Introduction à la simulation en probabilité

Philippe Briand

Année universitaire 2015–2016

1. Mise en route – Principe de simulation.

Connectez-vous puis lancer le programme `Scilab`.

La fonction `rand` génère des réels distribuées suivant la loi uniforme sur $[0,1]$: `rand(M,N)` fournit une matrice de taille $M \times N$ de réels distribués suivant la loi uniforme sur $[0,1]$.

Tapez `N=1000; u=rand(1,N);` pour obtenir un vecteur ligne de longueur `N`. La commande `mean(u)` donne la moyenne arithmétique du vecteur `u`. Le résultat doit être voisin de 1/2 la moyenne de la loi uniforme sur $[0,1]$. Expérimentez plusieurs valeurs de `N`.

La commande `N=1000; u=rand(1,N);` fournit un vecteur (ligne) de longueur N dont les composantes sont des réels distribués suivant la loi uniforme sur $[0,1]$. Les composantes du vecteur `u` s'interprètent comme une réalisation particulière – c'est à dire pour un $\omega \in \Omega$ donné – de N v.a. U_1, ..., U_N indépendantes et identiquement distribuées suivant la loi uniforme sur $[0,1]$. En clair, on considère que

$$\texttt{u=(u\_1,...,u\_N)} = (U_1(\omega), \ldots, U_N(\omega))\,.$$

La commande `mean(u)` renvoie la moyenne arithmétique du vecteur `u` c'est à dire

$$\texttt{mean(u) = (u\_1+...+u\_N)/N} = \frac{U_1(\omega) + \ldots + U_N(\omega)}{N} \;;$$

la loi forte des grands nombres garantit que, si N est grand, cette dernière quantité est voisine de $\mathbb{E}[U]$ où U suit la loi uniforme sur $[0,1]$.

Rappelons que si U suit la loi uniforme sur $[0,1]$ et $t \in [0,1]$, $\mathbb{P}(\{U \leq t\}) = t$. Nous allons le vérifier. Pour cela, on approche la probabilité $\mathbb{P}(\{U \leq t\})$ par la fréquence empirique de l'événement $\{U \leq t\}$:

$$\mathbb{P}(\{U \leq t\}) \simeq \frac{\text{Nombre de fois où } U_i(\omega) \leq t}{\text{Nombre total d'expériences}}.$$

Tapez `N=1000; u=rand(1,N);` pour obtenir un vecteur de longueur N dont les composantes sont censées être des réels distribués suivant la loi uniforme sur $[0,1]$. Pour compter le nombre de composantes de ce vecteur qui sont inférieures à t, on peut utiliser la commande `l=(u<=t)` qui renvoie un vecteur booléen formé de `T` (true) et de `F` (false) suivant que la condition `u<=t` est vérifiée ou pas. Il reste à faire la somme des composantes de l, `sum(l)` pour obtenir le nombre de fois où `u_i<=t`. En résumé, la ligne de commande

```
N=10000; t=0.3; u=rand(1,N); l=(u<=t); sum(l)/N
```

devrait fournir un résultat proche de t. Faites varier N et t. On peut convertir le vecteur booléen `(u<=t)` en un vecteur contenant des 0 et des 1 au moyen de la commande `bool2s(u<=t)` ; on peut alors utiliser la commande `mean` :

```
N=10000; t=0.3; u=rand(1,N); l=bool2s(u<=t); mean(l)
```

2. Simulation de variables aléatoires.

Pour simuler des variables aléatoires distribuées suivant une loi quelconque, l'idée est de se ramener à la loi uniforme sur $[0, 1]$.

2.1. Variable de Bernoulli.

Une variable aléatoire de Bernoulli de paramètre $p \in]0, 1[$ est une variable aléatoire X prenant deux valeurs 1 et 0 respectivement avec probabilité p et $1-p$. Vérifiez que si U suit la loi uniforme sur $[0, 1]$, $X = \mathbf{1}_{U \leq p}$ suit la loi $\mathcal{B}(p)$.

Utilisez cette observation pour générer un vecteur ligne de longueur N distribué suivant la loi $\mathcal{B}(p)$:

```
N=10000; p=0.3; u=rand(1,N); l=bool2s(u<=p);
```

Vérifiez que la probabilité $\mathbb{P}(\{X = 1\})$ est correctement estimée : `sum(l==1)/N`.

