

---

## Informatique - T.P n°10

### *Variables aléatoires - Episode 1*

---

On a déjà vu que, sans davantage de précision, tout appel de la fonction `rand()` renvoyait un nombre réel au hasard entre 0 et 1 (on dit que la fonction simule la loi uniforme sur  $[0; 1]$ , qui sera introduite avec plus de détails dans le chapitre sur les v.a.r à densité).

**Exercice 1.** (Loi uniforme sur  $[[a; b]]$ )

Écrire, à l'aide de la fonction `rand()` une fonction `unifZ()` prenant en arguments deux entiers relatifs  $a$  et  $b$  ( $a < b$ ) et simulant une loi uniforme sur  $[[a; b]]$ .

**Exercice 2.** Que font les suites d'instructions suivantes?

```
-->x=1:20; y=1+floor(6*rand(x)); bar(x,y)
-->clf; x=1:1000; y=1+floor(6*rand(x)); histplot(0:6,y)
```

**Exercice 3.** (Tirages dans une urne)

On considère une urne  $U$  contenant 3 boules bleues, 4 boules blanches et 5 boules rouges. On effectue  $n$  tirages dans cette urne et on note  $X_i$  la variable aléatoire égale à 1 si la boule tirée au  $i$ -ème coup est bleue, 2 si elle est blanche et 3 si elle est rouge.

- (1) Créer un programme qui demande à l'utilisateur un nombre de tirages  $n$  et simule l'expérience en affichant les valeurs des  $X_i$ . (☞ On pourra utiliser une loi uniforme sur  $[[1; 12]]$ .)
- (2) Déterminer la loi et calculer mathématiquement  $E(X_i)$ .
- (3) Modifier le programme pour qu'il affiche la moyenne des  $X_i$ . Qu'observe-t-on pour  $n$  grand?

**Exercice 4.** (Le lièvre et la tortue)

Un lièvre et une tortue partent d'un point  $O$ , origine d'un axe gradué, et se déplacent uniquement vers la droite. On lance un dé équilibré: si le résultat est inférieur ou égal à 5, la tortue avance d'une unité. Sinon, le lièvre avance de 6 unités. Le gagnant est celui qui arrive le premier au point d'abscisse 6.

- (1) Déterminer mathématiquement la probabilité que la tortue gagne la course.
- (2) Créer une fonction `course()` qui simule l'expérience en renvoyant 1 si la tortue gagne et 0 si c'est le lièvre.
- (3) Améliorer le programme précédent pour qu'il affiche la fréquence des courses gagnées par la tortue lors de la réalisation de  $n$  expériences. Comparer le résultat à celui obtenu à la première question.

**Exercice 5.** (Loi de Bernoulli et loi Binomiale)

- (1) Écrire une fonction `Bern()` qui prend en argument un nombre réel  $p \in ]0; 1[$  et retourne ensuite la valeur prise, lors d'une réalisation, par une variable aléatoire  $X \hookrightarrow \mathcal{B}(1, p)$ .
- (2) Créer à la suite une fonction `Bino()` qui prend en argument un entier naturel  $n \geq 1$ , un réel  $p \in ]0; 1[$  et renvoie la valeur prise par une v.a.  $S \hookrightarrow \mathcal{B}(n, p)$ .
- (3) Ajouter à la suite du programme des instructions pour demander à l'utilisateur  $n$  et  $p$  et effectuer ensuite 10000 réalisations de  $S$  puis afficher ensuite l'histogramme de ces réalisations.

**Exercice 6.** (D'après EDHEC 2000)

Soit  $n \geq 2$ . On lance  $n$  fois une pièce de monnaie, dont la probabilité d'apparition de *Pile* est  $p \in ]0; 1[$ . Pour tout entier naturel  $k \geq 2$ , on dira que le  $k$ -ième lancer est un *changement* s'il amène un résultat différent du  $(k - 1)$ -ième lancer. On note alors  $X_n$  la variable aléatoire égale au nombre de changements survenus au cours des  $n$  lancers.

- (1) Rappeler les instructions permettant d'écrire une fonction `Bern()` permettant de simuler une loi de Bernoulli de paramètre  $p$ .
- (2) Écrire une fonction `CountCh(n,p)` qui calcule et retourne la valeur prise par  $X_n$ .
- (3) Écrire une suite d'instructions permettant de calculer `CountCh(10000,p)` pour  $p$  allant de 0,1 à 0,9 avec un pas de 0,1 et qui affiche ensuite le diagramme en rectangles.

**Exercice 7.** (La ruine du joueur)

Notre ami José se rend au Casino avec  $n$  euros en poche. Il s'installe à la table de roulette américaine. À chaque partie remportée, il gagne un euro, sinon il perd un euro. On suppose que le casino dispose d'une fortune de  $N$  euros. José ne s'arrête que lorsque il est ruiné (ou que le casino est ruiné). Simuler l'évolution de sa fortune. (On écrira une fonction `casino(n,N,p)` renvoyant un vecteur dont les composantes seront les valeurs successives de sa fortune.)