Lycée Pierre-Gilles de Gennes 2024-2025

TP8 Python

Convergence en loi et TCL

1 Commandes utiles

1.1 Simulations de lois usuelles

En important le module `numpy.random` avec la commande :

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

on peut simuler les lois usuelles :

- `rd.exponential(lambda,n)` : simule un n échantillon de loi $\mathcal{E}(\lambda^{-1})$,
- `rd.normal(mu,sigma,n)` : simule un n échantillon de loi $\mathcal{N}(\texttt{mu},\texttt{sigma}^2)$,
- `rd.poisson(lambda,n)` : simule un n échantillon de loi $\mathcal{P}(\lambda)$,
- `rd.binomial(N,p,n)` : simule un n échantillon de loi $\mathcal{B}(N,p)$,
- ...

1.2 Histogramme et diagramme en barre

En important le module `matplotlib.pyplot` avec la commande :

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

la commande

```
plt.hist(L,n, density = True)
```

permet de tracer l'histogramme des fréquences de la liste `L` avec n barres : l'axe des abscisses est divisé en n intervalles $[x_k, x_{k+1}[$; les barres de l'histogramme sont les rectangles de base $[x_k, x_{k+1}[$ et dont la hauteur est la proportion d'éléments de L compris dans l'intervalle $[x_k, x_{k+1}[$.

La commande

```
plt.bar(X,Y)
```

permet de tracer le diagramme en barres de Y en fonction de X : pour chaque abscisse $X[i]$ on trace la barre de hauteur $Y[i]$.

2 Convergence en loi

2.1 Exemples de convergence en loi

Voir exercice 4 du TD 15.

Exercice 1. Soit $(X_n)_{n\in\mathbb{N}^*}$ une suite de variables aléatoires indépendantes de loi $\mathcal{U}([0,1])$. On note, pour tout $n \in \mathbb{N}$, $U_n = \max(X_1,\dots,X_n)$ et $V_n = \min(X_1,\dots,X_n)$.

1. (a) Écrire une fonction `Min(n)` qui simule la variable V_n.
   (b) Écrire une fonction `repartition_Min(x,n,N)` qui simule N fois la variable V_n et renvoie la proportion de fois où $[V_n \leq x]$ est réalisé.
   (c) Expliquer en détail pourquoi `repartition_Min(x,n,N)` est une valeur approchée de $F_{V_n}(x)$.
   (d) Pour $n = 100$ et $N = 1000$, tracer $x \mapsto$ `repartition_Min(x,n,N)` sur $[-5, 5]$. Que peut-on conjecturer sur la convergence en loi de la suite (V_n) ?
2. Montrer que $(\min(X_1, \ldots, X_n))_{n\in\mathbb{N}^*}$ converge en loi vers une variable aléatoire suivant une loi certaine que l'on précisera.
3. Reprendre l'exercice avec U_n.

Exercice 2. Pour tout $n \in \mathbb{N}$, on pose $Y_n = n(1 - U_n)$ où U_n est défini dans l'exercice précédent.

1. (a) Écrire une fonction `Y(n)` qui simule la variable Y_n puis une fonction `repartition_Y(x,n,N)` qui simule N fois la variable Y_n et renvoie une valeur approchée de $F_{Y_n}(x)$.
   (b) Pour $n = 100$ et $N = 1000$, tracer $x \mapsto$ `repartition_Y(x,n,N)` sur $[-5, 5]$ et comparer avec la fonction de répartition d'une loi $\mathcal{E}(1)$.
   (c) Que peut-on conjecturer sur la convergence en loi de la suite (Y_n) ?
2. Montrer que $(Y_n)_{n\in\mathbb{N}^*}$ converge en loi vers une variable aléatoire suivant une loi que l'on précisera.

2.2 Binomiales et Poisson

Exercice 3.

1. Écrire une fonction `Poisson(lambda_,n)` qui renvoie la liste `[`$\mathbb{P}(X = k)$ `for k in range(n+1)]` où X suit la loi de Poisson de paramètre `lambda_`.
2. Écrire une fonction similaire pour une loi $\mathcal{B}(n, p)$.
3. Pour $n = 30$, tracer le diagramme en barres de la loi de Poisson de paramètre 2 et celui de la loi $\mathcal{B}\left(n, \frac{2}{n}\right)$.
4. Que constate-t-on ? Quel résultat de cours cela illustre ?

3 Estimation du paramètre d'une Bernoulli

On dispose d'une urne contenant un certain nombre n de boules noires et b de boules blanches.
Les valeurs de n et b sont inconnues et on cherche à déterminer la proportion de boules noires.
Recopier les programmes suivants pour obtenir des valeurs de n et b inconnues :

```
nb_noires = rd.randint(2,15)
nb_blanches = rd.randint(2,15)
```

Travail demandé

1. Soit $\varepsilon > 0$ et soit $(X_n)_n$ une suite de variable aléatoire de Bernoulli indépendantes de paramètre $p \in]0,1[$. Montrer, à l'aide de l'inégalité de Bienaymée-Tchebychev que pour tout $n \in \mathbb{N}^*$:

$$\mathbb{P}(|M_n - p| > \varepsilon) \leq \frac{p(1-p)}{n\varepsilon^2} \leq \frac{1}{4n\varepsilon^2}.$$

2. En déduire que pour tout $n \in \mathbb{N}^*$:

$$\mathbb{P}(p \in [M_n - \varepsilon, M_n + \varepsilon]) \geq 1 - \frac{1}{4n\varepsilon^2}.$$

3. Écrire une fonction `nombre_tirage(p,alpha,eps)` qui renvoie un entier n pour lequel

$$\mathbb{P}(p \in [M_n - \texttt{eps}, M_n + \texttt{eps}]) \geq 1 - \alpha.$$

   L'intervalle $[M_n - \texttt{eps}, M_n + \texttt{eps}]$ est alors appelé un intervalle de confiance pour p au niveau de confiance $1 - \alpha$.

4. On considère la fonction suivante à recopier :

```python
def tirage(N):
    s = 0
    for k in range(N):
        if rd.rand()<nb_noires/(nb_noires+nb_blanches):
            s+=1
    return s/N
```

   Expliquer ce que fait cette fonction.

5. Proposer un protocole expérimental permettant avec une probabilité d'au moins 95%, d'obtenir une valeur approchée à ± 0.01 près de la proportion de boules noires. Programmer ce protocole et comparer votre estimation avec la valeur exacte `nb_noires/(nb_noires+nb_blanches)`.