2.2. Génération d'une binomiale.

Si $X_1, \ldots, X_n$ sont n variables aléatoires indépendantes et identiquement distribuées suivant la loi de Bernoulli $\mathcal{B}(p)$, la variable aléatoire

$$Y = X_1 + X_2 + \ldots + X_n$$

suit la loi binomiale $\mathcal{B}(n, p)$.

1. Pour générer un vecteur de longueur `N` de binomiales $\mathcal{B}(n, p)$, on commencera par générer un tableau $n \times N$ contenant des variables de Bernoulli X_i avec $\mathbb{P}(\{X_i = 1\}) = p$. Il suffit de faire alors la somme des lignes de ce tableau.

   ```
   u=rand(n,N); l=(u<=p); y=sum(l,1);
   ```

   Pour comprendre le rôle du deuxième argument de la fonction `sum`, tapez

   ```
   A=[1 2 3 ; 4 5 6]; disp(A); L=sum(A,1); disp(L); C=sum(A,2); disp(C);
   ```

   À l'aide de l'éditeur de `Scilab`, créer un répertoire `MATH532` puis à l'intérieur de ce répertoire un fichier `loisusuelles.sci` contenant

   ```
   function y=bin(n,p,N)
       u=rand(n,N); l=(u<=p); y=sum(l,1);
   endfunction
   ```

   Vous pourrez désormais vous servir de la fonction `bin`.

2. Pour estimer empiriquement la probabilité $\mathbb{P}(\{Y = k\})$, il suffit d'utiliser la commande `sum(y==k)/N`, `N` étant la longueur de `y`. La valeur théorique de $\mathbb{P}(\{Y = k\})$ est

$$\mathbb{P}(\{Y = k\}) = \binom{n}{k} p^k (1-p)^{n-k}.$$

On calculera les coefficients binomiaux à l'aide de la fonction Γ, $\Gamma(n+1) = n!$ pour $n \in \mathbf{N}$:

$$\binom{n}{k} = \frac{\Gamma(n+1)}{\Gamma(k+1)\Gamma(n-k+1)},$$

la fonction Γ s'appelle `gamma` dans `Scilab`. On prendra $n = 7$ et $p = 0.3$. L'objectif est de comparer, pour $0 \leq k \leq 10$, la valeur estimée (`pe`) et la valeur théorique (`pt`) de $\mathbb{P}(\{Y = k\})$. Créez un fichier `testbin.sce` contenant les instructions suivantes

```
exec('loisusuelles.sci');
n=7; p=0.3; N=100;
y=bin(n,p,N);
pe=zeros(n+1,1); pt=zeros(n+1,1);
for k=0:n
    pe(k+1)=sum(y==k)/N;
    pt(k+1)=gamma(n+1)/gamma(k+1)/gamma(n-k+1);
    pt(k+1)=pt(k+1)*p^k*(1-p)^(n-k);
end
disp([pe pt]);
```

puis exécutez-le. Pour exécuter ces instructions, placer vous dans `Scilab` dans le répertoire `MATH532` puis tapez simplement `exec('testbin.sce')` ou alors cliquez sur l'icone correspondant de l'éditeur. Les résultats sont-ils bons ? Prenez `N=1000`, puis `N=10000`, puis `N=100000`. Qu'observez-vous ? Retenez la structure des boucles `for` :

```
for i=1:n
    instructions
end
```

2.3. Lois amnésiques.

La densité de la variable exponentielle de paramètre $c > 0$ est

$$p(x) = ce^{-cx}1_{x\geq 0}.$$

On a vu en TD que lorsque U suit la loi uniforme sur $[0, 1]$, la variable aléatoire

$$X = \frac{-\ln(U)}{c}$$

suit la loi exponentielle de paramètre c. Refaites le calcul si besoin.

