L. DIDIER & R. CABANE

Déchiffrement d'Al Kindi

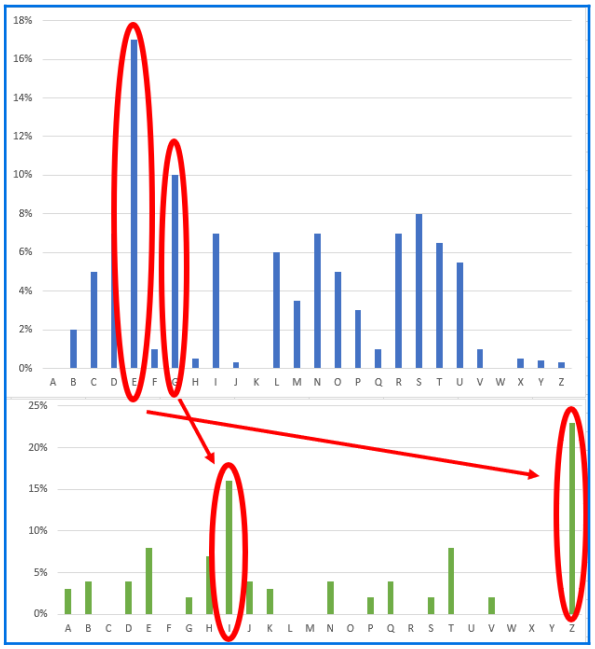
Présentation

Dans le programme (spécialité Première)

Exemples d'algorithmes	**Capacités attendues**
Fréquence d'apparition des lettres d'un texte donné, en français, en anglais. **Notion de liste** La génération des listes en compréhension et en extension est mise en lien avec la notion d'ensemble. Les conditions apparaissant dans les listes définies en compréhension permettent de travailler la logique.	Générer une liste (en extension, par ajouts successifs ou en compréhension). Manipuler des éléments d'une liste (ajouter, supprimer...) et leurs indices. Parcourir une liste. Itérer sur les éléments d'une liste.

Situation déclenchante

Al-Kindi[1], philosophe et savant arabe du IX^{e} siècle, traduisit et adapta de nombreux ouvrages grecs, et fut le premier auteur à avoir écrit un traité de cryptographie. C'est à Al-Kindi que nous devons l'invention du chiffrement « monoalphabétique » qui consiste à remplacer chaque lettre[2] d'un texte par une autre lettre, sachant que deux lettres distinctes doivent être chiffrées par deux lettres distinctes pour permettre un déchiffrement du message sans ambiguïté. Comment décoder un message codé de cette manière sans en connaître le codage ?



Une façon d'attaquer un chiffrement par substitution mono-alphabétique est l'analyse fréquentielle si le texte à décoder est assez long. On compare la fréquence d'apparition de chaque caractère dans le texte codé avec la fréquence moyenne des lettres dans la langue de référence. On peut ainsi établir une première correspondance.

Les esprits curieux pourront aussi s'intéresser au concours de décryptage Al-Kindi[3].

But à atteindre

Écrire un script Python permettant de trouver la fréquence d'apparition des lettres de l'alphabet dans un texte donné.

1 https://fr.wikipedia.org/wiki/Al-Kindi
2 Pour simplifier nous ne traiterons ici que des 26 lettres minuscules, ignorant tout autre lettre ou symbole.
3 https://concours-alkindi.fr

L. DIDIER & R. CABANE

Fiche méthode

Proposition de résolution

On crée **trois fonctions** dans ce script :

1 Une fonction `freq_lettre` qui prend comme arguments une chaîne de caractères et une lettre. Cette fonction renvoie la fréquence d'apparition de la lettre dans la chaîne de caractères.

2 Une fonction `freq_alphabet` qui prend comme argument une chaîne de caractères et qui renvoie la liste des lettres de l'alphabet accompagné de leurs fréquences d'apparition dans la chaîne de caractères (c'est donc une liste de listes).

