

# TP Informatique n° 9

## Statistique : estimation et traitement de données

⚠ Pour la bonne réalisation de ce TP, les fichiers `notes.txt`, `seqADN.py` et `seq.txt` doivent être chargés avant le TP dans le répertoire `\user\home\` de la clé USB. ⚠

Listing 1 – Bibliothèques utiles dans le TP

```

1 import numpy as np # pour les tableaux
2 import math
3 import matplotlib.pyplot as plt # pour les graphiques
4 import random as rd
5 import scipy.stats as sps # pour la simulation aléatoire
6 import os

```

## 1 Manipulation de fichiers et statistiques élémentaires

Le fichier `notes.txt` contient des notes, à raison d'une par ligne. Utiliser le script suivant pour lire les données dans le fichier et créer une liste contenant les notes utilisable avec Python. (On suppose que les fichiers sont dans le répertoire `\user\home\` de la clé USB.)

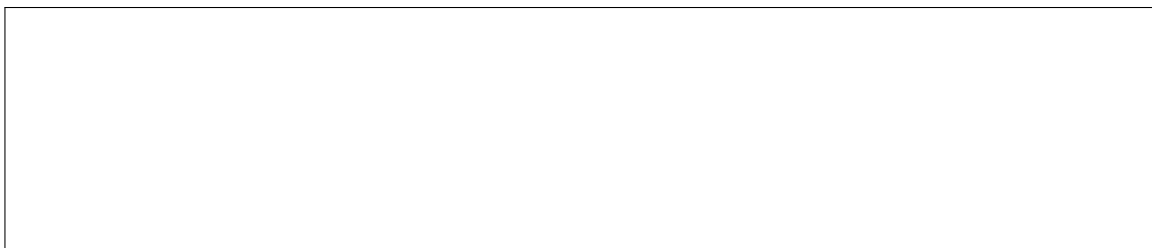
Listing 2 – Lecture de données dans un fichier

```

1 os.getcwd() # Affiche le répertoire courant
2 os.chdir("\user\home\") #Change le répertoire courant
3 os.listdir() # Liste le contenu du répertoire courant
4
5 #crée une liste vide
6 notes=[]
7
8 # ouvre en lecture le fichier contenant les notes
9 f=open('notes.txt','r')
10
11 #pour chaque ligne (note)
12 for n in f:
13     #ou ajoute la note à la liste
14     #on doit explicitement convertir en type float, car n est de type string
15     notes.append(float(n))
16
17 f.close() #on n'oublie pas de fermer le fichier en fin de lecture

```

■ **Exercice 1** Vérifier que la récupération des données s'est bien passée, puis calculer la moyenne et l'écart-type de cette série statistique, et la représenter graphiquement à l'aide d'un histogramme. ■



■ **Exercice 2** Effectuer 1000 simulations indépendantes selon la loi normale  $\mathcal{N}(m, \sigma^2)$ . ⚠ Choisir en secret des valeurs pour  $m$  et  $\sigma^2$ !

Enregistrer les valeurs numériques dans un fichier `prenom.txt` à raison d'une valeur par ligne. ■



⚠ Dans la suite du TP, le but est de deviner les valeurs **inconnues** de  $m$  et  $\sigma^2$  choisies par la voisine. (Pour cela, on échange les fichiers, les clés USB ou les places ...)

## 2 Estimation

### 2.1 Notion d'estimateur

**Définition 1**  $X$  étant une variable aléatoire d'espérance  $m$  et de variance  $\sigma^2$ , un  $n$ -échantillon est un  $n$ -uplet  $(X_1, \dots, X_n)$  de variables aléatoires indépendantes et de même loi que  $X$ .

**Remarque 1** Ainsi, dans notre problème, nous avons  $n = 1000$  et on suppose que les observations (valeurs numériques)  $(x_1, \dots, x_n)$  proviennent de variables aléatoires  $(X_1, \dots, X_n)$  indépendantes et de même loi.

