

1 Le Loto

Le jeu du Loto, dans sa forme la plus basique, consiste à choisir indépendamment :

- 5 numéros distincts non ordonnés dans une grille numérotée de 1 à 49,
- un numéro chance parmi les nombres 1 à 10.

Pour remporter le jackpot, il faut que ces 6 numéros sortent lors du tirage aléatoire effectué par la FDJ[®].

On modélise alors l'expérience par

- une suite de 5 tirages sans remise d'une boule dans une urne U_1 contenant 49 boules numérotées de 1 à 49. On note X_k la variable aléatoire égale au numéro tiré lors du k^{e} tirage.
- un tirage indépendant d'une boule dans une urne U_2 contenant 10 boules numérotées de 1 à 10. On note Y la variable aléatoire égale au numéro obtenu lors de ce tirage.

On note A l'événement : « la grille de Loto est gagnante ».

1. La commande `rd.choice(L,d,replace=False)` simule un d -tirage équiprobable sans remise sur l'ensemble L où L est un tableau ou une liste (le paramètre d est optionnel). Cette commande n'étant pas au programme, nous allons la recoder :

```

1  def Tirage_sans_remise(L,n):
2      U,R = L, []
3      for k in range(n):
4          i = rd.randint(0,len(U)) # choix d'un élément de la liste U
5          _____ # ajout de l'élément à la liste R
6          _____ # suppression de l'élément de la liste U
7      return R

```

2. (a) Ecrire une fonction **Python**, nommée `Approx_croissante`, qui renvoie une approximation de la probabilité de l'événement « les 5 numéros ont été tirés dans l'ordre croissant ».
- (b) Ecrire une fonction **Python**, nommée `Approx_grands_numeros`, qui renvoie une approximation de la probabilité de l'événement « les 5 numéros tirés sont supérieurs ou égaux à 40 ».
3. (a) Ecrire une fonction **Python**, nommée `Calc_binom`, qui prend en paramètres deux entiers naturels n et k et qui renvoie le coefficient binomial $\binom{n}{k}$ si $k \leq n$, et 0 sinon. On écrira $\binom{n}{k}$ comme un produit de k termes.
- (b) A l'aide de **Python**, calculer le nombre de grilles de Loto possibles puis une approximation de $p = \mathbb{P}(A)$.
4. On note G la variable aléatoire égale au gain algébrique d'un joueur jouant une seule grille de Loto. Une grille de Loto coûte 2,20 euros et on note J le montant du jackpot.
 - (a) Donner la loi de G puis calculer $\mathbb{E}(G)$.
 - (b) A quelle condition sur le montant du jackpot l'espérance de gain du joueur est-elle positive ? On utilisera **Python** pour faire le calcul approché.
5. Le journal local a fait un reportage sur Jean-Michel Leveinard après qu'il ait gagné le jackpot alors qu'il jouait depuis 40 ans. On suppose dans cette question que Jean-Michel jouait exactement une grille par semaine.
 - (a) Combien de grilles a-t-il joué en tout ?
 - (b) On note X la variable aléatoire égale au nombre de jackpot gagnés sur l'ensemble des grilles jouées sur 40 ans. Reconnaître la loi de X .
 - (c) Calculer à l'aide de **Python** une approximation de la probabilité que Jean-Michel avait de gagner au moins une fois en jouant ainsi pendant 40 ans.
6. Une autre habitante du coin, Micheline Lapatiente, décide en lisant le reportage qu'elle jouera une grille par semaine pendant n semaines consécutives. On note X la variable aléatoire égale au nombre de jackpot gagnés pendant ces n semaines de jeu.
 - (a) Quelle approximation peut-on faire sur la loi suivie par X lorsque n est grand ?
 - (b) Micheline souhaite avoir au moins une chance sur deux de réussir dans sa quête du jackpot. En utilisant l'approximation précédente, donner une condition sur n pour que l'on ait $\mathbb{P}([X \geq 1]) \geq \frac{1}{2}$. On utilisera **Python** pour faire le calcul approché. Combien d'années cela représente-t-il ?

2 Le Bon, la Brute et le Truand

Le Bon, la Brute et le Truand s'affrontent lors d'un dernier échange de coups de feux pour déterminer qui repartira vivant, mais surtout, avec les pièces d'or. On considère que les trois tireurs font feux jusqu'à ce qu'au moins deux d'entre eux soient hors de combat, selon les règles suivantes :

- Lorsque le Bon tire, la probabilité pour qu'il atteigne son adversaire vaut $\frac{2}{3}$.
- Lorsque le Truand tire, la probabilité pour qu'il atteigne son adversaire vaut $\frac{1}{2}$.
- Lorsque la Brute tire, la probabilité pour qu'il atteigne son adversaire vaut $\frac{1}{3}$.
- Lorsqu'un des tireurs est atteint, il est définitivement hors de combat.
- Tout le monde tire simultanément, et chacun d'eux vise le plus dangereux de ses rivaux encore en combat.
(Ainsi, au premier coup de feu, le Bon vise le Truand tandis que la Brute et le Truand visent le Bon)
- On suppose que tous les tirs effectués à un instant donné sont mutuellement indépendants.

Proposer une modélisation de cette situation qui s'appuie sur le cours de probabilités puis, en s'aidant de cette modélisation, écrire plusieurs fonctions **Python** permettant de répondre aux questions suivantes :

1. Qui a le plus de chances de remporter ce combat ? Plus précisément, quelle est la probabilité que le Bon remporte le combat ? Même question pour la Brute et le Truand.
2. Combien de rounds en moyenne va durer ce combat ?

Les réponses changent-elles lorsque les tirs se font à tour de rôle (par exemple le Bon tire, puis la Brute tire, puis le Truand tire, puis le Bon tire, etc) ?

Vous pouvez aussi inventer vos propres questions (éventuellement en changeant les règles du « jeu ») et essayer d'y répondre en utilisant **Python** !

