♦ TP12 – Approximation de la loi de Poisson par une loi binomiale

I. — Programmation

On fixe un réel $\lambda > 0$. Pour tout $n \in \mathbb{N}^*$, on considère une variable aléatoire X_n suivant une loi binomiale de paramètres n et $\dfrac{\lambda}{n}$. On désigne, de plus, par Y une variable aléatoire suivant une loi de Poisson de paramètre λ.

On peut démontrer (c'est le but du second paragraphe) que lorsque n devient grand, la variable X_n est une bonne approximation de Y. Autrement dit, lorsque n devient grand, on peut approcher la loi de Poisson $\mathscr{P}(\lambda)$ par la loi binomiale $\mathscr{B}\left(n, \dfrac{\lambda}{n}\right)$.

1. En utilisant la fonction `random` du module `random`, écrire une fonction `bernoulli` prenant en argument un réel `p` compris entre 0 et 1 et qui simule une variable aléatoire suivant une loi de Bernoulli de paramètre `p`.

Solution.

```
from random import random

def bernoulli(p):
  if random()<p:
    return 1
  else:
    return 0
```

2. En utilisant la fonction `bernoulli`, écrire une fonction `binomiale` prenant en argument un entier naturel `n` non nul et un réel `p` compris entre 0 et 1 et qui simule une variable aléatoire suivant une loi binomiale de paramètres `n` et `p`.

Solution.

```
def binomiale(n,p):
  S=0
  for k in range(n):
    S += bernoulli(p)
  return S
```

3. En déduire une fonction `approx_Poisson` qui prend en arguments un réel `lamb` strictement positif et un entier naturel `n` et qui simule une variable aléatoire dont la loi est approximativement une loi de Poisson pour `n` assez grand.

Solution.

```
def approx_Poisson(lamb,n):
  return binomiale(n,lamb/n)
```

4. Écrire une fonction `liste_approx_Poisson` qui prend en arguments un entier naturel `N` non nul, un réel strictement positif `lamb` et un entier naturel `n` et qui renvoie une liste de `N` résultats renvoyés par `approx_Poisson(lamb,n)`.

Solution.

```
def liste_approx_Poisson(N,lamb,n):
  return [ approx_Poisson(lamb,n) for i in range(N) ]
```

5. Écrire une fonction `frequence` prenant en arguments un nombre entier naturel `k` et une liste d'entiers naturels `L` et qui renvoie la fréquence d'apparition de `k` dans `L` i.e. le nombre d'occurrences de `k` dans la liste `L` divisé par le nombre total d'éléments de la liste.

Solution.

```
def frequence(k,L):
  c = 0
  for e in L:
    if e == k:
      c += 1
  return c/len(L)
```

6. Le tableau ci-dessous donne la valeur arrondie à 10^{-3} près de $\mathbf{P}(Y = k)$ pour différentes valeurs de λ et de k.

k \ λ	0,5	1	1,5	2	2,5	3	4	5	6
0	0,607	0,368	0,223	0,135	0,082	0,050	0,018	0,007	0,002
1	0,303	0,368	0,335	0,271	0,205	0,149	0,073	0,037	0,015
2	0,076	0,184	0,251	0,271	0,257	0,224	0,147	0,084	0,045
3	0,013	0,061	0,126	0,180	0,214	0,224	0,195	0,140	0,089
4	0,002	0,015	0,047	0,090	0,134	0,168	0,195	0,175	0,134
5	0,000	0,003	0,014	0,036	0,067	0,101	0,156	0,175	0,161
10	0,000	0,000	0,000	0,000	0,000	0,001	0,005	0,018	0,041

Pour quelques valeurs de k et de λ, comparer ces valeurs avec les résultats renvoyés, pour différentes valeurs de `n`, par `frequence(k,liste_approx_Poisson(1000,lamb,n))` lorsque `k` prend la valeur k et `lamb` la valeur λ.

Solution.

```
print(frequence(3,liste_approx_Poisson(1000, 2, 100)))
0.189
print(frequence(3,liste_approx_Poisson(1000, 2, 1000)))
0.182
print(frequence(3,liste_approx_Poisson(1000, 2, 10000)))
0.177
print(frequence(5,liste_approx_Poisson(1000, 1, 100)))
0.001
print(frequence(1,liste_approx_Poisson(1000, 5, 10000)))
0.04
print(frequence(2,liste_approx_Poisson(1000, 1.5, 10000)))
0.251
print(frequence(4,liste_approx_Poisson(1000, 6, 10000)))
0.131
```

Les valeurs sont en effet proches de celles du tableau.

II. — Étude mathématique

On fixe un réel $\lambda > 0$ et un entier $k \in \mathbb{N}$. Pour tout $n \geqslant k$, on considère une variable aléatoire X_n suivant une loi binomiale de paramètres n et $\dfrac{\lambda}{n}$. On désigne, de plus, par Y une variable aléatoire suivant une loi de Poisson de paramètre λ.

