TP 4

Simulation de variables aléatoires discrètes

Compétences attendues.

✓ Savoir simuler une loi discrète usuelle à l'aide de la fonction `grand`, ou uniquement à partir de la fonction `rand()`.

✓ Vérifier graphiquement la pertinence d'une simulation d'une loi.

Objectifs.

- On souhaite *simuler* une loi donnée, c'est-à-dire créer une fonction `Scilab` qui, à chaque appel, renvoie une réalisation d'une variable aléatoire X suivant cette loi.

  Exemple : Simuler la loi $\mathscr{U}(\llbracket 1,6 \rrbracket)$, c'est créer une fonction `Scilab` qui renvoie les valeurs 1,2,3,4,5,6, chacune d'entre elles apparaissant avec une fréquence égale à $1/6$.

- On souhaite créer une fonction `Scilab` qui renvoie un *échantillon observé de taille N* de la loi, c'est-à-dire un vecteur de taille N contenant les réalisations de variables aléatoires $X_1, \dots, X_N$ mutuellement indépendantes et suivant toutes cette loi.

- On souhaite vérifier graphiquement la qualité des simulations obtenues.

Mathieu Mansuy - Professeur de Mathématiques en ECS2 au Lycée Louis Pergaud (Besançon)
`mansuy.mathieu@hotmail.fr`

1 Simulation des lois discrètes usuelles

1.1 Fonctions Scilab

`Scilab` dispose d'une fonction simulant les lois discrètes usuelles : la fonction `grand`.

Définition.

- `grand(N,M,"loi",paramètres)` renvoie une matrice de taille $N \times M$ contenant des réalisations de variables aléatoires indépendantes suivant cette loi.
- `grand(N,M,"uin",n1,n2)` simule la loi $\mathscr{U}(\llbracket n1, n2 \rrbracket)$
- `grand(N,M,"bin",n,p)` simule la loi $\mathscr{B}(n,p)$
- `grand(N,M,"geom",p)` simule la loi $\mathscr{G}(p)$
- `grand(N,M,"poi",lambda)` simule la loi $\mathscr{P}(\lambda)$

Exemple. Exécuter plusieurs fois l'instruction suivante :

```
--> grand(1,10,"uin",1,6)
```

Dans la suite de cette partie, on explique comment simuler les lois usuelles en utilisant uniquement la fonction `rand`.

Définition.

`rand()` renvoie une réalisation d'une variable aléatoire suivant la loi $\mathscr{U}([0,1[)$.

`rand(N,M)` renvoie une matrice de taille $N \times M$ contenant des réalisations de variables aléatoires indépendantes suivant la loi $\mathscr{U}([0,1])$.

1.2 Loi uniforme

Exercice 1

Soient n et m deux entiers tels que $n \leq m$. On rappelle que si $U \hookrightarrow \mathscr{U}([0,1])$, alors on a $V = n + \lfloor (m-n+1)U \rfloor \hookrightarrow \mathscr{U}(\llbracket n, m \rrbracket)$.

1. Écrire une fonction `uniforme(n,m)` simulant la loi $\mathscr{U}(\llbracket n, m \rrbracket)$.
2. Écrire une fonction `Uniforme(N,n,m)` renvoyant un vecteur contenant N réalisations indépendantes de la loi $\mathscr{U}(\llbracket n, m \rrbracket)$.

1.3 Loi de Bernoulli

Exercice 2

1. Montrer que si $U \hookrightarrow \mathscr{U}([0,1])$ et $p \in [0,1]$, alors $P(U \leq p) = p$ et $P(U > p) = 1 - p$.
2. Compléter la fonction suivante afin qu'elle simule une loi de Bernoulli de paramètre p.

```
function y = bernoulli(p)
   u = rand() ;
   y = 0 ;
   if u ... then
      y = ...
   end
```

```
endfunction
```

3. Écrire une fonction `Bernoulli(N,p)` renvoyant un vecteur contenant N réalisations indépendantes de la loi $\mathscr{B}(p)$.

1.4 Loi binomiale

Propriété 1

Soit $n \in \mathbb{N}^*$. Soit $p \in [0,1]$.
Si $X_1, \ldots, X_n$ sont n variables aléatoires mutuellement indépendantes suivant toutes la loi $\mathscr{B}(p)$, alors $\sum_{i=1}^{n} X_i \hookrightarrow \mathscr{B}(n,p)$.

