TRAVAUX PRATIQUES n° 13
Variables aléatoires

Dans le module `random`, se trouvent deux fonctions indispensables pour simuler des phénomènes aléatoires avec Python : la fonction `random` et la fonction `randint`.

Écrivez dans le shell les lignes suivantes :

```
>>> from random import random, randint
>>> random()
>>> random()
>>> randint(1,6)
>>> randint(10,20)
```

L'appel de la fonction `random()` renvoie un nombre aléatoire U réel compris entre 0 et 1 : ce nombre aléatoire suit la loi uniforme sur l'intervalle $[0,1]$ (vous verrez cette loi l'an prochain). Elle a la propriété que pour tout intervalle $I \subset [0,1]$, $P(U \in I) = \text{longueur}(I)$.

L'appel de la fonction `randint(n,m)` renvoie un nombre entier aléatoire entre n et m compris, c'est-à-dire un nombre qui suit la loi uniforme sur $[\![n,m]\!]$.

Exercice 1 :

1) Écrire, à partir **uniquement** de la fonction `random`, votre propre fonction `randint` que vous appellerez `monrandint`.

   Pour cela, on découpera l'intervalle $[0,1]$ *en* $m-n+1$ *intervalles de même longueur.*

   Rappel : la fonction partie entière se nomme `floor` *et se trouve dans le module* `math`.
2) Écrire un script simulant 100000 variables aléatoires le loi uniforme sur $[\![1,6]\!]$ grâce à votre fonction `monrandint` et calculant le nombre de 6 apparus. Le résultat est-il conforme à vos attentes ?

Exercice 2 :

1) Écrire une fonction `bernoulli` prenant un paramètre p et simulant une variable aléatoire de loi de Bernoulli de paramètre p.
2) Écrire une fonction `binomiale` prenant deux paramètres n et p et simulant une variable aléatoire de loi de binomiale de paramètres n et p.
3) Pour chacune des lois précédentes, écrire une fonction simulant k variables aléatoires de cette loi et renvoyant la liste des résultats.
4) Écrire une fonction simulant k variables de loi binomiale de paramètres n et p et renvoyant une liste `L` de taille $n+1$ telle que pour tout i, `L[i]` est le nombre de fois où i est apparu dans la série.

   Par exemple, si $L = [3,0,2]$, cela signifie qu'on a obtenu 3 fois le nombre 0, 0 fois le nombre 1 et 2 fois le nombre 2 (ici $k=5$ et $n=2$).
5) Écrire le script suivant puis exécuter. Interpréter.

```
import matplotlib.pyplot as plt

x = [k for k in range(n+1)]
plt.bar(x,L)
plt.show()
```

Exercice 3 :

1) Écrire une fonction `cumsum` prenant en argument une liste `L` et renvoyant une liste `Lp` telle que pour tout $k \in [\![0,$ `len(L)-1`$]\!]$, `Lp[k] = L[0]+L[1]+...+L[k]`.
2) Écrire une fonction `indice` prenant en argument une liste `L` et un nombre réel `x`$\in [0,1]$, et renvoyant le premier indice `k`$\in [\![0,$ `len(L)-1`$]\!]$ tel que `L[0]+L[1]+...+L[k-1]` $\leqslant$ `x` $<$ `L[0]+L[1]+...+L[k]`.

   On fera attention au cas où `x<L[0]`.
3) Écrire une fonction `simule` qui prend en argument deux listes de nombres réels `valeurs` et `probas` et qui renvoie `valeurs[i]`, où `i = indice(probas,random())`.
4) Expliquer pourquoi la fonction simule sert à simuler une variable aléatoire ayant pour univers image `valeurs` et pour loi `probas`.
5) **Application** : Un joueur mise 10 euros sur un cheval dans une course. Si le cheval joué gagne la course, le joueur gagne 100 euros ; s'il fini 2e, le joueur gagne 50 euros ; s'il fini 3e, le joueur gagne 20 euros ; sinon le joueur ne gagne rien. La course comporte 20 chevaux et on suppose que tous les ordres d'arrivées sont équiprobables.

   a) Écrire une fonction qui simule le gain du joueur.
   b) Écrire une fonction qui simule un nombre fixé de fois le gain du joueur.
   c) Calculer informatiquement une valeur approchée de l'espérance du gain du joueur. Le jeu est-il favorable au joueur ? Comparer le résultat obtenu à l'espérance théorique du joueur.
   d) Calculer informatiquement une valeur approchée de la variance du joueur.

Exercice 4 :

On dispose de deux boîtes d'allumettes : la boîte « pile » et la boîte « face », et d'une pièce équilibrée. Au départ, les deux boîtes contiennent 20 allumettes chacune. On tire à pile ou face, si on obtient pile, on enlève une allumette de la boîte « pile » et si on obtient face, on enlève une allumette de la boîte « face ».

On s'arrête lorsqu'une boîte est vide et on note X le nombre d'allumettes dans la boîte non vide. Ainsi X est une variable aléatoire d'univers image $[\![1, 20]\!]$.

1) Écrire une fonction `simule` qui ne prend aucun argument et qui simule une valeur de X.
2) Écrire une fonction `simuleMFois` qui prend en argument un entier `m`, qui simule `m` variables aléatoires X et qui renvoie une liste `L` de taille 20 telle que pour tout $i \in [\![0, 19]\!]$, `L[i]` est le nombre d'expériences pour lesquelles le résultat a été `X = i+1`.
3) Réaliser 2000 expériences avec la fonction précédente.

   En utilisant uniquement les fonctions de base de python, estimer la valeur empirique de la moyenne de X et son écart-type.
4) En utilisant la fonction `bar`, dessiner un diagramme en bâton des valeurs de `L`.