**Définition 2** Un estimateur d'un paramètre  $\theta$  inconnu (par exemple, l'espérance  $m$  ou la variance  $\sigma^2$  de la loi de  $X$ , mais aussi la médiane, ou toute autre valeur caractéristique d'une loi de probabilité ...) est une suite de variables aléatoires  $(T_n)$ , chaque  $T_n$  étant fonction de  $(X_1, \dots, X_n)$  :

$$\forall n \in \mathbb{N}^*, \quad T_n = \varphi_n(X_1, \dots, X_n)$$

**Remarque 2**

1. La loi exponentielle  $\mathcal{E}(\lambda)$  est paramétrée naturellement par l'inverse de son espérance, il revient donc au même sur le plan pratique d'avoir un estimateur de l'une ou l'autre de ces quantités.
2. La loi uniforme  $\mathcal{U}[a, b]$  dépend de deux paramètres : les extrémités du segment  $[a, b]$ .
3. Chaque variable aléatoire  $T_n$  apporte de l'information sur le paramètre inconnu.
4. La valeur  $t_n = T_n(\omega)$  obtenue à partir de l'observation d'un échantillon est l'estimation du paramètre.

**Proposition 1** 1. On appelle moyenne empirique, et on note  $M_n$  (ou  $\bar{X}_n$ ) l'estimateur de l'espérance  $m$  défini par :

$$\forall n \in \mathbb{N}^*, \quad M_n = \frac{1}{n} \sum_{i=1}^n X_i$$

Il vérifie  $\mathbf{E}(M_n) = m$  et  $\mathbf{V}(M_n) = \frac{\sigma^2}{n}$ .

2. On appelle variance empirique, et on note  $S_n^2$  l'estimateur de la variance  $\sigma^2$  défini par :

$$\forall n \in \mathbb{N}^*, \quad S_n^2 = \frac{1}{n} \sum_{i=1}^n (X_i - M_n)^2 = \frac{1}{n} \left( \sum_{i=1}^n X_i^2 \right) - M_n^2$$

■ **Exercice 3** Calculer les estimations  $m_*$  et  $\sigma_*^2$  de l'espérance et de la variance de la loi de  $X$  à partir de votre échantillon. ■



## 2.2 Notion d'intervalle de confiance

On s'intéresse désormais à la précision de l'estimation, et donc à l'écart entre la valeur calculée à partir des observations et la valeur **inconnue** du paramètre.

Pour cela, on admet le théorème suivant :

**Théorème 1** *Théorème limite central avec estimation de l'écart-type*

Soit  $(X_n)$  une suite de variables aléatoires mutuellement indépendantes de même loi, admettant une espérance  $\mu$  et une variance  $\sigma^2$ . Alors, pour tous réels  $a$  et  $b$  tels que  $a < b$ ,

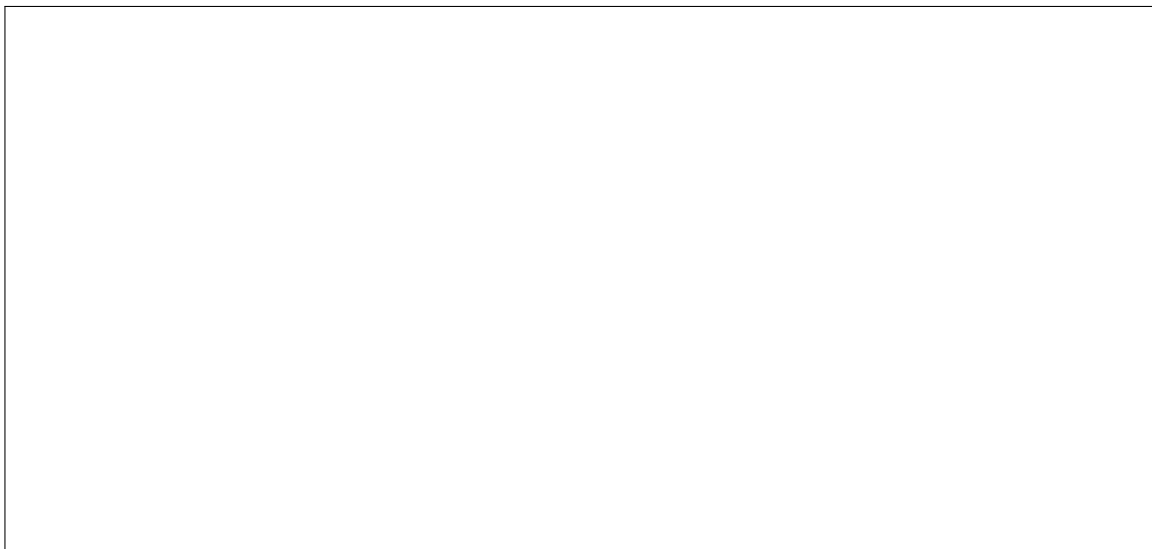
$$\mathbf{P}\left(a \leq \frac{M_n - \mu}{\frac{S_n}{\sqrt{n}}} \leq b\right) \xrightarrow{n \rightarrow +\infty} \frac{1}{\sqrt{2\pi}} \int_a^b e^{-\frac{t^2}{2}} dt$$