1. Faire une fonction `x=expo(c,N)` qui génère une suite de variables aléatoires exponentielles de longueur N. Ajoutez cette fonction au fichier `loisusuelles.sci`.
2. Créer un fichier `testexpo.sce` permettant de comparer, pour la loi exponentielle $\mathcal{E}(c)$, la valeur théorique de $\mathbb{P}(\{X > 1\})$ avec une estimation de cette probabilité. Inspirez-vous de `testbin.sce` :

```
exec('loisusuelles.sci');
c=2; N=1000;
x=expo(c,N);
pt=exp(-c);
pe=sum(x>1)/N;
disp([pe pt]);
```

Exécutez les instructions en tapant dans la console `exec('testexpo.sce')` ou en cliquant dans la barre d'outils de l'éditeur. Testez plusieurs valeurs de c et de N.

3. La loi exponentielle est une loi sans mémoire au sens où, si X est une v.a. de loi $\mathcal{E}(c)$, la probabilité conditionnelle

$$\mathbb{P}(\{X > t+h\} \mid \{X > t\}) = \frac{\mathbb{P}(\{X > t+h\})}{\mathbb{P}(\{X > t\})}$$

est égale à $\mathbb{P}(\{X > h\})$; la probabilité conditionnelle dépend seulement de h, pas de t. En effet, un calcul immédiat donne, dans le cas de la loi exponentielle,

$$\mathbb{P}(\{X > t+h\} \mid \{X > t\}) = \frac{\mathbb{P}(\{X > t+h\})}{\mathbb{P}(\{X > t\})} = \frac{e^{-c(t+h)}}{e^{-ct}} = e^{-ch} = \mathbb{P}(\{X > h\}).$$

À l'intérieur du répertoire `MATH532`, créer un fichier `amexpo.sce` Ce fichier doit afficher, pour la loi exponentielle $\mathcal{E}(c)$, la valeur théorique de la probabilité $\mathbb{P}(\{X > h\})$, une estimation de cette probabilité ainsi qu'une estimation de la probabilité conditionnelle $\mathbb{P}(\{X > t+h\} \mid \{X > t\}) = \mathbb{P}(\{X > t+h\})/\mathbb{P}(\{X > t\})$. On pourra commencer par `N=1000`, `c=0.5`, `t=2` et `h=1`. Faites grandir `t`. Qu'observez-vous ? Augmenter à présent la valeur de `N`.

4. La fonction `histplot` permet de tracer des histogrammes normalisés de sorte que l'intégrale sous la courbe soit égale à 1. Si `u=(u_1,...,u_N)` est un vecteur, `histplot(k,u)` répartit les points de `u` en `k` intervalles réguliers et trace `k` bâtons dont la hauteur est proportionnelle au nombre de points de `u` contenus dans chacun des intervalles.

Essayez : `N=1000; u=rand(1,N); histplot(25,u)` Faites grandir `N` ! Que constatez-vous ?

Créer un fichier `densiteexpo.sce` qui pour `N` réalisations de la loi exponentielle de paramètre c trace l'histogramme associé. Tracer également la densité de la loi exponentielle. On pourra utiliser la séquence d'instructions, si `x=expo(c,N)`,

```
m=max(x); t=0:m/100:m; d=c*exp(-c*t); plot(t,d);
```

Est-ce cohérent ? Augmentez `N` et le nombre de bâtons `k`. Qu'observez-vous ?

5. Soient U et V deux variables aléatoires indépendantes et identiquement distribuées suivant la loi uniforme sur $[0,1]$. Utilisez la méthode de l'histogramme pour déterminer la densité de $S = U + V$.

6. On a vu également que, si X suit la loi $\mathcal{E}(c)$, $Y = 1 + [X]$ suit une loi géométrique de paramètre $p = 1 - e^{-c}$:

$$\forall k \in \mathbf{N}^*, \qquad \mathbb{P}(\{Y = k\}) = q_k = p(1-p)^{k-1}$$

Faire une fonction `y=geom(p,N)` qui génère une suite de variables aléatoires géométriques de paramètre p de longueur `N`. Cette fonction appellera la fonction `expo` et sera placée dans le fichier `loisusuelles.sci`.

7. Simuler une telle suite et pour chaque valeur de k, estimez la probabilité $\mathbb{P}(\{Y = k\})$. On se contentera de $1 \leq k \leq 10$, et l'on prendra p aux alentours de 0,2. On appelle `pe(k)` cette estimée. Placez ces instructions dans le fichier `geolog`. Pensez aux instructions de `testbin.m` ! Tracer le graphe $k \longmapsto \ln($`pe(k)`$)$ en tapant `plot(1:10,log(pe));` Commencez par `N=100` puis augmentez `N`. Qu'observez-vous ? Est-ce normal ?

