessine_histogramme [-h] [-pdf] <calcul_de_frequences> <fichier_à_analyser>
    Exemples:
      ./dessine_histogramme ./frequence germinal_nettoye
         ouvre une fenêtre où l'histogramme est visualisé
      ./dessine_histogramme -pdf python3 frequence.py germinal_nettoye
         crée un <fichier_à_analyser>.pdf de l'histogramme

ou
    <calcul_de_frequences> | ./dessine_histogramme [-h] [-pdf]
    Exemples:
      ./nettoie < germinal_very_small | ./frequence.py | ./dessine_histogramme
         ouvre une fenêtre où l'histogramme du fichier est affiché
      ./nettoie <<< "Ceci est une phrase" | ./frequence.py | ./dessine_histogramme
         ouvre une fenêtre où l'histogramme de la phrase est affiché
```

Vous devez écrire le programme `frequence.py` qui produit les informations fournies à ce script.

Pour vous y retrouver voici un exemple de trace d'exécution :

```
# Version à base de fichiers
% cat germinal_very_small
Dans la plaine rase, sous la nuit sans étoiles, d'une obscurité et d'une épaisseur d'encre,\
un homme suivait seul la grande route de Marchiennes à Montsou, dix kilomètres de pavé cou\
```

```
pant tout droit, à travers les champs de betteraves.
% ./nettoie germinal_very_small germinal_very_small_nettoye
% cat germinal_very_small_nettoye
DANSLAPLAINERASESOUSLANUITSANSETOILESDUNEOBSCURITEETDUNEEPAISSEURDENCREUNHOMMESUIVAITSEULL\
AGRANDEROUTEDEMARCHIENNESAMONTSOUDIXKILOMETRESDEPAVECOUPANTTOUTDROITATRAVERSLESCHAMPSDEBET\
TERAVES%
% python3 frequence.py germinal_very_small_nettoye
A 0.0962566844919786
B 0.0106951871657754
C 0.026737967914438502
D 0.053475935828877004
E 0.15508021390374332
F 0.0
G 0.0053475935828877
H 0.016042780748663103
I 0.058823529411764705
J 0.0
K 0.0053475935828877
L 0.0427807486631016
M 0.03208556149732621
N 0.06951871657754011
O 0.058823529411764705
P 0.026737967914438502
Q 0.0
R 0.06417112299465241
S 0.10160427807486631
T 0.08021390374331551
U 0.06951871657754011
V 0.0213903743315508
W 0.0
X 0.0053475935828877
Y 0.0
Z 0.0
% ./dessine_histogramme python3 frequence.py germinal_very_small_nettoye
```



ou sans fichier intermédiaire à partir d'un texte dans un fichier ou directement une phrase :

