TP # 21

Simulation d'expériences aléatoires

Bibliothèque numpy.random

```
import numpy.random as rd
```

Premier succès

Exercice 1
On réalise une succession d'épreuves de Benoulli indépendantes de paramètre $p > 0$ jusqu'à obtenir un succès. Proposer une fonction qui renvoie le nombre d'épreuves réalisées.

```
def premier_succes(p) :
```

Tester 5 appels de la fonction avec $p = 0.1$ et noter les résultats obtenus :

....

Tester 5 appels de la fonction avec $p = 0.01$ et noter les résultats obtenus :

....

Tester 5 appels de la fonction avec $p = 0.001$ et noter les résultats obtenus :

....

Constater que même pour un p très petit (du moment que $p > 0$), le premier succès finit presque-sûrement par arriver.

Jeu d'argent avec arrêt aléatoire

Exercice 2
Un joueur avec un capital de départ de N euros, joue à un jeu composé d'une succession de parties, selon les modalités suivantes :
• Avant chaque partie (y compris la première), le joueur a une probabilité $\alpha \in]0,1[$ de s'arrêter, et quitte alors définitivement le jeu.
• Si le joueur n'a pas choisi de s'arrêter, il joue une partie.
Il gagne alors 1 euro avec probabilité $p \in]0,1[$ et perd 1 euro avec probablité $1 - p$.
Une nouvelle partie est ensuite proposée.

1. Proposer une fonction qui prend en entrée les valeurs de $N \in \mathbb{N}$, $\alpha \in]0,1[$, $p \in]0,1[$, simule la réalisation du jeu, et renvoie le capital du joueur à l'issue du jeu (quand il choisit de s'arrêter).

```
def jeu(N,alpha,p) :
```

2. On choisit, dans la suite, $N = 10$ et $\alpha = 0,01$.
Noter 5 résultats obtenus par la fonction jeu :

• Pour $p = 0,9$: ....

• Pour $p = 0,1$: ....

Constater la différence de capital final en fonction de la valeur de p.

3. On souhaite estimer l'espérance du capital final en faisant la moyenne des résultats obtenus sur 10^4 simulations. Compléter pour cela le programme suivant :

```
N = 10; alpha = 0.01; p = 0.9
S = ...
for k in range( ....... ) :

    S = S + ...............

print( .............. )
```

• Capital final moyen pour $p = 0,9 \simeq$ ....
• Capital final moyen pour $p = 0,1 \simeq$ ....

4. Améliorer la fonction `jeu` précédente pour que le jeu s'arrête automatiquement si le capital du joueur atteint 0. On considère alors que le joueur est ruiné.

```
def jeu(N,alpha,p) :
```

5. On souhaite estimer la probabilité que le joueur finisse ruiné à la fin du jeu, en calculant la proportion de joueurs ruinés sur 10^4 simulations.
Compléter pour cela le programme suivant :

```
N = 10; alpha = 0.01; p = 0.9

compteur = ...
for k in range( ....... ) :

    if ........................

        compteur = compteur + 1

print( ............... )
```

- Probabilité de ruine pour $p = 0,9 \simeq$ ....
- Probabilité de ruine pour $p = 0,7 \simeq$ ....
- Probabilité de ruine pour $p = 0,5 \simeq$ ....
- Probabilité de ruine pour $p = 0,3 \simeq$ ....
- Probabilité de ruine pour $p = 0,1 \simeq$ ....
- Probabilité de ruine pour $p = 0,01 \simeq$ ....

Expliquer pourquoi cette progression de la probabilité de ruine en fonction de p est cohérente :

6. On fixe maintenant $\alpha = 0,01$ et $p = 0,5$. On fait varier le capital de départ N.

- Probabilité de ruine pour $N = 10 \simeq$ ....
- Probabilité de ruine pour $N = 5 \simeq$ ....
- Probabilité de ruine pour $N = 3 \simeq$ ....
- Probabilité de ruine pour $N = 1 \simeq$ ....

Expliquer pourquoi cette progression de la probabilité de ruine en fonction de N est cohérente :