**Remarque 3** Quelle est la différence entre ce théorème et le théorème limite central ? Quel est son intérêt ?

■ **Exercice 4** 1. Etant donné un réel  $\alpha \in ]0, 1[$  appelé *risque*, proposer un *intervalle de confiance asymptotique*  $[a_n, b_n]$ , où  $a_n$  et  $b_n$  ne dépendent que de  $(X_1, \dots, X_n)$ , tel que

$$\lim_{n \rightarrow +\infty} \mathbf{P}(\mu \in [a_n, b_n]) = 1 - \alpha$$

2. Ecrire une fonction qui prend en paramètre une liste d'observations  $L$  et un réel **alpha** et qui renvoie un intervalle de confiance asymptotique pour l'espérance  $m$  au niveau de risque  $\alpha$ . On pourra s'aider de la fonction `sps.norm.ppf` pour déterminer le quantile d'ordre  $\alpha$  de la loi  $\mathcal{N}(0, 1)$ . ■



■ **Exercice 5** Superposer l'histogramme des données avec la densité usuelle de la loi  $\mathcal{N}(m_*, \sigma_*^2)$ . Quel est le pourcentage de valeurs appartenant à l'intervalle  $[m_* - 1, 96\sigma_*, m_* + 1, 96\sigma_*]$  ? ■



### 3 Statistiques sur une séquence d'ADN

On souhaite décoder les séquences d'ADN présentes dans le fichier `seq.txt` sous la forme de chaînes de caractères. Pour cela, on va utiliser les fonctions présentes dans le fichier `seqADN.py`.

Listing 3 – Décodage d'une séquence d'ADN

```

1  # determine si une sequence est correcte
2  def seqADNCorrecte(seq):
3      for cara in seq:
4          if cara not in 'actg':
5              return False
6      return True
7
8  #affiche la fréquences des bases sur une sequence
9  def afficherStatBaseADN(seq):
10     nbr=len(seq)
11     nbrA=0
12     nbrC=0
13     nbrT=0
14     nbrG=0
15
16     for cara in seq:
17         if cara=='a':
18             nbrA+=1
19         if cara=='c':
20             nbrC+=1
21         if cara=='t':
22             nbrT+=1
23         if cara=='g':
24             nbrG+=1
25
26     print("a:",nbrA/nbr,"%")
27     print("c:",nbrC/nbr,"%")
28     print("t:",nbrT/nbr,"%")
29     print("g:",nbrG/nbr,"%")
30
31  def traduction(seq):
32     prot=''
33     i=0
34     #pour chaque codon
35     while i<len(seq):
36         #on calcul l'acide aminé correspondant et on l'ajoute à la protéine
37         prot+=code[seq[i:i+3]]
38         i+=3
39     return prot
40
41  f=open("seq.txt","r")
42  for s in f:
43     #enleve le retour à la ligne, si il est présent
44     if(s[len(s)-1]=='\n'):
45         s=s[:len(s)-1]
46     print("Sequence :")

```

```
47     print(s)
48     if seqADNCorrecte(s) :
49         afficherStatBaseADN(s)
50         prot=traduction(s)
51         print(prot)
52     else:
53         print("Sequence incorrecte")
54
55 f.close()
```

---

■ **Exercice 6** Tester les fonctions une à une avec la courte séquence d'essai `seq`. Décrire en commentaire ce que fait chaque fonction.

Quel est le type de la variable `code`? Expliquer son intérêt. Comment fonctionne la traduction de la séquence d'ADN en protéine? ■