TP # 22

Variables aléatoires en Python (Part. 3)

(Variables aléatoires discrètes et loi des grands nombres)

Bibliothèque numpy.random

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

Lois usuelles : Binomiale, Uniforme, Géométrique, Poisson

- `rd.binomial(n,p)` génère un entier aléatoire selon la loi $\mathcal{B}(n,p)$.

  On peut rajouter un paramètre à cette instruction pour générer plusieurs valeurs :

  - `rd.binomial(n,p,m)` génère un vecteur ligne de taille `m` contenant des réalisations indépendantes.
  - `rd.binomial(n,p,(m1,m2))` génère un tableau à `m1` lignes et `m2` colonnes contenant des réalisations indépendantes.

- `rd.randint(a,b+1)` génère un entier aléatoire selon la loi $\mathcal{U}(\llbracket a,b \rrbracket)$.

  Même principe pour `rd.randint(a,b+1,m)` et `rd.randint(a,b+1,(m1,m2))`.

- `rd.geometric(p)` génère un entier aléatoire selon la loi $\mathcal{G}(p)$.

  Même principe pour `rd.geometric(p,m)` et `rd.geometric(p,(m1,m2))`.

- `rd.poisson(lam)` génère un entier aléatoire selon la loi $\mathcal{P}(\lambda)$. (avec $\lambda =$`lam`)

  Même principe pour `rd.poisson(lam,m)` et `rd.poisson(lam,(m1,m2))`.

Exercice 1

On rappelle qu'une variable $X \hookrightarrow \mathcal{G}(p)$ compte le nombre de répétitions d'épreuves de Bernoulli indépendantes de paramètre p pour obtenir un succès.

En utilisant `rd.random()`, proposer une fonction `geometrique(p)` (alternative à `rd.geometric(p)`) qui simule une variable aléatoire de loi géométrique.

```
def geometrique(p):
```

Exercice 2

Estimation de $E(X)$ par des moyennes empiriques (loi géométrique)

1. Compléter la fonction `moyenne(p,n)` pour qu'elle calcule la moyenne empirique $M_n = \dfrac{1}{n}\sum_{i=1}^{n} X_i$, où les X_i sont des valeurs aléatoires indépendantes générées selon la loi géométrique $\mathcal{G}(p)$.

```
def moyenne(p,n) :

    S = ....

    for i in range( ... ) :

        S = S + .......................

    return ........
```

Exemple : `moyenne(1/10, 10**5)` renvoie la valeur : ....

Ceci est cohérent puisque si $X \hookrightarrow \mathcal{G}(\frac{1}{10})$, $\quad E(X) =$ ....

2. Compléter la fonction `vecteur_moyenne(p,n)` pour renvoyer un vecteur contenant la suite des moyennes empiriques : `V = [` $M_1, M_2, \ldots, M_n$ `]` .

```
def vecteur_moyenne(p,n) :

    V = .................. ; S = .... ;

    for k in range(  ) :

        S = S + .......................

        V[k] = ................

    return V
```

Exemple : `vecteur_moyenne(1/10,10)` renvoie les moyennes empiriques :

[, , , , , , , , ,]

3. A la suite de la fonction précédente, compléter le script suivant pour afficher :
- La suite des moyennes empiriques $M_1, M_2, \ldots, M_N$
- La droite horizontale d'équation $y = \frac{1}{p}$ en pointillés rouges.

```
import matplotlib.pyplot as plt
p = 1/10 ; N = 200

X = .......................... ; Y = ..........................

plt.plot(X,Y)
plt.plot([ ... , ... ], [ ........ , ........ ], 'r--')
plt.show()
```

Tester plusieurs fois cette fonction et observer les graphes affichés pour les valeurs $N = 200$, $N = 500$, $N = 1000$ et enfin $N = 10000$.

On pourra également tester différentes valeurs de $p \in]0,1[$.

Allure du graphe pour $N = 10000$:

On constate ici la **loi des grands nombres** : les moyennes empiriques de variables aléatoires indépendantes générées selon la même loi convergent (en un certain sens...) vers l'espérance associée à cette loi :

$$\text{Si } X_1, \dots, X_n \text{ sont indépendantes, de loi } \mathcal{G}(p), \quad \frac{1}{n}\sum_{i=1}^{n} X_i \xrightarrow[n\to+\infty]{} E(X_1) = \frac{1}{p}$$

Exercice 3

Estimation de $E(X)$ par des moyennes empiriques (loi de Poisson)

Adapter le programme de l'exercice précédent pour que `vecteur_moyenne(lam,n)` renvoie la suite des moyennes empiriques d'une loi de Poisson $\mathcal{P}(\lambda)$ (où `lam`$= \lambda$).

```
def vecteur_moyenne(lam,n) :
    V = ................... ; S = .... ;
    for k in range(  ) :
        S = S + ........................
        V[k] = ................
    return V
```

Constater de nouveau, sur les graphes, la convergence des moyennes empiriques vers l'espérance d'une loi de Poisson. On adaptera en conséquence le script de la question 3. en on choisira par exemple $\lambda = 5$ et $N = 5000$.

Exercice 4

Et si X n'admet pas d'espérance ?

1. On considère $X \hookrightarrow \mathcal{G}(\frac{1}{2})$.
Justifier que la variable aléatoire $Y = 3^X$ n'admet pas d'espérance.

2. Adapter la fonction `vecteur_moyenne` précédente pour qu'elle renvoie la suite des moyennes empiriques `V = [` $M_1, M_2, \dots, M_n$ `]` avec cette fois-ci :

$$M_n = \frac{1}{n}\sum_{i=1}^{n} Y_i, \quad \text{où } Y_i = 3^{X_i} \text{ avec } X_i \hookrightarrow \mathcal{G}\left(\frac{1}{2}\right).$$

```
def vecteur_moyenne(n) :
    V = ................... ; S = .... ;
    for k in range(  ) :
        x = .................
        S = S + 3**x
        V[k] = ................
    return V
```

3. Afficher le graphe représentant $M_1, M_2, \dots, M_N$ (avec $N = 10^5$) et constater que cette fois les moyennes empiriques ne convergent pas !
On pourra tester plusieurs fois et observer la diversité des comportements affichés.

```
N = 10**5
X = ........................... ; Y = ...........................
plt.plot(X,Y) ; plt.show()
```