Exercice 3

1. Écrire une fonction `binomiale(n,p)` simulant la loi $\mathscr{B}(n,p)$.

2. Écrire une fonction `Binomiale(N,n,p)` donnant un échantillon de taille N de la loi $\mathscr{B}(n,p)$

1.5 Loi géométrique

Propriété 2

Soit $p \in [0,1]$.
Si X est le rang du premier succès lors de la répétition d'épreuves de Bernoulli indépendantes, toutes de probabilité de succès p, alors $X \hookrightarrow \mathscr{G}(p)$.

Exercice 4

1. Écrire une fonction `geom(p)` simulant la loi $\mathscr{G}(p)$ en utilisant la fonction `bernoulli` et une boucle `while`.

2. Écrire une fonction `Geom(N,p)` afin de simuler un échantillon de taille N de la loi $\mathscr{G}(p)$.

   Simuler un échantillon de taille $N = 10000$ de la loi $\mathscr{G}(0.2)$, puis vérifier que la valeur moyenne de cet échantillon est cohérente avec ce qu'on attend.

2 Représentations graphiques

2.1 Comparaison diagramme en bâtons des fréquences / probabilités théoriques

Soit X une variable aléatoire discrète. Supposons qu'on dispose d'une fonction `Loi` permettant de simuler la loi de X. Pour juger de la qualité des simulations, on peut utiliser des représentations graphiques. Pour cela, on procèdera comme suit :

- On crée un échantillon de taille N c'est-à-dire un vecteur ligne `x` contenant N réalisations de la fonction `Loi`.

- On compare alors graphiquement les fréquences empiriques obtenues (c'est à dire grâce à l'échantillon) avec les probabilités théoriques pour vérifier la pertinence de la simulation.

Dans le cas de simulation de variables aléatoires discrètes, on va comparer plus particulièrement :

- le *diagramme en bâtons des fréquences* de notre échantillon de taille N ;
- les *probabilités théoriques* $P(X = k)$ pour $k \in X(\Omega)$.

Si notre simulation est bonne, on doit constater que :

Théorème 3 (*Théorème d'Or de Bernoulli*)

Pour N « suffisamment grand », les fréquences observées doivent être proches des probabilités théoriques.

On a besoin pour cela des commandes suivantes.

Définition.

- Si `x` est un vecteur, `y=tabul(x,"i")` est une matrice à deux colonnes, la première contenant les valeurs prises par les composantes de `x` rangées dans l'ordre croissant (décroissant sans `"i"`) et la seconde contenant le nombre d'occurrences de chaque valeur.
- Si `x` et `y` sont des vecteurs, `bar(x,y)` trace un diagramme en bâtons d'abcisse `x`, d'ordonnée `y`.

Méthode.
Pour tracer le diagramme des fréquences d'un échantillon `x`, on procède ainsi :

```
y=tabul(x,"i")
bar(y(:,1),y(:,2)/length(x))
```

Remarque. Pour réaliser plusieurs graphiques dans une même fenêtre et ainsi pouvoir mieux les comparer, on peut utiliser l'instruction `subplot(1,m,k)` avant chaque instruction de tracé de graphique, qui découpe la fenêtre graphique en m sous-graphiques, k indiquant le placement souhaité pour chaque graphique.

2.2 Exemples

Exercice 5

On se donne le code suivant :

```
//Echantillon
x = Uniforme(100000,1,6) ;

//Diag en batons des fréquences
y = tabul(x,"i") ;
subplot(1,2,1) ;
bar(y(:,1),y(:,2)/length(x)) ;

//Diag en batons des proba theoriques
u = (1/6)*ones(1,6)
subplot(1,2,2) ;
bar(1:6,u) ;
```

Exécuter ce code et interpréter le graphique ainsi obtenu. En particulier, à quoi correspond le vecteur u ?

Exercice 6 (★ - Simulation de la loi binomiale)

1. À quoi correspond le vecteur `c=[1,cumprod([n:-1:1]./[1:n])]` ? Expliquer.

2. À l'aide d'un diagramme en bâtons, tester la simulation de la loi $\mathscr{B}(10, 0.2)$ obtenue à l'aide de la fonction `Binomiale`.

Exercice 7 (★★ - Simulation d'une loi géométrique à partir d'une loi exponentielle)
Soit $\lambda \in]0, +\infty[$. On a montré en TD que si $X \hookrightarrow \mathscr{E}(\lambda)$, alors $\lfloor X \rfloor + 1 \hookrightarrow \mathscr{G}(1 - e^{-\lambda})$.

