Le paradoxe d'un duc de Toscane – Parier sur la somme des points de 3 dés

C. Problème historique d'un duc de Toscane – première S terminale S (fichier professeur : JeuDeDe0.py)

- Un duc de Toscane (du XVIe au XIXe siècle) avait remarqué que, bien qu'il y ait autant de façons d'écrire 9 et 10 comme somme de 3 nombres compris entre 1 et 6, on obtient plus souvent un total de 10 lorsqu'on lance 3 dés. L'objectif de l'activité est de vérifier la constatation du duc en faisant des simulations et d'expliquer le paradoxe en élaborant et en utilisant un modèle probabiliste.
- Nous allons généraliser le problème historique en recherchant la distribution de la variable aléatoire S3d somme des valeurs obtenues en jetant aléatoirement 3 dés équilibrés. Cette distribution nous permet de répondre au paradoxe soulevé par le problème historique. Nous verrons deux façons de travailler sur cette distribution ; par la simulation pour obtenir une distribution de fréquences simulées et par l'obtention de la distribution de probabilités.

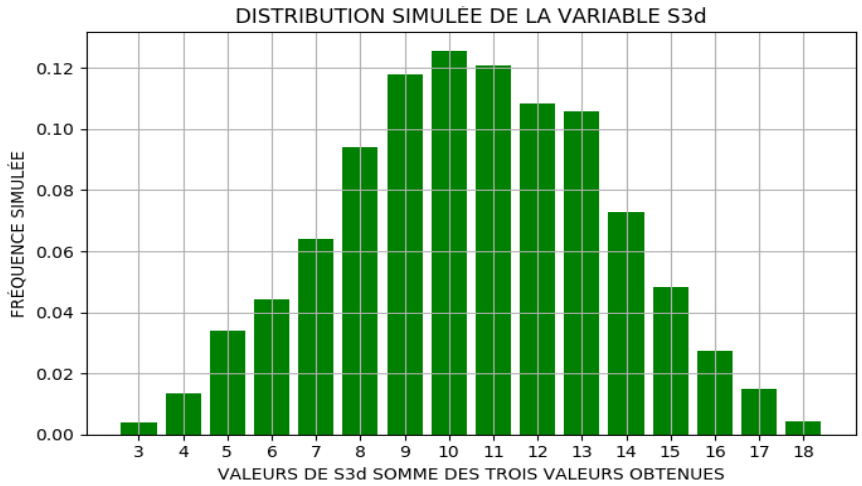
C.1. Une simulation (fichier professeur : JeuDeDe0.py lignes 9 à 68)

L'algorithme simule nbsim lancers de trois dés et réalise (selon divers algorithmes) le tableau des effectifs et des fréquences des sommes obtenues. La variable somme des 3 chiffres obtenus est appelée S3d.

► Un exemple des résultats obtenus avec la fonction `S3d-1(5000)` :

S3di	3	4	5	6	7	8	9	10
pi	0,0038	0,0136	0,034	0,0442	0,064	0,094	0,1178	0,1256

S3di	11	12	13	14	15	16	17	18
pi	0,1208	0,1086	0,1058	0,073	0,0484	0,0272	0,0148	0,0044



C.2. La distribution de probabilités (fichier professeur : JeuDeDe0.py lignes 71 à 121)

L'algorithme détermine l'ensemble de tous les triplets possibles, en créant une liste 'en compréhension' sur trois indices. Cette liste est ensuite traitée pour en tirer le tableau de distribution de probabilité de la variable aléatoire S.

C.3. Prolongements ☺

a. Dans les fonctions, faire afficher les fréquences simulées et les probabilités des valeurs 9 et 10, précédé d'un message.

b. Élaborer une fonction Python permettant de **simuler la distribution des fréquences** de la somme des chiffres obtenus en lançant **2 dés** à 6 faces équilibrées.

c. Élaborer une fonction Python permettant d'**obtenir la distribution des probabilités** de la somme des chiffres obtenus en lançant **2 dés** à 6 faces équilibrées.

Nom Prénom :

TP_ALGO_C.1. (Fiche élève 1/3, fichier JeuDeDe0_Elev.py, lignes 9 à 33, algorithme S3-1)

