Le paradoxe des boîtes de Bertrand

Niveau : première générale spécialité.

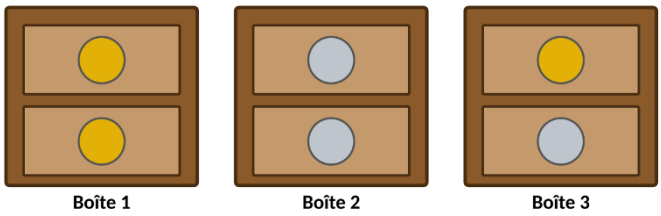
Lien avec le programme : probabilités conditionnelles : arbre pondéré, formule des probabilités totales, calcul de $P_A(B)$; variable aléatoire, espérance. Algorithmique et programmation : simulation d'une expérience aléatoire, fréquence observée.

Lien avec *Les Maths au quotidien* : Insolite.

En 1889, le mathématicien français Joseph Bertrand publie un traité, *Calcul des probabilités*, dans lequel il propose l'énigme suivante.

On dispose de trois boîtes identiques, possédant chacune deux tiroirs. Chaque tiroir contient une pièce :

- la boîte 1 contient deux pièces d'or ;
- la boîte 2 contient deux pièces d'argent ;
- la boîte 3 contient une pièce d'or et une d'argent.



On choisit une boîte au hasard, puis on ouvre au hasard l'un de ses deux tiroirs : il contient une **pièce d'or**. Quelle est la probabilité que l'autre tiroir de la même boîte contienne lui aussi une pièce d'or ?

Partie A : expérimenter

1. Avant tout calcul, quelle réponse vous semble la bonne ? Noter votre réponse.

2. On remplace les boîtes par trois cartes : une carte dorée sur ses deux faces, une carte argentée sur ses deux faces, et une carte dorée d'un côté et argentée de l'autre. On met les cartes dans un sac, on en tire une au hasard et on la pose sur la table sans regarder : seule la face du dessus est visible.

a. Expliquer pourquoi cette expérience reproduit exactement celle des boîtes.

b. Par groupes, fabriquer les trois cartes et réaliser 30 tirages. Ne conserver que les tirages où la face visible est dorée, et calculer la fréquence des cas où la face cachée est aussi dorée. Mettre en commun les résultats de la classe.

3. On simule l'expérience des boîtes avec le programme Python ci-dessous. La fonction `tirage` choisit une boîte et un tiroir au hasard, puis renvoie la couleur de la pièce vue et celle de la pièce cachée.

```
from random import choice
boites = [("or", "or"), ("argent", "argent"), ("or", "argent")]
def tirage():
    boite = choice(......)          # (1) choix d'une boîte au hasard
    i = choice([0, 1])            # choix d'un tiroir au hasard
    vue = boite[i]
    cachee = boite[......]          # (2) la pièce de l'autre tiroir
    return vue, cachee
```

La suite du programme répète l'expérience 10 000 fois et compte les tirages utiles :

```
n_or = 0       # nombre de tirages où la pièce vue est en or
n_or_or = 0    # nombre de tirages où les deux pièces sont en or
for k in range(......):             # (3) répéter 10 000 fois
    vue, cachee = tirage()
    if vue == "or":                       # (4)
        n_or = .........
        if ............... :               # (5)
            n_or_or = n_or_or + 1
print(...............)                    # (6) afficher la fréquence cherchée
```

a. Compléter les pointillés repérés de (1) à (6).

b. Exécuter le programme plusieurs fois. Quelles fréquences obtient-on ? Comparer avec votre réponse à la question 1 et avec les résultats de la classe à la question 2.

c. Pourquoi divise-t-on par n_or et non par 10 000 ? Quelle valeur obtiendrait-on en divisant par 10 000 ?

Partie B : calculer

On note B_1, B_2, B_3 les événements « la boîte choisie est la boîte 1, 2 ou 3 » et O l'événement « la pièce vue est en or ».

1. Construire l'arbre pondéré de l'expérience (choix de la boîte, puis choix du tiroir).

2. Calculer $P(O)$ à l'aide de la formule des probabilités totales.

3. Remarquer que l'autre pièce est en or si et seulement si la boîte choisie est la boîte 1. Calculer $P_O(B_1)$ et conclure.

4. Beaucoup de personnes raisonnent ainsi : « Puisque j'ai vu une pièce d'or, ce n'est pas la boîte 2. Il reste la boîte 1 et la boîte 3, donc il y a une chance sur deux que l'autre pièce soit en or. » Où se trouve l'erreur de ce raisonnement ? On pourra compter les pièces d'or qui peuvent avoir été vues.

Partie C : l'arnaque des trois cartes

Le même piège est utilisé dans une arnaque de rue décrite notamment par le mathématicien américain Warren Weaver en 1950 : l'arnaque des trois cartes.

Un escroc propose le jeu suivant avec les trois cartes de la question 2. Il tire une carte au hasard et la pose sur la table. Quelle que soit la couleur visible, il parie que la face cachée est **de la même couleur** que la face visible. Chaque joueur mise 10 € ; le gagnant empoche les 20 €.

Le joueur pense faire une bonne affaire : « Si la face visible est dorée, la carte est la dorée-dorée ou la dorée-argentée : j'ai une chance sur deux. »

1. Démontrer que la probabilité que la face cachée soit de la même couleur que la face visible vaut $\frac{2}{3}$, quelle que soit la couleur visible.

2. On note G le gain algébrique de l'escroc lors d'une partie. Donner la loi de G et calculer son espérance. Combien l'escroc peut-il espérer gagner, en moyenne, sur 30 parties ?

3. Pour que le jeu soit équitable, combien l'escroc devrait-il payer au joueur lorsque celui-ci gagne, si l'escroc continue de gagner 10 € quand il a raison ?

Partie D : et si l'on change les boîtes ?

On ajoute une quatrième boîte contenant, elle aussi, une pièce d'or et une pièce d'argent. On choisit une boîte au hasard parmi les quatre, puis un tiroir au hasard : il contient une pièce d'or. Calculer la probabilité que l'autre pièce soit en or. Le résultat est-il toujours $\frac{2}{3}$? Expliquer.

Point info :

Le même piège se cache dans le célèbre **problème de Monty Hall**, inspiré d'un jeu télévisé américain. En 1990, la journaliste Marilyn vos Savant en donna la bonne réponse dans un magazine : elle reçut des milliers de lettres de protestation, dont certaines signées par des mathématiciens convaincus qu'il y avait « une chance sur deux ». Le paradoxe de Bertrand rappelle qu'une information (ici, la couleur de la pièce vue) ne se contente pas d'éliminer des cas : elle modifie aussi le poids de ceux qui restent.