

1BCPST2 Représentation graphique d'une loi par simulations

On suppose qu'une variable aléatoire X est simulée par une fonction python *simulX* et que $X(\Omega) \subset \llbracket 0, n \rrbracket$.

Par exemple, si X suit la loi binomiale de paramètres $10, \frac{1}{2}$ alors *simulX* pourra être définie de la façon suivante :

```
import random as rd
def simulX():
    cpt=0
    for k in range(10):
        if rd.random()<1/2:
            cpt+=1
    return cpt
```

Dans ce cas on choisira $n = 10$.

1 Création d'un tableau de fréquence d'une variable aléatoire X

Le code suivant crée un tableau de fréquence de X avec r simulations. Ce tableau représente la loi empirique de X .

```
tab = [0]*(n+1)
for _ in range(r):
    i = simulX()
    tab[i] += 1/r # construction du tableau de fréquences
```

Ce script se retrouvera à l'intérieur des fonctions suivantes.

2 Diagramme en barre de la loi empirique de X

```
import matplotlib.pyplot as plt
def diag(r):
    val = [k for k in range(n+1)]
    freq = [0]*(n+1)
    for _ in range(r):
        i = simulX()
        freq[i] += 1/r # construction du tableau de fréquences
    plt.bar(val,freq,label="Loi empirique de X")
    plt.legend()
    plt.show()
```

3 Courbe de la fonction de répartition empirique de X

```
def repart(r):
    f = [0]*(n+1)
    for j in range(r):
        i = simulX()
        f[i] += 1/r # construction du tableau de fréquences
    for k in range(1,n+1):
        f[k] += f[k-1] # construction du tableau de fréquences cumulées
    plt.plot([-1,0],[0,0],"black",label="Fonction de répartition empirique de X")
    for k in range(n+1):
        plt.plot([k,k+1],[f[k],f[k]],"black")
    plt.legend()
    plt.show()
```

4 Options graphiques

Pour améliorer la lisibilité de la représentation graphique on peut adjoindre dans la figure courante des objets graphiques obtenus par des fonctions du module *matplotlib.pyplot* que l'on va décrire brièvement.

Commandes	Actions
<code>plt.bar(X,Y,label="L",width=a)</code>	Crée un diagramme en barre de longueur Y au-dessus de X avec la légende L (facultatif) et une largeur de barre égale à a (facultatif)
<code>plt.plot(X,Y,label="L",'+-r')</code>	Génère la courbe des points définis par les listes X et Y (abscisses et ordonnées) avec la légende L (facultatif) et les options : <ul style="list-style-type: none">• symbole : '.' point, 'o' rond, 'h' hexagone, '+' plus, 'x' croix, '*' étoile, ...• ligne : '-' trait plein, '- -' pointillé, '-.' alterné, ...• couleur : 'b' bleu, 'r' rouge, 'g' vert, 'c' cyan, 'm' magenta, 'k' noir, ...
<code>plt.legend()</code>	Fait ressortir les labels dans la figure courante
<code>plt.grid()</code>	Crée un quadrillage
<code>plt.xlim(xmin,xmax)</code>	Fixe les bornes de l'axe des abscisses
<code>plt.ylim(ymin,ymax)</code>	Fixe les bornes de l'axe des ordonnées

Les commandes suivantes sont moins importantes mais peuvent faire ressortir encore plus d'informations

Commandes	Actions
<code>plt.figure("Exercice 1 question 1")</code>	Crée la figure nommée <i>Exercice 1 question 1</i> qui accueillera les courbes
<code>plt.title("Titre de la figure")</code>	Crée le titre <i>Titre de la figure</i> pour la figure courante
<code>plt.xlabel("Titre axe des x")</code>	Crée le titre <i>Titre axe des x</i> pour l'axe des abscisses de la figure courante
<code>plt.ylabel("Titre axe des y")</code>	Crée le titre <i>Titre axe des y</i> pour l'axe des ordonnées de la figure courante
<code>plt.text(x,y,"Insertion")</code>	Insère le texte <i>Insertion</i> au point de coordonnées (x,y) de la figure courante
<code>plt.subplot(n,p,i)</code>	Crée une matrice de sous-figures à n lignes et p colonnes et choisit la <i>i</i> ^{ème}