● Dans Spyder, exécuter les lignes de commandes des **lignes 9 à 33** et compléter le tableau suivant en recopiant les résultats affichés dans la console.

```
"------------------- ALGO S3-1 ----------------------------------"
"Distribution simulée de la somme S3d des faces de trois dés à 6 faces équilibrées"
"Le tableau des effectifs est construit au fur et à mesure des lancers."
"Le lancer des dés est fait dans une boucle et l'effectif des numéros obtenus augmente de 1 à chaque tour."
"La liste des effectifs des 16 valeurs (3 à 18) est initialisée à 0."
from random import randint

"Nombre de simulations"
nbsim = 5000

"Initialisation à 0, de la série des effectifs des 16 sommes possibles"
serieEffectifs = [0] * 16
serieEffectifs

Résultat :     ... ;     ... ;     ... ;     ... ;     ... ;     ... ;     ... ;     ... ;

               ... ;     ... ;     ... ;     ... ;     ... ;     ... ;     ... ;     ...

"nbsim simulations du lancer de 3 dés. Somme des 3 valeurs obtenues"
for n in range(nbsim):
    de1 = randint(1, 6)
    de2 = randint(1, 6)
    de3 = randint(1, 6)
    troisDes = de1 + de2 + de3
    "Les effectifs des sommes obtenues sont cumulés dans la liste serieEffec"
    serieEffectifs[troisDes - 3] += 1
serieEffectifs

Résultat :     ... ;     ... ;     ... ;     ... ;     ... ;     ... ;     ... ;     ... ;

               ... ;     ... ;     ... ;     ... ;     ... ;     ... ;     ... ;     ...
```

Nom Prénom :

TP_ALGO_C.1. (Fiche élève 2/3, fichier JeuDeDe0_Elev.py, lignes 35 à 54, algorithme S3-1)

1° Dans Spyder, exécuter les lignes de commandes des **lignes 35 à 55** et compléter le tableau suivant en recopiant les résultats affichés dans la console.

2° Pour garder une trace du graphique obtenu, le copier dans une page du traitement de texte (Libre office writer par exemple).

```
"Construction et affichage du 'tableau' des effectifs, S3di, ni"
tabloEffec = dict(zip(range(3, 19), serieEffectifs))
tabloEffec

Résultat :     ... ;     ... ;     ... ;     ... ;     ... ;     ... ;     ... ;     ... ;

               ... ;     ... ;     ... ;     ... ;     ... ;     ... ;     ... ;     ...

"Construction et affichage du 'tableau'des fréquences, S3di, fi"
serieFreq = [round(serieEffectifs[i] / nbsim, 4) for i in range(16)]
serieFreq

Résultat :     ... ;     ... ;     ... ;     ... ;     ... ;     ... ;     ... ;     ... ;

               ... ;     ... ;     ... ;     ... ;     ... ;     ... ;     ... ;     ...

tabloFreq = dict(zip(range(3, 19), serieFreq))
tabloFreq

Résultat :     ... ;     ... ;     ... ;     ... ;     ... ;     ... ;     ... ;     ... ;

               ... ;     ... ;     ... ;     ... ;     ... ;     ... ;     ... ;     ...

"-----Le graphique en barres du tableau des fréquences simulées------"
x = range(3, 19)
from matplotlib import pyplot as plt
plt.figure()
plt.bar(x, serieFreq, color = 'green') ; plt.xticks(x)
plt.title("DISTRIBUTION SIMULÉE DE LA VARIABLE S3d")
plt.xlabel("VALEURS DE S3d SOMME DES TROIS VALEURS OBTENUES")
plt.ylabel("FRÉQUENCE SIMULÉE")
plt.grid() ; plt.show()
```

Nom Prénom :

TP_ALGO_C.1. (Fiche élève 3/3, fichier JeuDeDe0_Elev.py, lignes 57 à 83, algorithme S3-1)

1° Compléter les lignes manquantes de la fonction suivante pour simuler la somme des valeurs obtenues en lançant 3 dés équilibrés et qui affiche le "tableau" des effectifs, le "tableau" des fréquences et le graphique.

2° Dans Spyder copier-coller les lignes de commandes correspondant aux lignes 62 à 73. Exécuter et utiliser la fonction **S3d-1(…)** pour faire des simulations de 100, 10000 et 1000000 répétitions. Les résultats seront copiés-collés dans le traitement de texte.

```
def S3d-1(nbsim):
     from random import randint
     from matplotlib import pyplot as plt
     ...
     ...
          ...
          ...
          ...
          ...
          ...
     ...
     ...
     ...
     ...
     ...
     # Construction et affichage du graphique
     x = range(3, 19)
     plt.figure()
     plt.bar(x, serieFreq, color = 'green') ; plt.xticks(x)
     plt.title("DISTRIBUTION SIMULÉE DE LA VARIABLE S3d")
     plt.xlabel("VALEURS DE S3d SOMME DES TROIS VALEURS OBTENUES")
     plt.ylabel("FRÉQUENCE SIMULÉE")
     plt.grid() ; plt.show()
```

Nom Prénom :

TP_ALGO_C.2. (Fiche élève 1/3, fichier JeuDeDe0_Elev.py, lignes 86 à 107, algorithme P-S3-1)

