

Corrigé

Code de partage avec Capytale : 1060-1458970

Echauffement

Simuler une partie de bataille.

On modélise le problème en supposant que 2 joueurs (A et B) disposent d'un chacun de 26 cartes. On considère qu'à chaque tour, chaque joueur a une chance sur deux de gagner (on génère un nombre aléatoire, s'il est compris entre 0 et 0,5, on considère que le joueur A a gagné et de fait on augmente d'une unité son nombre de cartes). On rejoue un tour tant que chaque joueur a au moins une carte.

```
import numpy.random as rd # on importe la
    bibliothèque numpy.random
tas_a=26
tas_b=26
c=0 # on crée un compteur pour compter le
    nombre de tours
while (tas_a>0 and tas_b>0) :
    a=rd.randint(1,13)
    b=rd.randint(1,13)
    c=c+1 # pour compter le nombre de tours
    if a>b :
        tas_a=tas_a+1
        tas_b=tas_b-1
    elif b>a:
        tas_a=tas_a-1
        tas_b=tas_b+1
disp(a,b,c)
```

Exercice 1

1. Compléter la fonction Python `tirage` suivante qui permet de simuler le tirage d'une boule dans une urne contenant b boules blanches numérotées de 1 à b , et r boules rouges numérotées de $b+1$ à $b+r$, et qui renvoie la valeur 1 lorsque la boule piochée est blanche, et renvoie la valeur 0 sinon.

```
import numpy.random as rd

def tirage(b,r):
    n=rd.randint(1,b+r) # on tire un numéro entre 1 et b+r
    if n <= b : # si le numéro tiré est inférieur ou égal à r
        return 1 # on considère qu'il s'agit d'une boule blanche
    else :
        return 0 # sinon qu'il s'agit d'une boule rouge
```

2. Ecrire une fonction Python qui donne le nombre de boules blanches obtenues au cours de n tirages successifs (avec remise) dans une urne contenant b boules blanches et r boules rouges (on devra utiliser la fonction `tirage` de la question précédente.)

```
def NbBlanche(b,r,n) :
    c=0 # on crée un
    compteur
    for i in range(0,n):
        c=c+tirage(b,r)
    return c/n # pour avoir
    la fréquence
```

Avec une boucle `for`, on répète n fois le tirage à l'aide de la fonction précédente.

On fait le choix ici de renvoyer la fréquence (plutôt que le nombre de tirages donnant une boule blanche). Ensuite, il suffit d'utiliser la fonction avec des commandes comme `NbBlanche(10,10,100)` ou `NbBlanche(10,10,10000)`. Plus on augmente le nombre de tirages, plus la fréquence doit s'approcher de la probabilité théorique qui vaut 0,5 avec ces exemples.

3. On souhaite évaluer la fréquence d'apparition d'une boule blanche, lors de 1 000 tirages successifs, effectués (avec remise) dans une urne contenant 4 boules blanches et 6 boules rouges.

- (a) Faire le calcul grâce à la fonction précédente. Refaire le même calcul mais pour 10 000 tirages. Interpréter les résultats.

En exploitant la fonction précédente, il suffit de lancer la commande `NbBlanche(4,6,1000)` puis `NbBlanche(4,6,10 000)`.

Au bout d'un grand nombre de tirages, la fréquence d'apparition d'une boule blanche doit s'approcher de la probabilité : 0,4.

- (b) Compléter la fonction Python suivante qui permet elle aussi d'obtenir cette fréquence.

```
def frequencebl(n):  
    s=0  
    for i in range(1,n+1):  
        if tirage(4,6)>0 :  
            s=s+1  
    return s/n
```

De la même manière que plus haut, on répète le tirage n fois et on calcule le nombre de boules blanches, la fréquence est alors le rapport $\frac{s}{n}$.