Simulation de variables aléatoires discrètes

1 La commande `plt.bar`

- ▷ Il faut commencer par charger la bibliothèque en tapant l'instruction suivant : `import matplotlib.pyplot as plt`
- ▷ La commande `plt.bar` prend a minima deux arguments que l'on appelle ici `x` et `y`. Le vecteur `x=(`$x_1, \ldots, x_n$`)` correspond aux valeurs en abscisse et le vecteur `y=(`$y_1, \ldots, y_n$`)` correspond aux valeurs en ordonnées. La commande `plt.bar` trace ensuite un diagramme en bâtons. Le bâton i est situé à la valeur x_i et est de hauteur y_i.
- ▷ Voici la liste des 29 notes obtenues par l'ECG1 au CB2 (arrondies à l'unité) :

$$11, 3, 14, 11, 4, 8, 16, 12, 9, 11, 5, 7, 2, 3, 8, 16, 10, 10, 2, 8, 0, 8, 10, 10, 6, 12, 13, 8, 4$$

  Tracer le diagramme en bâtons représentant la répartition de ces notes.

2 Simuation des lois usuelles

Les lois usuelles sont prédéfinies en Python, les commandes sont les suivantes (en ayant préalablement chargé la bibliothèque `numpy.random` avec la commande `import numpy.random as rd`)

- ▷ La commande `rd.randint(n,m)` renvoie un entier compris entre n et $m-1$ selon la loi uniforme.
- ▷ La commande `rd.binomial(n,p)` renvoie un entier compris entre 0 et n suivant la loi binomiale de paramètre n et $p \in [0,1]$. Avec cette commande, on peut également simuler une loi de Bernoulli de paramètre p en tapant `rd.binomial(1,p)`.
- ▷ La commande `rd.geometric(p)` renvoie un entier appartenant à $\mathbb{N}^*$ suivant la loi géométrique de paramètre p.
- ▷ La commande `rd. poisson(L)` renvoie un entier appartenant à $\mathbb{N}$ suivant la loi de Poisson de paramètre L.

Notre objectif est de tracer un diagramme en bâtons donnant la fréquence d'apparition des valeurs prises par une variable aléatoire suivant une loi usuelle et ce pour chaque loi usuelle.

1. Soit X une variable aléatoire suivant une **loi de Bernoulli** de paramètre p donné.
   - ▷ Recopier, compléter les instructions Python suivantes qui :
     - ⋆ définissent une fonction appelée `frequencebern` qui étant donnés `p` $\in [0,1]$ et `nb` $\in \mathbb{N}$, renvoie un vecteur donnant la fréquence d'apparition des valeurs prises par une v.a. suivant une loi de Bernoulli de paramètre `p` lorsque on effectue `nb` simulations de cette loi,
     - ⋆ puis trace le diagramme en bâtons de la répartition des valeurs prises par une v.a. suivant une loi de Bernoulli de paramètre $\frac{1}{3}$ simulée 1000 fois.

```
def frequencebern(p,nb):
    S=np.zeros(2)
    for k in range(nb):
        i = ..........
        S[i]=S[i]+1
    return S/nb
x=np.arange(0,2)
y=frequencebern(1/3,1000)
plt.bar(x,y)
```

   - ▷ Exécutez le programme (plusieurs fois). Qu'observez-vous ? Est-ce conforme à vos attentes ? Commentez. Vous pouvez également tester votre programme en changeant les valeurs de `p` et de `nb`.
2. Soit X une variable aléatoire suivant une **loi binomiale** de paramètres n et p donnés.
   - ▷ En s'inspirant des instructions précédentes, écrire une fonction Python appelée `frequencebin` qui étant donnés `p` $\in [0,1]$, `n` $\in \mathbb{N}^*$ et `nb` $\in \mathbb{N}$, renvoie un vecteur donnant la fréquence d'apparition des valeurs prises par une v.a. suivant une loi binomiale de paramètres `n` et `p` lorsque on effectue `nb` simulations de cette loi.

- ▷ Puis écrire des instructions permettant de tracer le diagramme en bâtons de la répartition des valeurs prises par une v.a. suivant une loi binomiale de paramètres 15 et $\frac{1}{3}$ simulée 1000 fois.
- ▷ Exécutez le programme (plusieurs fois). Qu'observez-vous ? Est-ce conforme à vos attentes ? Commentez.

