Informatique

TP 3

Simulation des lois discrètes usuelles

Partie 1 : principe et premiers exemples

Exercice 1 : Diantre !

En guise d'introduction au générateur `rand()` de nombres pseudo-aléatoires fourni par Scilab, taper dans la console les lignes puis comparer les résultats renvoyés avec ceux des voisins :

```
-->rand()
-->rand()
```

Scilab propose deux générateurs de nombres pseudo-aléatoires.

- L'instruction `rand()` simule la loi uniforme sur l'intervalle $[0,1]$. Comme on l'a mis en évidence dans le premier exercice, ce générateur doit être initialisé par les lignes suivantes avant d'être utilisé :

  ```
  rand('seed', getdate('s'))
  rand('uniform')
  ```

  L'objectif de ce TP est de simuler les lois discrètes usuelles à l'aide de ce seul générateur de nombres aléatoires.
  L'instruction `rand(m,n)` renvoie une matrice $m \times n$ dont les coefficients sont des simulations indépendantes de loi uniforme sur $[0,1]$.
- L'instruction `grand(m,n,...)` renvoie une matrice $m \times n$ dont les coefficients sont des simulations indépendantes d'une loi donnée (on renvoie à l'aide de Scilab pour davantage de détails) :
  - `grand(m,n,'uin',a,b)` pour la loi uniforme sur l'intervalle d'entiers $[\![a,b]\!]$;
  - `grand(m,n,'bin',N,p)` pour la loi binomiale $\mathcal{B}(\mathrm{N},p)$;
  - `grand(m,n,'geom',p)` pour la loi géométrique $\mathcal{G}(p)$;
  - `grand(m,n,'poi',mu)` pour la loi de Poisson d'espérance μ.

  Ce deuxième générateur ne sera pas utilisé au cours de ce TP.

Exercice 2 : loi de Bernoulli

Dans un même script, implémenter les fonctions et commandes répondant aux questions ci-dessous.

1. Écrire une fonction `y=simul_bernoulli(p)` renvoyant le résultat d'une simulation de la loi de Bernoulli de paramètre p.
2. Pour tester l'adéquation de cette fonction au modèle théorique, déterminer la proportion de 1 obtenus au cours de n simulations et la comparer à la probabilité p (on prendra par exemple $p = \frac{1}{4}$ et on fera des essais avec différentes valeurs de n).

À partir d'une matrice X de données et d'un vecteur $\mathrm{S} = (s_1, \dots, s_n)$ tel que $s_1 < \cdots < s_n$, la commande `[I,E]=dsearch(X,S,'d')` renvoie :

- un vecteur E de même format que S où, pour tout $k \in [\![1,n]\!]$, E_k est le nombre de coefficients de la matrice X égaux à s_k ;
- une matrice I de même format que X, dont l'élément générique est l'indice $k \in [\![1,n]\!]$ tel que l'élément correspondant de X s'écrive $x = s_k$, s'il existe, et vaut 0 sinon.

Exercice 3 : loi uniforme

Dans un même script, implémenter les fonctions et commandes répondant aux questions ci-dessous.

1. Écrire une fonction `y=simul_uniforme(n)` renvoyant le résultat d'une simulation de la loi uniforme sur l'intervalle d'entiers $[\![1, n]\!]$ (pour $n \geqslant 1$ entier).
2. Pour tester l'adéquation de cette fonction au modèle théorique, superposer sur une même figure les diagrammes en bâtons de la distribution statistique observée lors d'une succession de N simulations et de la distribution de probabilités attendue (on prendra par exemple $n = 10$ et on fera des essais avec différentes valeurs de N).

Partie 2 : simulation par l'expérience-type dont la loi est issue

Exercice 4 : loi binomiale

1. Rappeler l'expérience-type associée à la loi binomiale de paramètre (n, p). En déduire une fonction simulant une variable aléatoire de loi binomiale. Confronter cette simulation au modèle théorique.
2. Écrire une autre version de la fonction de simulation utilisant les fonctionnalités matricielles de Scilab.

Exercice 5 : loi géométrique

Écrire une fonction simulant une variable aléatoire de loi géométrique de paramètre $p \in \,]0, 1[$. Confronter cette simulation au modèle théorique.

