Feuille de cours 11 (informatique) : probabilités

Via des techniques non triviales, Python peut générer des résultats aléatoires. On peut donc utiliser l'outil informatique pour *simuler* des expériences aléatoires et pour *estimer* la probabilité d'un évènement ou l'espérance d'une variable aléatoire.

On utilise la bibliothèque `random` qu'on importe classiquement avec l'alias `rd`. On écrira donc :

1 Simuler

1.1 Fonctions usuelles

Parmi les fonctions de la bibliothèque `random` il faut connaître les fonctions suivantes :
- `rd.random()` : renvoie un nombre réel aléatoire entre 0 et 1
- `rd.uniform(`a`,`b`)` : renvoie un nombre réel aléatoire entre a et b
- `rd.randint(`a`,`b`)` : renvoie un nombre entier aléatoire entre a et b (inclus !)
- `rd.choice(L)` : renvoie un élement aléatoire de la liste `L`
- `rd.sample(L,`k`)` : renvoie une liste de k éléments de `L` pris sans répétition

Exercice 1
1. Alice choisit au hasard un nombre réel entre -3 et 3 puis l'élève au carré. Écrire une fonction `Alice` ne prenant rien en argument et renvoyant le résultat calculé par Alice.
2. Bob choisit au hasard deux nombres entiers entre -3 et 3 et les additionne. Écrire une fonction `Bob` ne prenant rien en argument et renvoyant le résultat calculé par Bob.

Exercice 2
Une urne contient 3 boules rouges, 2 boules vertes et 1 boule jaune. On tire 4 boules dans l'urne et on note les résultats dans une liste `L`.
1. Créer une liste `urne` correspondant au contenu de l'urne.
2. On suppose que les tirages sont successifs et avec remise. Écrire un code permettant d'obtenir la liste `L`.
3. Même question avec des tirages sans remise.

1.2 Simuler un évènement de probabilité donnée

Les fonctions précédentes permettent uniquement de simuler des résultats choisi *uniformément* au hasard. Par exemple, si `L = ["a","b","c"]` alors l'instruction `rd.choice(L)` renverra `"a"`, `"b"` ou `"c"` chacun avec probabilité .

Cependant, il est courant de vouloir obtenir des résultats avec une certaine probabilité fixée, et ne correspondant pas à une situation d'équiprobabilité. Par exemple, on peut vouloir renvoyer `"a"` avec probabilité $2/3$ et `"b"` avec probabilité $1/3$.
Que pourrait-on faire pour traiter cet exemple ?

Cette astuce est adaptée dans des cas simples, mais devient difficile à mettre en œuvre lorsqu'on souhaite simuler un évènement de probabilité p quelconque (par exemple $p = 0,276$, $p = \sqrt{2} - 1$, ou p une variable prise en argument).
On utilise alors une autre technique usuelle à connaître.
Dessinez ci-dessous un segment de longueur L et placez deux points A et B sur ce segment distants d'une longueur $AB = \ell$. On imagine qu'on place un point M au hasard sur le segment. Quelle est la probabilité que M appartienne au segment $[A, B]$?

En pratique, pour aller au plus simple : on prend $L = 1$ et on place le point A tout à gauche du segment. Ainsi, **pour simuler un évènement de probabilité p, on suivra la technique suivante :**

- tirer un nombre t entre 0 et 1 au hasard avec la commande :
- puis tester si

On écrit donc :

```
t = rd.random()
if t<=p :
    # instructions à faire avec proba p
else :
    # instructions à faire avec proba 1-p
```

Exercice 3
Écrire une fonction `alea` prenant en argument un réel $p \in [0, 1]$ et renvoyant 1 avec probabilité p et 0 avec probabilité $1 - p$.

Exercice 4
Écrire une fonction `pièce` ne prenant rien en argument et simulant le résultat d'une pièce truquée tombant sur Pile avec probabilité $0,35$.

Remarque : lorsqu'on choisit un nombre *réel* t au hasard entre 0 et 1, quelle est la probabilité que t soit exactement égal à p ?
En conséquence, dans le code ci-dessus, on peut remplacer le test `t <= p` par `t < p`.