1. Écrire une fonction `Geom2(N,p)` simulant un échantillon de taille N de la loi $\mathscr{G}(p)$. On utilisera pour cela la fonction `grand(1,N,"exp",1/λ)` pour simuler un échantillon de taille N de la loi exponentielle $\mathscr{E}(\lambda)$.

2. À l'aide d'un diagramme en bâtons, juger de la qualité de cette simulation.

---

3 Méthode d'inversion discrète

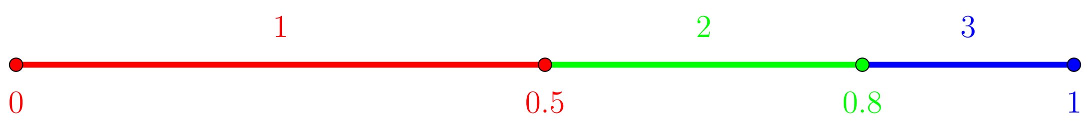
3.1 Principe

La méthode d'inversion est une méthode générale pour simuler la loi d'une variable discrète X à partir d'une variable aléatoire $U \hookrightarrow \mathscr{U}([0, 1])$. Elle consiste à « inverser » la fonction de répartition de X.

Exemple. On souhaite simuler une variable aléatoire X ayant pour support $X(\Omega) = \{1, 2, 3\}$ et pour probabilités ponctuelles :

$$P(X = 1) = 0.5, \quad P(X = 2) = 0.3 \quad \text{et} \quad P(X = 3) = 0.2.$$

On va pour cela découper l'intervalle $[0, 1]$ en trois intervalles de longueurs 0.5, 0.3 et 0.2.



Pour simuler la variable X, on tire alors au hasard un nombre t de $[0, 1]$ avec la fonction `rand()` et on retourne :

- $x = 1$ si $t \in [0, 0.5]$, ce qui se produit avec la probabilité 0.5 ;
- $x = 2$ si $t \in]0.5, 0.8]$, ce qui se produit avec la probabilité 0.3 ;
- $x = 3$ si $t \in]0.8, 1]$, ce qui se produit avec la probabilité 0.2.

On peut ainsi simuler la loi de la variable X à l'aide de la fonction suivante :

```
function x = simulation()
t = rand()
x = 3
if t <=0.5 then
      x = 1
```

```
        else if t <=0.8 then
              x = 2
             end
end
endfunction
```

Exercice 8 (★)

On souhaite simuler une loi uniforme $\mathscr{U}(\llbracket 1,6 \rrbracket)$.

1. Déterminer le découpage correspondant de l'intervalle $[0,1]$ pour cette loi.
2. Écrire une fonction `unif()` simulant la loi $\mathscr{U}(\llbracket 1,6 \rrbracket)$ à l'aide de la méthode d'inversion.

Cas général. Soit à présent X une variable discrète qui prend les valeurs $x_1 < x_2 < \cdots < x_n < \ldots$ Rappelons que sa fonction de répartition est en escalier, donnée par :

$$\forall x \in \mathbb{R}, \quad F(x) = P(X \leq x) \begin{cases} 0 & \text{si } x < x_1, \\ \displaystyle\sum_{i=1}^{n} P(X = x_i) & \text{si } x_n \leq x < x_{n+1}. \end{cases}$$

La méthode d'inversion présentée sur les exemples précédents se généralise de la manière suivante.

Méthode.
Pour simuler la variable discrète X, on procèdera comme suit :

- on choisit un paramètre t de manière aléatoire sur $[0,1]$ à l'aide de la fonction `rand()` ;
- on détermine l'intervalle $I_k =]F(x_{k-1}), F(x_k)]$ tel que $t \in I_k$ (si $k=1$, $I_1 = [0, F(x_1)]$) ;
- on retourne x_k.

Remarque. Soit $k \geq 2$. Le programme retourne x_k si et seulement si $t \in]F(x_{k-1}), F(x_k)]$, où t est une réalisation d'une variable $U \hookrightarrow \mathscr{U}([0,1])$. Or ceci se réalise avec une probabilité :

$$P(F(x_{k-1}) < U \leq F(x_k)) = P(U \leq F(x_k)) - P(U \leq F(x_{k-1})) = F_U(F(x_k)) - F_U(F(x_{k-1}).$$

Comme de plus $F_U(x) = x$ pour tout $x \in [0,1]$, et que $F(x_k), F(x_{k-1}) \in [0,1]$, on obtient :

$$P(F(x_{k-1}) < U \leq F(x_k)) = F(x_k) - F(x_{k-1}) = \sum_{i=1}^{k} P(X = x_i) - \sum_{i=1}^{k-1} P(X = x_i) = P(X = x_k).$$

Le même argument vaut aussi pour $k=1$. Ainsi la probabilité qu'un tel programme retourne x_k est bien $P(X = x_k)$.