Exercice 6 : loi hypergéométrique (hors-programme)

Pour $\mathrm{N}, n \in \mathbb{N}$ tels que $n \leqslant \mathrm{N}$ et $p \in \,]0, 1[$, la loi *hypergéométrique* $\mathcal{H}(\mathrm{N}, n, p)$ est celle du nombre de boules blanches obtenues lors de n tirages successifs sans remise dans une urne contenant initialement N boules dont une proportion p de boules blanches.

1. Écrire une fonction `simul` qui simule la loi $\mathcal{H}(\mathrm{N}, n, p)$ en suivant la composition de l'urne tout au long de l'expérience. On passera en troisième paramètre l'entier $\mathrm{N}p$ plutôt que p.
2. Expliciter la loi $\mathcal{H}(\mathrm{N}, n, p)$.

Exercice 7 : loi de Pascal (hors-programme)

La loi de Pascal $\mathcal{P}(r, p)$ de paramètres $r \in \mathbb{N}^*$ et $p \in \,]0, 1[$ est celle du temps d'attente du r-ième succès lors d'une répétition d'expériences de Bernoulli indépendantes de même paramètre p.
Simuler cette loi et montrer que si X est une variable aléatoire de loi $\mathcal{P}(r, p)$, alors elle prend ses valeurs dans $[\![r, +\infty[\![$ avec :

$$\forall k \in [\![r, +\infty[\![, \quad \mathbb{P}(\mathrm{X} = k) = \binom{k-1}{r-1} p^r q^{k-r},$$

admet une espérance et une variance données par :

$$\mathbb{E}(\mathrm{X}) = \frac{r}{p} \quad \text{et} \quad \mathbb{V}(\mathrm{X}) = \frac{rq}{p^2}.$$

Partie 3 : utilisation de la fonction de répartition

On suppose que X est une variable aléatoire réelle discrète prenant ses valeurs dans l'ensemble $\{x_n\}_{n\in\mathbb{N}}$, où $(x_n)_{n\in\mathbb{N}}$ est une suite strictement croissante de réels. On note $p_n = \mathbb{P}(\mathrm{X} = x_n)$ pour tout $n \in \mathbb{N}$.
La fonction de répartition d'une telle variable aléatoire est alors donnée par :

$$\mathrm{F_X} : x \longmapsto \mathbb{P}(\mathrm{X} \leqslant x) = \sum_{\substack{n\in\mathbb{N}\,:\\ x_n\leqslant x}} p_n.$$

On note en particulier $\mathrm{F}_{-1} = 0$ et

$$\forall n \in \mathbb{N}, \quad \mathrm{F}_n = \mathrm{F_X}(x_n) = \sum_{k=0}^{n} p_k.$$

On définit la *fonction quantile* de X sur $]0,1[$ par :

$$\mathrm{Q_X} : u \in\,]0,1[\, \longmapsto \sup\{x \in \mathbb{R} : \mathrm{F_X}(x) < u\}.$$

On considère également une variable aléatoire U de loi uniforme sur $[0,1]$. On rappelle que cela signifie que pour tous $a < b \in\,]0,1[$, on a $\mathbb{P}(\mathrm{U} \in\,]a,b]) = b - a$.

Exercice 8

1. Pour $u \in\,]0,1[$, justifier que le réel $\mathrm{Q_X}(u)$ est bien défini.
2. Représenter graphiquement la fonction de répartition $\mathrm{F_X}$. On placera sur l'axe des abscisses les points x_{n-1}, x_n et x_{n+1} et sur celui des ordonnées les points F_{n-2}, F_{n-1}, F_n et F_{n+1}.
3. Préciser le comportement de la fonction $\mathrm{Q_X}$ sur l'intervalle $]\mathrm{F}_{n-1}, \mathrm{F}_n]$. Quelles sont les valeurs prises par la fonction $\mathrm{Q_X}$?
4. Montrer que :
$$\forall n \in \mathbb{N}, \quad \mathbb{P}(\mathrm{F}_{n-1} < \mathrm{U} \leqslant \mathrm{F}_n) = p_n.$$
5. Justifier que la variable aléatoire $\mathrm{Q_X}(\mathrm{U})$ suit la même loi que X.

Exercice 9 : loi de Poisson

En déduire une fonction `y=simul_poisson(lambda)` réalisant la simulation d'une variable aléatoire de loi de Poisson $\mathcal{P}(\lambda)$ de paramètre $\lambda > 0$. Confronter cette simulation au modèle théorique.