

**PREMIER PROBLÈME DU CHEVALIER DE MÉRÉ**

Les joueurs de dés savent depuis des siècles que lorsqu'on lance 4 dés, on a environ autant de chances d'avoir un 6 (au moins) que de ne pas en avoir. Dans ce TP, on va effectuer des simulations pour vérifier cela.



**I/ Lancer des quatre dés**

Tester le script Python suivant pour jouer 10 parties :

```
from random import *
for n in range(10):
    print([randint(1,6) for i in range(4)])
```

Sur les 10 parties, combien sont-elles gagnantes (au moins un 6 parmi les 4 résultats) ?.....

Pour faciliter le comptage, on peut l'automatiser avec le script suivant :

```
for n in range(10):
    print([randint(1,6) for i in range(4)].count(6))
```

Cette fois-ci, combien de parties gagnantes compte-t-on sur les 10 parties ?.....

**II/ Fréquences de gains**

Pour simplifier la suite, on va créer une fonction qui affiche si la partie est gagnante ou non :

```
def youpi():
    return [randint(1,6) for i in range(4)].count(6)>0
```

Après avoir créé cette fonction, rejouer les 10 parties en faisant

```
liste=[youpi() for n in range(10)]
print(liste)
```

Cette fois-ci, combien de parties gagnantes sur les 10 ?.....

Compléter le script ci-dessus pour qu'il affiche la fréquence des parties gagnantes sur les 10 :

```
liste=[youpi() for n in range(10)]
print( )
```

Quelle est la fréquence des parties gagnantes sur les 10 ?.....

Modifier le script ci-dessus pour qu'il affiche la fréquence des parties gagnantes sur 10000 :

```
liste=[youpi() for n in range( )]
print( )
```

Donner la fréquence des parties gagnantes :.....

Cette fréquence est-elle supérieure ou inférieure à 50 % ? .....

En fait, ce résultat était connu du Chevalier de Méré (1607-1684) comme en témoigne ce que dit de lui Blaise Pascal (1623-1662) dans une lettre écrite en 1654 à Pierre de Fermat (?-1665) :

Il me disait donc qu'il avait trouvé fausseté dans les nombres par cette raison :  
 Si on entreprend de faire un six avec un dé, il y a avantage de l'entreprendre en 4, comme de 671 à 625.  
 Si on entreprend de faire Sonnez avec deux dés, il y a désavantage de l'entreprendre en 24.  
 Et néanmoins 24 est à 36 (qui est le nombre des faces de deux dés) comme 4 à 6 (qui est le nombre des faces d'un dé).

La seconde phrase se traduit en langage moderne par « la probabilité d'avoir au moins un 6 en lançant 4 dés est  $\frac{671}{671 + 625}$  ». La troisième phrase du Chevalier de Méré se traduirait par « la probabilité d'avoir un double 6 en lançant deux dés 24 fois de suite est inférieure à  $\frac{1}{2}$  ». La dernière phrase se traduirait par  $\frac{24}{36} = \frac{4}{6}$ .

Calculer  $\frac{671}{671 + 625}$  à  $10^{-4}$  près :.....

Comparer avec la fréquence mesurée sur 10 000 parties, en donnant l'erreur d'approximation :.....

Simplifier les deux fractions  $\frac{24}{36}$  et  $\frac{4}{6}$  :

$$\frac{24}{36} = \frac{\quad}{\quad}; \text{ et } \frac{4}{6} = \frac{\quad}{\quad}$$

### III/ Exercices non notés

On peut retrouver le résultat du Chevalier de Méré en calculant la probabilité avec Python (à compléter) :

```
omega=[(a,b,c,d) for a in range(1,7) for b in range(1,7) ....]
print(len(omega))
print(6**4)
print(671+625)
```

Compléter alors le script qui donne le numérateur :

```
gain=[(a,b,c,d) for (a,b,c,d) in omega if (
print(len(gain))
```

Puis, si le temps le permet, essayer de vérifier le deuxième phénomène dénoncé par le Chevalier de Méré : En lançant 24 fois de suite deux dés, la probabilité d'avoir au moins une fois un double 6 est légèrement inférieure à  $\frac{1}{2}$ . On pourra faire des boucles en Python pour calculer la probabilité, ou faire une simulation en répétant un grand nombre de fois la série de 24 lancers et en évaluant la fréquence de l'apparition d'au moins un double 6 parmi les 24 lancers. On pourra utiliser un tableur...