3.2 Exemples

Exercice 9 (★★ - Simulation de la loi de Poisson par la méthode d'inversion)

On souhaite simuler une loi de Poisson de paramètre $\lambda > 0$ par la méthode d'inversion.

1. On tire au hasard un nombre $t \in [0,1]$. Dans quel intervalle I_k le nombre t doit se trouver pour qu'on retourne k ?
2. On se donne le code suivant :

```
function k = poisson(lambda)
t = rand() ;
k = 0 ;
u = exp(-lambda) ;
s = u ;
while s<t
   k = k+1 ;
   u = u*(lambda/k) ;
   s = s+u ;
end
endfunction
```

Expliquer le fonctionnement de la fonction `poisson`.

3. Écrire une fonction `Poisson(N,lambda)` renvoyant un vecteur contenant N réalisations indépendantes de la loi $\mathscr{P}(\lambda)$.

4. À l'aide d'un diagramme en bâtons, juger de la qualité de cette simulation.

Exercice 10 (★★ - Simulation de la loi géométrique par méthode d'inversion)

1. Écrire une fonction `geom3` qui prend en paramètre un nombre $p \in]0,1[$, puis à l'aide de la méthode d'inversion, simule une loi géométrique de paramètre p.

2. Écrire une fonction `Geom3` renvoyant un échantillon de taille N de la loi $\mathscr{G}(p)$.

3. À l'aide d'un diagramme en bâtons, comparer les simulations de la loi $\mathscr{G}(0.2)$ à l'aide des fonctions `Geom1`, `Geom2` et `Geom3`.

---

4 Exercices

Exercice 11 (★★)

À un guichet, des clients peuvent venir expédier ou retirer un colis. Au cours d'une journée, le nombre de clients N qui s'y présentent suit la loi de Poisson de paramètre λ (où $\lambda \in \mathbb{R}_+^*$). Chaque client a une probabilité p (où $p \in]0,1[$) de venir pour expédier un colis et $1-p$ pour en retirer un.
On note C le nombre de colis expédiés dans la journée.

a) Pour tout $k \in \mathbb{N}$, déterminer l'espérance de C conditionnellement à l'événement $[N = k]$.

b) En déduire l'espérance de C.

c) Que fait le programme suivant ? Comparer l'espérance empirique obtenue avec l'espérance théorique.

```
p = 0.3 ;
lambda = 2.5 ;
n = 10000 ;
N = grand(1,n,"poi",lambda) ;
C = [ ] ;
for i = 1:n
   C = [C, grand(1,1,"bin",N(i),p)] ;
end

```

```
Cbar = tabul(C) ;
bar(Cbar(:,1),Cbar(:,2)/n) ;

disp(mean(C))
```

Exercice 12 (★★★ - Un problème de supermarché : la loi de Borel)

On considère une caisse de supermarché fonctionnant sur le principe suivant : chaque client passe une minute à la caisse, et pendant qu'un client est servi, le nombre de clients arrivant à la caisse est une variable aléatoire suivant une loi de Poisson $\mathscr{P}(\mu)$.

1. Écrire une fonction `caisse` de paramètre μ estimant le nombre de minutes nécessaires entre l'arrivée du premier client jusqu'à ce que la caisse soit vide.

   Essayer cette fonction à l'aide de plusieurs valeurs de μ.

   Remarque. Si un programme semble ne pas s'arrêter, on peut utiliser la combinaison de touches ***`Ctrl + C`*** *suivie de* **`abort`**.

2. Constater que pour $\mu < 1$, la moyenne est de l'ordre de $\dfrac{1}{1-\mu}$.

   Expliquer ce résultat en considérant le nombre moyen de clients se présentant à la caisse pendant que le premier est servi, puis le nombre moyen de clients arrivant à la caisse pendant que ces "clients de première génération" sont servis, etc...

3. La probabilité que ce nombre de clients soit égal à n est égale à $\dfrac{e^{-\mu n}(\mu n)^{n-1}}{n!}$. Vérifier ceci graphiquement.