```
# Version sans fichier intermédiaire
% ./nettoie <<< "Ceci est une phrase" | ./frequence.py | ./dessine_histogramme
% ./nettoie < germinal_very_small | ./frequence.py | ./dessine_histogramme
```

1. Écrire un programme Python `frequence.py` qui étant donné un nom de fichier texte codé en ASCII comme argument analyse la fréquence d'apparition de chacun de ses caractères.

Il s'agit essentiellement d'écrire une fonction qui affiche un tableau de correspondance caractère $\longrightarrow$ fréquence sous forme de pourcentage (on pourra commencer par renvoyer le nombre d'apparition de chaque lettre et le nombre total de lettres). Cette fonction sera réutilisée dans les exercices et séances suivantes pour la cryptanalyse de textes. Ici on se contente d'illustrer cette fonction avec un programme principal qui affiche le tableau.

2. Établir un histogramme des fréquences pour des textes écrits en français, en anglais ou tout autre langue (on pourra consulter le site web).

   Ici il s'agit d'établir des tableaux de fréquences de référence pour différentes langues, qui nous serviront de comparaison par la suite pour la cryptanalyse de textes.

   Pour ce faire, vous utiliserez la fonction de la question précédente sur un texte d'une langue donnée. Ces textes seront supposés écrits en majuscules, sans ponctuation et suffisamment longs. Pour produire de tels fichiers texte, vous pourrez utiliser les sites web `http://abu.cnam.fr/index.html` pour des textes en français exclusivement ou `http://www.gutenberg.org/wiki/Main_Page` et les mettre au format simplifié avec le script `nettoie`.

   Le script `dessine_histogramme` vous fournit une représentation graphique du résultat.

   Comparer les histogrammes de plusieurs textes écrits dans la même langue, observer les similitudes.

3. Les fichiers `texte1` à `texte10` fournis sous Moodle sont écrits dans 5 langues 2 par 2. Grâce à leur histogramme de fréquences retrouver les textes écrits dans la même langue. Procédez de la même façon en utilisant les distingueurs vus en cours.

Exercice 2 – Chiffrements et déchiffrements

1. Écrire un programme permettant de réaliser le chiffrement et déchiffrement de César.

   Vous spécifierez bien votre programme et vérifierez son comportement sur les exemples vus en TD.

   Comparez l'histogramme des fréquences du clair et du chiffré.

2. Écrire un programme permettant de réaliser un chiffrement et déchiffrement mono-alphabétique.

   Vous spécifierez bien votre programme et vérifierez son comportement sur les exemples de l'exercice 4 de la première feuille.

Exercice 3 – Cryptanalyse d'un chiffrement mono-alphabétique

1. (**Cryptanalyse à la main**) En utilisant le programme `cryptanalyse_subst.py` fourni sous Moodle, cryptanalyser à la main le texte fourni `mono.chiffre`.

   Le programme `cryptanalyse_subst.py` permet interactivement de cryptanalyser un chiffrement mono-alphabétique en calculant pour l'utilisateur les fréquences des caractères et des bigrammes du texte, celles de la langue pour comparaison et la possibilité de substituer les correspondances petit à petit : le texte chiffré est écrit en majuscules, on transforme le chiffré majuscule en déchiffré minuscule. Les explications sont fournies au fur et à mesure du déroulement. Pour faciliter la cryptanalyse cette version utilise la fréquence des bigrammes, c'est-à-dire de couples de lettres dans le texte à cryptanalyser et dans le texte de référence de la langue.

2. (**Cryptanalyse itérative**) Comme vous avez certainement pu le constater précédemment, la cryptanalyse à la main est assez pénible, surtout au commencement et sans entraînement. Nous vous proposons ici de programmer une méthode itérative simple pour dégrossir l'essentiel du travail.

   Pour cela nous allons associer à un texte une quantité e qui sera la somme des logarithmes du nombre d'occurrences des tétragrammes (groupes de 4 caractères consécutifs dans un texte) qui y apparaissent. Par exemple pour le mot `CRYPTO`,

$$e(\texttt{CRYPTO}) = \log(e(\texttt{CRYP})) + \log(e(\texttt{RYPT})) + \log(e(\texttt{YPTO}))$$

et $e(<\textit{tétragramme}>)$ est lue dans une table à partir d'un fichier contenant le nombre d'occurrences des tétragrammes d'une langue, calculé à partir d'un corpus de texte significatif.

Lorsqu'on rencontre un tétragramme inconnu, on lui associe un nombre d'occurrences très faible mais non nul (puisque log n'est pas défini en 0) tel que `0.001`.

Le principe est d'appliquer une substitution à un texte, de calculer la quantité e associée à ce nouveau texte et si cette quantité a augmenté on essaie d'échanger aléatoirement deux caractères dans la substitution pour voir si on peut faire encore mieux, sinon on fait la même chose à partir de la substitution précédente. On s'arrête soit parce que le nombre d'étapes fixé à l'avance est atteint, soit parce qu'on fait du sur place depuis trop longtemps.

Le fichier du nombre d'occurrences des tétragrammes dans un corpus français significatif `nb_tetra_fr.csv` est fourni sur Moodle.