3 Une fonction `affichage` qui prend comme argument la liste renvoyée par `freq_alphabet` et qui renvoie une liste composée des listes de lettres si la fréquence d'apparition est non nulle, avec une fréquence arrondie à 3 décimales.

```
PYTHON SHELL
alpha
>>> # Shell Reinitialized
>>> # L'exécution de FREQLETT
>>> from FREQLETT import *
>>> freq_lettre("abracadabra","a
")
0.4545454545454545
```

```
PYTHON SHELL
>>> freq_alphabet("abracadabra")
```

```
909090909092], ['e', 0.0], ['f',
 0.0], ['g', 0.0], ['h', 0.0], [
'i', 0.0], ['j', 0.0], ['k', 0.0
], ['l', 0.0], ['m', 0.0], ['n',
 0.0], ['o', 0.0], ['p', 0.0], [
'q', 0.0], ['r', 0.1818181818181
818], ['s', 0.0], ['t', 0.0], ['
u', 0.0], ['v', 0.0], ['w', 0.0]
, ['x', 0.0], ['y', 0.0], ['z',
0.0]]
```

```
PYTHON SHELL
>>> affichage(freq_alphabet("abr
acadabra"))
[['a', 0.455], ['b', 0.182], ['c
', 0.091], ['d', 0.091], ['r', 0
.182]]
```

Étapes de résolution

1. Pour `freq_lettre`, on commence par chercher le nombre de caractères à traiter : l'instruction `len(texte)` permet justement de déterminer la longueur de la chaîne de caractères (ici nommée `texte`). La boucle « `for` » permet ensuite de compter le nombre d'apparitions de la lettre dans le texte.

2. Pour la fonction `freq_alphabet`, on utilise plusieurs fois la fonction précédente.

```
ÉDITEUR : FREQLETT
LIGNE DU SCRIPT 0018
def freq_lettre(texte,lettre):
    nbcar=len(texte)
    compteur=0
    for k in range(nbcar):
        if texte[k]==lettre:
            compteur=compteur+1
    return compteur/nbcar
```

```
ÉDITEUR : FREQLETT
LIGNE DU SCRIPT 0008
def freq_alphabet(texte):
    alphabet="abcdefghijklmnopqrst
        uvwxyz"
    frequences=[]
    for lettre in alphabet:
        frequences.append([lettre,fr
            eq_lettre(texte,lettre)])
    return frequences
```

Un principe à retenir : on peut appeler une fonction (ici, la fonction `freq_lettre`) à l'intérieur d'une autre fonction (ici, `freq_alphabet`). Voir l'appendice 1 à ce sujet.

3. On construit la liste appelée `frequences`, initialisée avec une liste vide. L'instruction `frequences.append([lettre,freq_lettre(texte,lettre)])` permet à chaque passage de boucle d'ajouter une lettre de l'alphabet accompagnée de son pourcentage d'apparition dans le texte en faisant appel à la fonction `freq_lettre`.

La fonction `affichage` permet d'obtenir des résultats plus lisibles en n'affichant que les fréquences non nulles et arrondies au millième.

```
def affichage(l):
    aff=[]
    for i in range (0,25):
        if l[i][1]!=0:
            l[i][1]=round(l[i][1],3)
            aff.append(l[i])
    return aff
```

L'instruction `round(x,3)` permet d'arrondir la valeur `x` à 3 chiffres après la virgule.

On note que la fonction `freq_alphabet` parcourt 26 fois le texte. Une autre approche algorithmique (code ci-contre) permettrait de parcourir une seule fois le texte et donc de gagner en efficacité. Cette approche utilise une boucle du type `for ch in txt`, revenant à faire parcourir à la variable `ch` la chaîne de caractères `txt`.

```
ÉDITEUR : FREQALPH
LIGNE DU SCRIPT 0009
def frqalpha(txt):
    freq=[0 for k in range(26)]
    for ch in txt:
        k=ord(ch)-ord('a')
        if (0<=k<=26):
            freq[k]+=1
    return freq
```

La fonction `ord` prend en paramètre un caractère et renvoie le numéro associé à cette lettre.

Lors de l'exécution de cette fonction, on observe le tableau des effectifs associés à chaque lettre de l'alphabet.

```
PYTHON SHELL
alpha
>>> # Shell Reinitialized
>>> # L'exécution de FREQALPH
>>> from FREQALPH import *
>>> frqalpha("portez ce vieux whisky au juge blond qui fume" )
[1, 1, 1, 1, 5, 1, 1, 1, 3, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 5, 1, 1, 1, 1, 1]
>>>
Fns… | a A # | Outils | éditer | Script
```

Complément : il est possible d'analyser des textes plus longs, sur plusieurs lignes.