● Dans Spyder, exécuter les lignes de commandes des **lignes 96 à 107** et compléter le tableau suivant en recopiant les résultats affichés dans la console.

```
"Univers de la variable aléatoire S3d : somme des faces de trois dés équilibrés à six faces"
"Tableau des effectifs des valeurs de la v.a. S3d"
"Distribution de probabilité de la v.a. S3d"

"Liste des 6x6x6=216 triplets obtenus en lançant 3 dés"
UniversTriplets = [(i, j, k) for i in range(1, 7) for j in range(1, 7) for k in range(1, 7)]
"J'affiche la liste des 8 premiers triplets obtenus"
UniversTriplets[0:7]

Résultat :     ... ;     ... ;     ... ;     ... ;     ... ;     ... ;     ... ;     ... ;

"Cardinal de l'univers = longueur de la liste UnivTriplets"
cardUniversTriplets = len(UniversTriplets)

Résultat :     ...

"J'utilise UniversTriplets pour faire la somme de chacun des 216 triplets"
UnivS3d = [sum(UniversTriplets[i]) for i in range(cardUniversTriplets)]
"J'affiche la liste des 8 premières sommes obtenues"
UnivS3d[0:7]

Résultat :     ... ;     ... ;     ... ;     ... ;     ... ;     ... ;     ... ;     ... ;

"Initialisation à 0 de la liste des effectifs des valeurs de la v.a. S3d"
listeEff = [0] * 16
"J'affiche la liste des 8 premières valeurs initiales des effectifs"
listeEff[0:7]

Résultat :     ... ;     ... ;     ... ;     ... ;     ... ;     ... ;     ... ;     ... ;
```

Nom Prénom :

TP_ALGO_C.2. (Fiche élève 2/3, fichier JeuDeDe0_Elev.py, lignes 109 à 128, algorithme P-S3-1)

● Dans Spyder, exécuter les lignes de commandes des **lignes 109 à 128** et compléter le tableau suivant en recopiant les résultats affichés dans la console.

```
"Création de la liste des effectifs des valeurs de S3d"
for k in range(cardUniversTriplets):
    listeEff[UnivS3d[k] - 3] += 1
"J'affiche la liste des effectifs des 16 valeurs de la variable S3d"
listeEff[0:7]
Résultat :     ... ;     ... ;     ... ;     ... ;     ... ;     ... ;     ... ;     ... ;

               ... ;     ... ;     ... ;     ... ;     ... ;     ... ;     ... ;     ...
"Calcul des probabilités des valeurs de S3d"
Proba = [round(listeEff[i] / cardUniversTriplets, 4) for i in range(16)]
"J'affiche la liste des probabilités des 16 valeurs de la variable S3d"
Proba[0:7]
Résultat :     ... ;     ... ;     ... ;     ... ;     ... ;     ... ;     ... ;     ... ;

               ... ;     ... ;     ... ;     ... ;     ... ;     ... ;     ... ;     ...
"Tableau de la distribution des effectifs et du tableau de la distribution de probabilité de S3d"
tabloEffec = dict(zip(range(3, 19), listeEff))
tabloProba = dict(zip(range(3, 19), Proba))
"J'affiche le tableau des effectifs et le tableau de la distribution de probabilité"
tabloEffec
tabloProba
Résultat tabloEffec :    ... ;     ... ;     ... ;     ... ;     ... ;     ... ;     ... ;     ... ;

                         ... ;     ... ;     ... ;     ... ;     ... ;     ... ;     ... ;     ...
Résultat tabloProba :    ... ;     ... ;     ... ;     ... ;     ... ;     ... ;     ... ;     ... ;

                         ... ;     ... ;     ... ;     ... ;     ... ;     ... ;     ... ;     ...
```

Nom Prénom :

TP_ALGO_C.2. (Fiche élève 3/3, fichier JeuDeDe0_Elev.py, lignes 141 à 160, algorithme P-S3-1)

1° En utilisant des lignes de commandes des pages précédentes, compléter la fonction suivante qui détermine la distribution de probabilité de la somme (S3d) des valeurs obtenues en lançant 3 dés équilibrés et qui affiche le tableau des probabilité et sa représentation graphique.

2° Dans Spyder copier-coller les lignes de commandes correspondant aux **lignes 145 à 152**. Exécuter et utiliser la fonction **probaS3d-1(…).** Copier coller les résultats et le graphique obtenu dans un traitement de texte (LibreOffice writer par exemple).

```
def probaS3d():
    from matplotlib import pyplot as plt

    ...

    ...

    ...

    for k in range(cardUniversS3d):

        ...

    ...

    ...

    ...
    x = range(3, 19)
    plt.figure()
    plt.bar(x, Proba) ; plt.xticks(x)
    plt.title("DISTRIBUTION DE PROBABILITÉ DE LA VARIABLE S3d")
    plt.xlabel("VALEURS DE S3d SOMME DES TROIS CHIFFRES OBTENUS")
    plt.ylabel("PROBABILITÉ")
    plt.grid() ; plt.show()
```

Nom Prénom : **PROLONGEMENTS**

TP_ALGO_C.3. (Fiche élève ...../.....)

Proposition de fonction Python répondant aux questions proposées en **C.3.** À faire valider par l'enseignant.

```
Def ......:
    ....
    ...
    ...
    ...
    ....


    ...
    ...
    ...
    ...
    ...
    ...
    ...
    ...
    ...
    ...
    ...
    ...
```