Une variante importante du raisonnement ci-dessus consiste à simuler une expérience à trois issues (ou plus) dont on connaît les probabilités p_1, p_2 et $1 - p_1 - p_2$. Il faut alors faire attention à utiliser des évènements *disjoints* de probabilités respectives p_1, p_2 et $1 - p_1 - p_2$. Si t est un nombre réel aléatoire entre 0 et 1 on peut par exemple utiliser les évènements $(t \leq p_1)$, $(p_1 < t \leq p_1 + p_2)$ et $(p_1 + p_2 < t \leq 1)$.

On écrit :

```
t = rd.random()
if t <= p1 :
    # instructions à faire avec proba p1
elif t <= p1+ p2 : # et non t <= p2
    # instructions à faire avec proba p2
else :
    # instructions à faire avec proba 1-p1-p2
```

Attention, placer la condition `t <= p2` au test `elif`, reviendrait en fait à utiliser l'évènement $(p_1 < t \leq p_2)$ qui est de probabilité $p_2 - p_1$ (si $p_2 > p_1$) et non p_2 comme souhaité !

Exercice 5
Une urne contient 100 boules dont 45 sont rouges, 25 sont bleues et 30 sont jaunes. Écrire une fonction `tirage` permettant de simuler le tirage d'une boule dans cette urne.

2 Estimer

Une fois qu'on a créé un programme Python permettant de simuler une expérience aléatoire, on peut s'en servir pour estimer la probabilité d'un évènement ou l'espérance d'une variable aléatoire. Attention, contrairement à ce qu'on demande de faire en mathématiques, il ne s'agit pas ici de faire un calcul exact de probabilités (mais il est souvent demandé de comparer le résultat obtenu informatiquement avec le résultat exact).

Pour obtenir cette esimtation, on utilise une *approche fréquentielle*. Cela signifie qu'on estime une probabilité (ou une espérance) en répétant un très grand nombre de fois la même expérience.

2.1 Estimer une probabilité

Lorsqu'un évènement a une probabilité p de se produire et qu'on répète N fois l'expérience aléatoire, on s'attend à ce que cet évènement se produise environ   fois. Ainsi, pour estimer la probabilité p, on répète N fois l'expérience, on compte le nombre X de fois que l'évènement s'est produit, et p est alors environ égal à $\dfrac{X}{N}$.

Attention, pour obtenir une bonne approximation de la probabilité, il faut répéter un *grand* nombre de fois l'expérience. Prendre par exemple $N = 10\ 000$.
On écrit donc :

```
X = 0
N = 10000 # prendre N grand
for k in range(N):
    # on fait l'expérience aléatoire (à l'intérieur du for !)
    if succès : # si l'évènement souhaité se produit
        X = X + 1 # on compte un de plus
p = X/N
```

Exercice 6
On reprend l'expérience de l'exercice 1 où Alice choisit un nombre réel au hasard entre -3 et 3 et l'élève au carré. On rappelle qu'on a écrit une fonction `Alice` renvoyant le résultat de cette expérience. Estimer la probabilité pour que le résultat obtenu soit supérieur à 4.

Exercice 7
On choisit 2 nombres entiers au hasard entre 0 et 10. Estimer la probabilité que leur somme soit égale à 12. On commencera par écrire une fonction `expérience` renvoyant le résultat de cette expérience.

2.2 Estimer une espérance

Dire qu'une variable aléatoire X a une espérance m correspond intuitivement à dire que X vaut m *en moyenne*. C'est-à-dire que si on prend plusieurs réalisations indépendantes de X alors leur moyenne vaut environ m. Ainsi pour estimer la valeur de m, on peut prendre N réalisations indépendantes $X_1, X_2, \ldots, X_N$ de X, et m est alors environ égal à leur moyenne $\dfrac{X_1 + X_2 + \cdots + X_N}{N}$.

Attention, pour obtenir une bonne approximation de la probabilité, il faut répéter un *grand* nombre de fois l'expérience. Prendre par exemple $N = 10\ 000$.

