TP3 - Méthode d'inversion

Dans tout le TP, on importe les modules suivants :

```
import numpy as np
import numpy.random as rd
```

1 Pour se lancer

1.1 Loi exponentielle et loi géométrique

Cette partie est relativement déconnectée de ce qui suit. Le but est de se remettre dans le bain et d'en profiter pour retravailler les probabilités discrètes et à densité.

Exercice 1 ⋆

1. Rappeler ce que fait l'instruction `rd.geometric(p)`.
2. Que peut faire l'instruction `rd.exponential(1/mu)` ?
3. Que fait le bloc d'instructions suivant ?

```
import numpy as np
import numpy.random as rd

mu = 2
N = 1000
A = rd.exponential(1/mu,N)
print(np.mean(A))
print(np.var(A))
```

Vérifier que l'affichage est cohérent avec vos attentes.

Exercice 2 ⋆⋆

Pour tout $x \in \mathbb{R}$, on désigne par $\lfloor x \rfloor$ la partie entière de x c'est-à-dire l'entier tel que $\lfloor x \rfloor \leq x < \lfloor x \rfloor + 1$.
Soit $X \hookrightarrow \mathcal{E}(\lambda)$ avec $\lambda > 0$.

1. Soit $n \in \mathbb{N}$. Justifier que $P(n \leq X < n+1) = \int_n^{n+1} f(t)\mathrm{d}t$ où f est une densité de X. Calculer cette probabilité.
2. On note $Y = \lfloor X \rfloor$. On admet que Y est une variable aléatoire. Justifier que c'est une variable aléatoire discrète.
3. Soit $n \in \mathbb{Z}$.
   (a) Que peut-on dire de $P(Y = n)$ si $n < 0$?
   (b) En s'appuyant sur la première question, déterminer $P(Y = n)$ pour tout $n \in \mathbb{N}$.
   (c) En déduire la loi de $Y + 1$.
4. Déduire des questions précédentes une méthode pour simuler une loi géométrique en s'appuyant sur la commande `rd.exponential`. Écrire une fonction d'en-tête `def new_geom(p):` simulant un tirage d'une variable suivant une loi géométrique.

1.2 Méthode d'inversion - cas discret

On va désormais aborder le cœur de ce TP : la méthode d'inversion. Comme nous l'avons vu en cours, la loi uniforme sur $[0,1]$ n'est pas forcément naturelle comme modèle mais elle est *relativement* facile à simuler. Les autres lois (lois discrètes ou à densité) sont généralement plus difficiles à simuler.
La méthode d'inversion est une technique permettant de partir d'une loi uniforme et de simuler presque n'importe quelle loi. Son domaine d'application est normalement les lois à densité mais nous allons commencer en étudiant un cas simplifié discret.

Exercice 3 $\star$

1. Rappeler ce que fait l'instruction `rd.random`.
2. Soit $X \hookrightarrow \mathcal{U}([0,1])$. Soit $p \in [0,1]$. Que vaut $P(X \leq p)$?
3. En déduire une manière de simuler une loi de Bernoulli à partir de l'instruction `rd.random`.

Exercice 4 $\star\star$

Soit X une variable aléatoire discrète prenant les valeurs $\{x_1, x_2, \ldots, x_k\}$ avec les probabilités $\{p_1, p_2, \ldots, p_k\}$. On cherche à écrire une fonction capable de simuler cette variable.

1. Soit $Y \hookrightarrow \mathcal{U}([0,1])$. Que vaut $P(0 \leq Y < p_1)$? Et $P(p_1 \leq Y < p_1 + p_2)$?
2. Justifier que $p_1 + p_2 + \cdots + p_k = 1$.
   On place des points ℓ_i sur le segment $[0,1]$ définis par $\ell_i = \sum_{m=1}^{i} p_m$. Faire le schéma correspondant. Où se situe ℓ_0 ? Et ℓ_k ?
3. Que vaut $P(Y \in [\ell_{i-1}, \ell_i[)$ pour $i \in [\![1,k]\!]$?
4. On propose de partir du code suivant :

```python
def loi_finie(valeurs,probabilites):
  points = np.cumsum(probabilites)
  y = ...
  for i in range(len(valeurs)):
    if ...:
      return valeurs[i]
```

   (a) À quoi sert l'instruction `np.cumpsum` ?
   (b) Quelle loi doit suivre Y ? Compléter la ligne `y = ...`.
   (c) À quelle condition doit-on renvoyer la valeur x_i ? Compléter la condition `if ...:` en conséquence.

2 Exercices

2.1 Méthode d'inversion

Exercice 5 - Approche théorique $\star\star\star$

Dans cet exercice, nous allons développer l'aspect théorique de la méthode d'inversion.
Soit X une variable aléatoire à densité. On suppose de plus que sa fonction de répartition F_X est strictement croissante sur $X(\Omega)$. Pour simplifier, on suppose que $X(\Omega) = [a,b]$ avec $a < b$.

