

On importera pour ce TP les modules `numpy`, `random` et `matplotlib`.

## 1 Loi des grands nombres

**Question 1.** Écrire une fonction qui simule une variable aléatoire  $X$  suivant la loi  $\mathcal{E}(1)$ .

Rappeler l'espérance et la variance de  $X$  : .....

**Question 2.** Écrire une fonction qui prend en argument un entier  $n$  non nul, qui simule  $n$  variables aléatoires indépendantes  $X_1, \dots, X_n$  suivant la loi  $\mathcal{E}(1)$  et qui renvoie la moyenne empirique  $M_n$  et la variance empirique  $S_n^2$  de l'échantillon  $(X_1, \dots, X_n)$ . On rappelle que :

$$M_n = \frac{1}{n} \sum_{k=1}^n X_k \text{ et } S_n^2 = \frac{1}{n} \sum_{k=1}^n (X_k - M_n)^2 = \frac{1}{n} \left( \sum_{k=1}^n X_k^2 \right) - M_n^2.$$

**Question 3.** Représenter graphiquement une simulation de  $M_n$  en fonction de  $n$  pour tout  $n \in \llbracket 1, 2000 \rrbracket$ . Commenter l'allure du graphique.

.....  
 .....

## 2 Théorème central limite

**Question 4.** Écrire une fonction `mat_alea` qui renvoie une matrice  $\mathbf{M}$  de taille  $30 \times 10000$  dont les coefficients sont la réalisation de  $30 \times 10000$  variables aléatoires indépendantes suivant la loi exponentielle  $\mathcal{E}(1)$ .

**Question 5.** Écrire une fonction `moyennes` qui prend en argument un entier  $n$  non nul et une matrice `mat` (telle que renvoyée par la fonction `mat_alea`). La fonction devra renvoyer une matrice ligne dont chaque coefficient est la moyenne des coefficients des  $n$  premières lignes de `mat` de la même colonne.

*On pourra utiliser `np.sum(M, 0)` qui renvoie la matrice des sommes des coefficients de chaque colonne de  $M$ .*

**Question 6.** Exécuter le code ci-dessous et expliquer ce qu'il fait. Que constate-t-on ?

---

```
N=30
mat = mat_alea()
for n in range(1, N+1):
    plt.clf()
    plt.grid()
    plt.axis([0, 2, 0, 3])
    plt.hist(moyenne(n, mat), density=True, bins=80, label="n="+str(n))
    plt.legend(loc="upper right")
    plt.pause(.2)
plt.show()
```

---

La fonction `hist`, du module `matplotlib.pyplot`, permet d'afficher un histogramme des valeurs de la liste ou l'array en argument. Lorsque l'argument `bins` est un entier, il correspond au nombre de rectangles. L'option `density=True` assure que la somme totale des aires des rectangle de l'histogramme est égale à 1.

.....  
 .....  
 .....

**Question 7.** Compléter le code ci-dessous afin d'illustrer le théorème central limite.

---

```
N=30
mu, sigma = 1, 1
mat = ...
for n in range(1,N+1):
    plt.clf()
    plt.grid()
    plt.axis([-5,5,0,0.7])
    Mnetoiles = ...
    plt.hist(..., density=True, bins=80, label=...)
    x = np.linspace(-4,4,100)
    densiteN01=np.exp(-x**2/2)/np.sqrt(2*np.pi)
    plt.plot(x,densiteN01,color="red",label=...)
    plt.legend(loc="upper right")
    plt.pause(...)
plt.show()
```

---

**Question 8.** En vertu du théorème central limite, par quelle loi peut-on approcher la loi de  $M_n$  ?  
Reprendre le code de la question 6 afin d'illustrer ce résultat.