3. Soit X une variable aléatoire suivant une **loi géométrique** de paramètre et p donné.
   - ▷ En s'inspirant des instructions précédentes, écrire une fonction Python appelée `frequencegeo` qui étant donnés `p` $\in [0,1]$ et `nb` $\in \mathbb{N}$, renvoie un vecteur donnant la fréquence d'apparition des valeurs prises par une v.a. suivant une loi géométrique de paramètre `p` lorsque on effectue `nb` simulations de cette loi.
   - ▷ Puis écrire des instructions permettant de tracer le diagramme en bâtons de la répartition des valeurs prises par une v.a. suivant une loi géométrique de paramètre $\frac{1}{4}$ simulée 1000 fois.
   - ▷ Exécutez le programme (plusieurs fois). Qu'observez-vous ? Est-ce conforme à vos attentes ? Commentez.

   - ▷ Faites varier le paramètre p. Qu'observez-vous ?
4. Mêmes questions avec X une variable aléatoire suivant une **loi de Poisson** de paramètre $L > 0$.
5. Mêmes questions avec X une variable aléatoire suivant une **loi uniforme** sur $[\![n, m]\!]$.

3 Lien entre loi binomiale et loi de Poisson

Soient $\lambda \in \mathbb{R}_+^*, n \in \mathbb{N}^*$ et X_n une variable aléatoire suivant une loi binomiale de paramètre $\left(n, \frac{\lambda}{n}\right)$. On a le théorème suivant :

Théorème : Soit $\lambda > 0$. Pour $n \in \mathbb{N}^*$, on considère $X_n \hookrightarrow \mathcal{B}\left(n, \frac{\lambda}{n}\right)$. Alors pour tout k dans $\mathbb{N}$, on a :

$$\lim_{n \to +\infty} P(X_n = k) = P(X = k)\,,$$

où X est une variable aléatoire suivant une loi de Poisson de paramètre λ.

1. **Observation graphique du résultat**
   (a) Créer un vecteur ligne $U = (U_1, U_2, \dots, U_{10000})$ contenant un échantillon de 10000 simulations de la loi de Poisson de paramètre 1 puis tracer le diagramme en bâton des fréquences obtenues.
   (b) Écrire une procédure qui, étant donné un entier n, crée un vecteur ligne $U = (U_1, U_2, \dots, U_{10000})$ contenant un échantillon de 10000 simulations de la loi binomiale de paramètres n et $\frac{1}{n}$ et trace le diagramme en bâton des fréquences obtenues. Tester pour de grandes valeurs de n et comparer avec le résultat obtenu à la question précédente.
2. **Preuve du théorème**
   Le but de cette question est de démontrer le théorème énoncé ci-dessus.
   (a) Montrer que pour tout $k \in [\![0, n]\!]$:
   $$P\left(X_n = k\right) = \frac{\lambda^k}{k!} \frac{n(n-1)\dots(n-k+1)}{n^k} \left(1 - \frac{\lambda}{n}\right)^{n-k}.$$
   (b) Montrer que :
   $$\lim_{n \to +\infty} \frac{n(n-1)\dots(n-k+1)}{n^k} = 1 \text{ et } \lim_{n \to +\infty} \left(1 - \frac{\lambda}{n}\right)^{n-k} = e^{-\lambda}$$
   On pourra utiliser pour la deuxième limite que $\lim_{x \to 0} \frac{\ln(1+x)}{x} = 1$.
   (c) En déduire que :
   $$\lim_{n \to +\infty} P\left(X_n = k\right) = \frac{\lambda^k}{k!} e^{-\lambda}$$
3. **Application :** En pratique, si $n \geq 30$, si $p \leq 0,1$ et si $np \leq 5$, alors la loi $\mathcal{B}(n, p)$ est proche de la loi $\mathcal{P}(np)$.

   Exercice : Suite à une vaccination contre le paludisme, dans une population à risque, on estime à 2%, compte tenu du délai d'immunisation, la proportion de personnes qui seront pourtant atteintes de la maladie.
   Quelle est la probabilité de constater, lors d'un contrôle d'un village de 100 habitants tous récemment vaccinés, plus d'une personne malade ?