On écrit donc :

```
S = 0
N = 10000 # prendre N grand
for k in range(N):
    # on fait l'expérience aléatoire (à l'intérieur du for !)
    X = mon_experience
    S = S + X
esperance = S/N
```

Pour estimer une variance par une approche fréquentielle on utilise que $\mathbb{V}(X) = \mathbb{E}(X^2) - \mathbb{E}(X)^2$ donc on fait deux calculs d'espérance : un pour $\mathbb{E}(X^2)$ et un pour $\mathbb{E}(X)$. On écrit :

```
S = 0
S_2 = 0
N = 10000 # prendre N grand
for k in range(N):
    # on fait l'expérience aléatoire (à l'intérieur du for !)
    X = mon_experience
    S = S + X
    S_2 = S_2 + X**2
esperance = S/N
esperance_2 = S_2/N
variance = esperance_2 - esperance**2
```

Exercice 8

1. Écrire une fonction `bernoulli` prenant en argument un réel p et simulant une variable aléatoire de loi $b(p)$.
2. Estimer son espérance et sa variance.
3. Mêmes questions pour une loi $\mathcal{B}(n, p)$.

3 Exercices

Exercice 9
Écrire une fonction `list_alea` prenant en argument un entier n et renvoyant une liste contenant n nombres entiers compris entre 0 et 100 et choisis au hasard.

Exercice 10
Écrire une fonction `ADN` prenant en argument un entier n et renvoyant une liste de n acides aminés choisis aléatoirement. On commencera par créer une liste contenant les acides aminés.

Exercice 11
Alice et Bob choisissent chacun indépendamment un nombre entier entre 0 et 2. Alice choisit uniformément au hasard. Bob choisit 0 avec probabilité $0,2$; 1 avec probabilité $0,5$ et 2 avec probabilité $0,3$.

1. Écrire des fonctions `Alice` et `Bob` simulant cette expérience.
2. En utilisant ces fonctions, estimer la probabilité pour que le nombre choisi par Alice soit strictement plus grand que celui choisi par Bob.
3. *Question de mathématiques.* Quelle est la valeur exacte de la probabilité estimée à la question précédente ?

Exercice 12
On pioche simultanément 2 boules dans une urne contenant 3 boules rouges et 5 boules vertes.

1. Écrire une fonction `tirage` simulant cette expérience.
2. En utilisant la fonction précédente, estimer la probabilité d'obtenir 2 boules vertes.

Exercice 13
Un athlète s'entraîne au pentathlon en choisissant chaque jour au hasard sa discipline. Il choisit : l'escrime avec probabilité $0,2$, la natation avec probabilité $0,25$, l'équitation avec probabilité $0,15$, le tir avec probabilité $0,1$, et la course avec probabilité $0,3$.

1. Écrire une fonction `choix` ne prenant rien en argument et renvoyant la discipline choisie.
2. En déduire une fonction `programme` prenant en argument un entier n et renvoyant un programme d'entraîenement sur n jours présenté sous forme d'une liste de disciplines.
3. En utilisant la fonction `programme` estimer la probabilité pour que l'athlète enchaîne, dans cet ordre, tir - escrime - natation les 3 premiers jours.

Exercice 14
Une urne contient 10 boules numérotées de 1 à 10. On effectue 4 tirages successifs d'une boule avec remise, et on compte le nombre de fois X où on a tiré la boule numéro 10.

1. Écrire une fonction `tirage` ne prenant rien en argument et renvoyant le résultat du tirage d'une seule boule.
2. Écrire une fonction `nb_dix` ne prenant rien en argument et renvoyant le nombre X.
3. Écrire un programme Python permettant d'estimer la probabilité p_k d'obtenir exactement k fois la boule numéro 10. *On doit obtenir par exemple* $p_2 \simeq 0,0486$.
4. Écrire un code Python permettant d'estimer l'espérance et la variance de X. *On doit obtenir* $\mathbb{E}(X) \simeq 0,4$ *et* $\mathbb{V}(X) \simeq 0,036$.
5. *Question de mathématiques :* quelle est la loi de X ? Confirmer alors les résultats des deux questions précédentes.