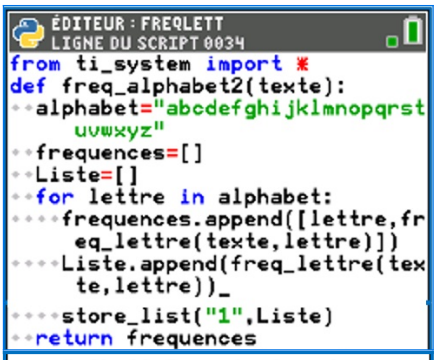
Pour ce faire, on doit insérer le texte entre des triples guillemets (voir appendice 1) :

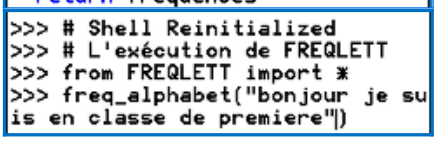
```
s="""Première ligne
Seconde
Troisième"""
print(s)
```

Pour aller plus loin

Approfondissement possible

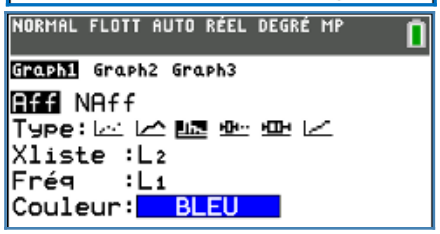
On peut représenter graphiquement les résultats à l'aide d'un histogramme. Pour cela, il faut créer une liste qui va être exportée en dehors de l'application Python.



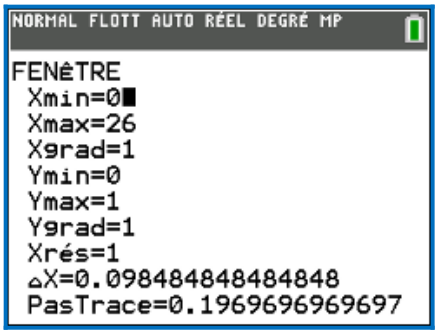


TI-83 Cette opération nécessitera d'importer la bibliothèque `ti_system`. Les fréquences sont stockées dans une liste (Python) appelée `Liste` qui sera exportée au sein du menu *liste* dans la variable (système) $\mathbf{L_1}$ grâce à l'instruction `store_list("1",Liste)` (voir l'appendice 2 pour des détails).

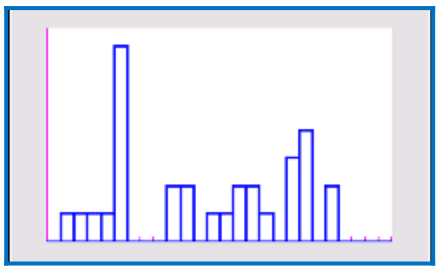
Une fois le programme exécuté, il faut quitter l'application Python et aller dans la rubrique *graph stats* (touches 2nde puis f(x)).



Il faut régler les paramètres d'affichage du graphique statistique (on peut par exemple mettre les numéros de 1 à 26 dans la liste $\mathbf{L_2}$, la liste $\mathbf{L_1}$ contenant les fréquences calculées par le programme).



Ne pas oublier de régler la fenêtre d'affichage (touche **fenêtre**) pour avoir un affichage adapté (exemple ci-contre) !



Prolongements possibles – et un défi

1. Utiliser ce script pour déterminer les fréquences des lettres employées en français (dans un texte de référence).
2. Appliquer à un problème de décodage.

Voici un texte codé :

```
fg ozjdgw v uldgvvg uxa gqvgzyquqwv lgvzduqw wdujuzoogd o uoyrdzwkpztxg
zqfoxvg luqv og sdryduppg lg vsgfzuozwg puwkgpuwztxgv uxvvz ezgq gq fouvvg lg
sdgpzgdg gw lg wgdpzquog gw sdrsrvg lgv uoyrdzwkpgv wdgv judzgv. og fkrza lg
ou fuofxouwdzfg u vrq zpsrdwuqfg.
```

On suppose que le codage utilisé est une substitution mono-alphabétique. Quelle lettre code le « e » ? On pourra utiliser les programmes précédents ainsi que les pourcentages de référence d'apparition des lettres dans la langue française[4]. Et la suite du décodage … est votre défi !

4 https://fr.wikipedia.org/wiki/Fréquence_d'apparition_des_lettres_en_français
https://bibmath.net/crypto/index.php?action=affiche&quoi=chasseur/frequences_francais