

# DM

Informatique pour tous, première année

Julien REICHERT

Ce sujet propose de faire réaliser par l'ordinateur un calcul de probabilités pointu. Il a été étoffé pour mettre un maximum de questions de programmation relativement aisées. Les questions sont préfixées par une lettre : M pour mathématiques, I pour informatique fondamentale et P pour programmation.

Le contexte est le jeu « Yam's ». Aucune connaissance du jeu ou de la théorie des jeux n'est nécessaire ici.

Le but du jeu est de réaliser diverses combinaisons à l'aide de cinq dés, avec la possibilité de relancer un ou plusieurs dés jusqu'à deux fois.

La meilleure combinaison, rapportant le plus de points, est le Yam's : cinq dés identiques. C'est à celle-ci uniquement que nous nous intéresserons.

Dans tout le sujet, on assimilera un lancer de  $k$  dés à un tirage indépendant de  $k$  entiers de 1 à 6 suivant tous la loi uniforme sur l'univers associé. On le représentera par la liste de ces entiers.

Question M1 : Quelle est la probabilité de réaliser un Yam's au premier lancer ?

Question P1 : Écrire en Python une fonction qui détermine si un lancer (en argument) correspond à un Yam's.

Question P2 : Écrire en Python une fonction qui simule  $n$  lancers ( $n$  en argument) de 5 dés et qui calcule la fréquence des Yam's. Tester la fonction pour de grandes valeurs de  $n$  et comparer avec la probabilité calculée à la question M1.

Après le premier lancer, le joueur a tout intérêt de conserver un sous-ensemble de dés qui maximise sa probabilité de réaliser finalement un Yam's (on se place dans l'hypothèse où il ne souhaite pas tenter d'autre combinaison).

Question I1 : Justifier qu'il n'est pas strictement optimal de relancer tous les dés, quel que soit le résultat du premier lancer.

Question I2 : Plus généralement, justifier que le joueur a tout intérêt à conserver un sous-ensemble de dés identiques de taille maximale.

Nous supposerons par la suite que le joueur conserve effectivement un sous-ensemble de dés identiques de taille maximale.

Question M2 : Déterminer la probabilité que le plus grand sous-ensemble de dés identiques au premier lancer soit de taille quatre exactement, puis de même pour trois exactement et deux exactement. En déduire la probabilité que les cinq dés aient une valeur différente au premier lancer.

Question P3 : Écrire en Python une fonction qui détermine le nombre maximum de dés identiques d'un lancer (en argument).

C'est maintenant que les choses se compliquent : pour chacun des cinq scénarios envisagés, le nombre de dés relancés sera différent, et les cinq scénarios possibles après le deuxième lancer auront une probabilité différente suivant le scénario après le premier lancer. Ainsi, le calcul global sera obtenu par un programme.

Question M3 : Déterminer la probabilité que trois dés exactement soient identiques au deuxième lancer, alors que deux dés identiques avaient été conservés après le premier lancer.

Question M4 : En remarquant que le nombre de dés identiques ne peut pas diminuer d'un lancer à l'autre, déterminer le nombre de probabilités conditionnelles à calculer pour passer du premier lancer au deuxième lancer.

Question M5 : Justifier que pour le troisième lancer, les mêmes probabilités conditionnelles peuvent être utilisées, et qu'elles ne dépendent en particulier que du nombre de dés conservés au deuxième lancer, sans tenir compte du résultat du premier lancer. Il s'agit de la notion de chaînes de Markov.

Question M6 : Calculer les autres probabilités conditionnelles.

Question P4 : Écrire en Python un programme qui détermine à partir de toutes les données déterminées dans cet exercice la probabilité de réaliser un Yam's.

Questions M7 et P5 : Une partie de Yam's se déroule en douze tours. Quelle est la probabilité de réussir un Yam's deux fois au cours d'une partie, en considérant qu'on tente de le réaliser à chaque tour ? (Donner la formule à partir de la probabilité  $p$  de réaliser un Yam's au cours d'un tour et écrire un programme qui applique cette formule.)