1. Justifier que F_X est une bijection de $X(\Omega)$ dans $[0,1]$.
2. On note $Y = F_X(X)$. Montrer que Y est une variable aléatoire à densité. Quell est sa fonction de répartition ?
3. Soit $Z \hookrightarrow \mathcal{U}([0,1])$. En utilisant les questions précédentes, construire une nouvelle variable $f(Z)$ (f est à déterminer) telle que $f(Z)$ et X aient la même loi.
4. **Bonus :** Quel est le lien entre ce que nous venons de faire et le cas discret précédent ?

Exercice 6 - Loi exponentielle ⋆⋆

1. Rappeler une densité et la fonction de répartition de la loi exponentielle de paramètre μ.
2. En déduire une manière de compléter le code suivant :

```
def new_exponential(mu):
  y = rd.random()
  x = ...
  return x
```

afin qu'il simule le tirage d'une variable de loi exponentielle de paramètre μ.

Exercice 7 - Loi de Cauchy ⋆⋆⋆

La loi de Cauchy est la loi donnée par la densité définie sur $\mathbb{R}$ suivante : $f(x) = \frac{1}{\pi(1+x^2)}$.

1. Vérifier que la fonction donnée est bien une densité de probabilité.
2. Calculer la fonction de répartition associée.
3. Montrer alors que si $Y \hookrightarrow \mathcal{U}([0,1])$ alors $X = \tan\left[\pi\left(Y - \frac{1}{2}\right)\right]$ suit une loi de Cauchy.
4. Compléter le code `Python` suivant afin qu'il simule un tirage d'une variable suivant une loi de Cauchy :

```
def cauchy():
  y = rd.random()
  x = ...
  return x
```

2.2 Pour aller plus loin

Exercice 8 - Loi de Poisson ⋆⋆⋆

Dans le cas discret, nous n'avons utilisé la méthode d'inversion que pour des lois à univers image fini. On va essayer ici de généraliser la technique à des cas de variable discrète pouvant prendre une infinité de valeur en étudiant le cas de la loi de Poisson.

1. Chercher à placer des points sur le segment $[0,1]$ de manière analogue au cas fini de l'exercice 4 pour pouvoir simuler une loi de Poisson.
2. On part du code suivant :

```
def new_poisson(mu):
  y = rd.random()
  k = 0
  u = np.exp(-mu)
  s = u
  while s < y:
    k = k+1
    u = u*mu/k
    s = s+u
  return k
```

   (a) Que représente la variable `y` ?
   (b) Que représente la variable `u` ?
   (c) Que représente la variable `s` ?
   (d) À quoi sert la condition `s < y` dans la boucle ?

Exercice 9 ⋆⋆⋆

Simuler une loi uniforme $\mathcal{U}(\llbracket 1, n \rrbracket)$ par la méthode d'inversion sans utiliser de boucle.

Exercice 10 - Loi de Rayleigh ⋆⋆⋆⋆

Soit $\sigma > 0$ (comment se prononce cette lettre ?). On pose la fonction définie sur $\mathbb{R}_+$ par $f(x) = \frac{1}{\sigma^2} x e^{-\frac{x^2}{2\sigma}}$.

1. Montrer que f est une densité.
2. Donner la fonction de répartition F associée.
3. Montrer que F est une bijection de $\mathbb{R}_+^\star$ sur $]0, 1[$.
4. En déduire une fonction `Python` permettant de simuler la loi donnée par f en utilisant la méthode d'inversion.
5. X suit la loi de densité f, quelle est la loi de $Y = X^2$?

3 Travail à préparer pour le prochain TP

Exercice 11 - C'est sans espérance ⋆⋆⋆

On a vu dans le cours que la loi de Cauchy n'admet pas d'espérance (seriez-vous capable de le redémontrer ?). On va chercher ici à le visualiser.
On réutilise la fonction `cauchy` définie à l'exercice 7.

1. On part du code suivant :

```python
def serie_cauchy(n):
  A = np.zeros(n)
  for i in range(n):
    A[i] = cauchy()
  return A
```

   Que fait-il ?

2. On complète le code de la manière suivante :

```python
def serie_cauchy(n):
  A = np.zeros(n)
  for i in range(n):
    A[i] = cauchy()
  B = np.cumsum(A) / np.arange(1,n+1)
  return A,B
```

   (a) Rappeler ce que fait l'instruction `np.cumsum`.
   (b) Que fait l'opération `/` entre deux tableaux `numpy` ?
   (c) Que contient donc le tableau `B` à la fin de la fonction ?

3. On utilise la fonction précédente dans le code suivant :

```python
import matplotlib.pyplot as plt

serie, moyennes = serie_cauchy(10000)
plt.plot(moyennes)
plt.show()
```

   (a) Rappeler le rôle de `plt.plot` et `plt.show`.
   (b) Exécuter les instructions `Python` sur un ordinateur. Qu'observe-t-on ? Comment interpréter cela vis-à-vis de l'espérance de la loi de Cauchy ? Au besoin comparer avec le même graphe pour une loi exponentielle.