TP3B - Exercices de probabilités discrètes

1 Exercices niveau concours

Exercice 1 - Simulation de mobile - EDHEC 2018 (adapté et modifié) ⋆⋆

Cette exercice s'inspire de l'exercice 3 du second[1] sujet de l'EDHEC 2018.
Un mobile se déplace aléatoirement sur un axe dont l'origine est la point O d'abscisse 0. Au départ, à l'instant 0, le mobile se situe sur le point O. Le mobile se déplace selon la règle suivante : à l'instant n ($n \in \mathbb{N}^\star$), il se place de façon équiprobable sur l'un des points d'abscisses 0, 1, ..., n.
Pour tout entier naturel n, on note X_n l'abscisse de ce point à l'instant n. On a donc $X_0 = 0$. On admet que $(X_n)_{n\in\mathbb{N}}$ est une suite de variables aléatoires mutuellement indépendantes.
On note enfin T l'instant auquel, pour la première fois, le mobile revient à l'origine. On a donc $X_T = 0$.

1. On rappelle que l'on peut simuler un tirage d'une loi uniforme avec `rd.randint`. Compléter la fonction suivante suivante afin qu'elle simule un tirage de la variable aléatoire T :

```python
def simul_T():
  val = 1
  while ... :
    val = val + 1
  return val
```

2. Compléter la fonction suivant pour qu'elle remplisse un tableau `numpy` avec k simulations de T :

```python
def simul_Ts(k):
  simulations = np.zeros(k)
  for i in range(...):
    simulations[...] = simul_T()
  return simulations
```

   Utiliser cette fonction pour générer 10 000 simulations de T et les stocker dans un tableau que l'on nommera A.

3. Dans le sujet de l'EDHEC, il a été prouvé que $P(T = n) = \frac{1}{n(n+1)}$. **On ne cherchera pas à prouver cette formule.** Nous allons plutôt utiliser l'ordinateur pour la vérifier sur des simulations.

   (a) On considère les codes suivants :

```python
u = np.zeros(10)
for i in range(10000):
  if A[i] <= 10:
    index = int(A[i])-1
    u[index] = u[index] + 1
u = u / 10000
```

```python
v = np.zeros(10)
for i in range(1,11):
  v[i-1] = 1/(i*(i+1))
```

   Expliquer ce que fait chaque code. Les recopier et les exécuter.

   (b) On peut tracer des courbes en utiliser la bibliothèque `matplotlib`. On l'importe avec :

```python
import matplotlib.pyplot as plt
```

   Une fois importée, et en vous reportant à la fiche rappel de `Python` de début d'année, tracer sur le même graphique la loi de T et la simulation de cette loi. Les probabilités doivent apparaître en ordonnées et les valeurs possibles pour T en abscisses.
   Les commandes importantes sont `plt.plot` et `plt.show`.

1. Suite à des problèmes en série, 3 sujets différents ont été donnés l'année 2018 pour l'EDHEC. Le premier sujet a été diffusé par erreur avant le concours. Un second sujet (dont s'inspire cet exercice) a été utilisé pour le jour de l'épreuve. Mais comme une erreur a été commise dans un centre qui a donné le premier sujet, l'épreuve a été annulée et un troisième sujet a été utilisé pour une nouvelle épreuve.

4. (a) Montrer que T n'admet pas d'espérance.
   (b) On peut pourtant calculer l'espérance de T sur un grand nombre de simulations. Écrire une fonction qui remplit un tableau avec les espérances d'un nombre croissant de simulation et utiliser `matplotlib` pour en tracer la courbe. Que remarque-t-on ?

Exercice 2 - Loi de Borel ⋆⋆⋆⋆

On considère une caisse de supermarché fonctionnant sur le principe suivant : chaque client passe une minute à la caisse et pendant qu'un client est servi, le nombre de clients arrivant à la caisse est une variable aléatoire suivant une loi de Poisson $\mathcal{P}(\mu)$.

1. Écrire une fonction `caisse` de paramètre `mu` estimant le nombre de minutes nécessaires après l'arrivée du premier client avant que la caisse soit vide. Essayer cette fonction à l'aide de plusieurs valeurs de μ.
2. Constater que pour $\mu < 1$, la moyenne est de l'ordre de $\frac{1}{1-\mu}$. Essayer d'expliquer ce résultat de manière intuitive en considérant le nombre moyen de clients se présentant à la caisse pendant que le premier est servi, puis le nombre moyen de clients arrivant à la caisse pendant que ces « clients de première génération » sont servis, etc.
3. La probabilité que ce nombre de clients soit égal à n est égale à $\frac{\mathrm{e}^{-\mu n}(\mu n)^{n-1}}{n!}$. Vérifier cela à l'aide d'un graphique comme dans l'exercice 1.

2 Travail à préparer pour le prochain TP

Exercice 3 - Viande ou poisson ⋆⋆

Le nombre de clients qui entrent dans un restaurant un soir de semaine est une variable aléatoire X qui suit une loi de Poisson de paramètre μ. Chaque client choisit indépendemment du choix de ses voisins de la viande avec une probabilité p et du poisson avec une probabilité $1 - p$. On note Y le nombre de clients qui choisissent de ma viande.

1. Écrire une fonction d'en-tête :

```python
def restaurant_simul(mu,p):
```

   qui prend les deux paramètres pertinents et renvoient la valeur simulée de X et Y pour une soirée au restaurant.

   Note : pour renvoyer deux valeurs à la fois, il faut les séparer par des virgules. Par exemple :

```python
return x,y
```

2. Écrire une fonction :

```python
def restaurant_esp_y(mu,p,k):
```

   qui calcule, sur un échantillon de k soirées simulées, l'espérance de Y.

   Note : pour récupérer deux valeurs à la fois, il faut séparer par des virgules les deux variables utilisées pour les stocker. Par exemple :

```python
x,y = soiree_restaurant(mu,p)
```