1. Soit un entier $n \geqslant k$. Rappeler les valeurs de $\mathbf{P}(X_n = k)$ et $\mathbf{P}(Y = k)$.

Solution. Par définition, $\mathbf{P}(X_n = k) = \binom{n}{k}\left(\dfrac{\lambda}{n}\right)^k\left(1-\dfrac{\lambda}{n}\right)^{n-k}$ et $\mathbf{P}(Y=k) = \dfrac{\lambda^k}{k!}\mathrm{e}^{-\lambda}$.

2. Soit un entier $n \geqslant k$. Rappeler les valeurs de $\mathbf{E}(X_n)$, $\mathbf{V}(X_n)$, $\mathbf{E}(Y)$ et $\mathbf{V}(Y)$ puis vérifier que $\mathbf{E}(X_n) \xrightarrow[n\to+\infty]{} \mathbf{E}(Y)$ et $\mathbf{V}(X_n) \xrightarrow[n\to+\infty]{} \mathbf{V}(Y)$.

Solution. Par propriété, $\mathbf{E}(X_n) = n \times \dfrac{\lambda}{n} = \lambda$ et $\mathbf{E}(Y) = \lambda$ donc $\mathbf{E}(X_n) \xrightarrow[n\to+\infty]{} \mathbf{E}(Y)$.

De même, $\mathbf{V}(X_n) = n\left(\dfrac{\lambda}{n}\right)\left(1-\dfrac{\lambda}{n}\right) = \lambda\left(1-\dfrac{\lambda}{n}\right) \xrightarrow[n\to+\infty]{} \lambda$ et $\mathbf{V}(Y) = \lambda$ donc $\mathbf{V}(X_n) \xrightarrow[n\to+\infty]{} \mathbf{V}(Y)$.

3. Montrer que, lorsque n tend vers $+\infty$, $\binom{n}{k} \sim \dfrac{n^k}{k!}$.

Solution. Par définition,

$$\binom{n}{k} = \frac{n!}{k!(n-k)!} = \frac{n(n-1)(n-2)\cdots(n-k+1)}{k!}.$$

Or, pour tout $i \in [\![0, k-1]\!]$, $n-i \sim n$ donc, par produit, $n(n-1)(n-2)\cdots(n-k+1) \sim n^k$ et finalement $\binom{n}{k} \sim \dfrac{n^k}{k!}$.

4. Soit un entier $n \geqslant k$. Écrire $\left(1-\dfrac{\lambda}{n}\right)^{n-k}$ à l'aide de la fonction exponentielle.

Solution. Par propriété, $\left(1-\dfrac{\lambda}{n}\right)^{n-k} = \mathrm{e}^{(n-k)\ln\left(1-\frac{\lambda}{n}\right)}$.

5. Déduire de la question précédente que $\lim\limits_{n\to+\infty}\left(1-\dfrac{\lambda}{n}\right)^{n-k} = \mathrm{e}^{-\lambda}$.

Solution. Comme $\lim\limits_{n\to+\infty}\dfrac{\lambda}{n} = 0$, $\ln\left(1-\dfrac{\lambda}{n}\right) \sim -\dfrac{\lambda}{n}$ donc

$$(n-k)\ln\left(1-\frac{\lambda}{n}\right) \sim -(n-k)\frac{\lambda}{n} \sim -n\frac{\lambda}{n} \sim -\lambda.$$

Ainsi, $\lim\limits_{n\to+\infty}(n-k)\ln\left(1-\frac{\lambda}{n}\right) = -\lambda$ donc, par continuité de exp sur $\mathbb{R}$, $\lim\limits_{n\to+\infty}\left(1-\dfrac{\lambda}{n}\right)^{n-k} = \mathrm{e}^{-\lambda}$.

6. Conclure que $\mathbf{P}(X_n = k) \xrightarrow[n\to+\infty]{} \mathbf{P}(Y=k)$.

Solution. Comme $\lim\limits_{n\to+\infty}\left(1-\dfrac{\lambda}{n}\right)^{n-k} = \mathrm{e}^{-\lambda} \neq 0$, $\left(1-\dfrac{\lambda}{n}\right)^{n-k} \sim \mathrm{e}^{-\lambda}$ donc

$$\mathbf{P}(X_n = k) = \binom{n}{k}\left(\frac{\lambda}{k}\right)^n\left(1-\frac{\lambda}{n}\right)^{n-k} \sim \frac{n^k}{k!} \times \frac{\lambda^k}{n^k} \times \mathrm{e}^{-\lambda} \sim \frac{\lambda^k}{k!}\mathrm{e}^{-k} = \mathbf{P}(Y=k)$$

donc $\mathbf{P}(X_n = k) \xrightarrow[n\to+\infty]{} \mathbf{P}(Y=k)$.