3. Méthode de Monte-Carlo.

L'objectif est de calculer $\int_0^1 f(x)\,dx$ pour f fonction de $[0,1]$ dans $\mathbf{R}$. Le principe de la méthode de Monte-Carlo repose sur la loi forte des grands nombres. En effet, si $(U_i)_{i\geq 1}$ est une suite de v.a. i.i.d. suivant la loi uniforme sur $[0,1]$, lorsque $N \to +\infty$,

$$\frac{1}{N}\sum_{i=1}^{N} f(U_i(\omega)) \longrightarrow \mathbb{E}[f(U_1)] = \int_0^1 f(x)\,dx,$$

la convergence ayant lieu pour presque tout $\omega \in \Omega$ dès que $\int_0^1 |f(x)|\,dx < +\infty$.

Revenons à `Scilab`. La commande `N=1000; u=rand(1,N);` fournit un vecteur (ligne) de longueur N dont les composantes sont des réels distribués suivant la loi uniforme sur $[0,1]$. Les composantes du vecteur `u` s'interprètent comme une réalisation particulière – c'est à dire pour un $\omega \in \Omega$ donné – de N v.a. $U_1, \ldots, U_N$ indépendantes et identiquement distribuées suivant la loi uniforme sur $[0,1]$. En clair, on considère que

$$\texttt{u=(u\_1,...,u\_N)} = (U_1(\omega), \ldots, U_N(\omega)).$$

C'est parti. Tapez `N=1000; u=rand(1,N);` puis `f=u.*u;` : cette dernière commande élève tous les termes du vecteur `u` au carré. Finalement, la commande `mean(f)` retourne la moyenne arithmétique du vecteur `f`. Cette valeur est-elle proche de 1/3 ? C'est normal puisque $\int_0^1 x^2\,dx = 1/3$. Essayez d'autres valeurs de `N`. Qu'observez-vous quand `N` grandit ?

Essayez à présent cette procédure avec la fonction $x \longmapsto \sin(\pi x)$. Pour cela, il suffit d'utiliser la commande `f=sin(%pi*u); mean(f)`. Faites varier `N`.

4. Le premier 6.

Dans cet exercice, on essaie de trouver le temps moyen nécessaire à l'obtention d'un 6 lorsqu'on joue au dé. Mathématiquement, on cherche l'espérance du temps T où

$$T = \inf\{n \geq 1 : X_n = 6\},$$

avec $(X_n)_{n\geq 1}$ i.i.d. suivant la loi uniforme sur $\{1, \ldots, 6\}$.

On peut facilement simuler une v.a. X de loi uniforme sur $\{1, \ldots, 6\}$ à partir d'une v.a. U uniforme sur $[0,1]$ via la formule $X = 1 + [6U]$.

En `Scilab`, la fonction partie entière s'appelle `floor` : les instructions

```
n=1000; u=rand(1,n); x=1+floor(6*u);
```

fournissent une vecteur `x` de longueur `n` de loi uniforme sur $\{1, \ldots, n\}$. Pour vous en convaincre, tapez la suite d'instructions

```
n=1000; u=rand(1,n); x=1+floor(6*u); mean(x)
```

en vous rappelant que la valeur moyenne d'un dé vaut 3,5. Si vous tapez les commandes

```
n=1000; u=rand(1,n); x=1+floor(6*u); sum(x==6)/n
```

qu'obtenez-vous ? Pourquoi ?

Nous aurons besoin dans cet exercice des boucles `for` et `while` dont la structure est la suivante :

```
for i=1:mc                          while condition
    instructions                        instructions
end                                 end
```

Ainsi pour obtenir une réalisation de T, on peut écrire dans le fichier `premiersix.sce` :

```
x=0; T=0;
while x<6
    u=rand(1,1); x=1+floor(6*u);
    T=T+1;
end
disp(T);
```

Essayez plusieurs fois en tapant `exec('premiersix.sce')`.

Nous allons à présent estimer l'espérance de T en calculant la moyenne arithmétique de `mc` réalisations indépendantes de T. Modifiez `premiersix.sce`

```
mc=10000; mT=0;
for i=1:mc
    x=0; T=0;
    while x<6
        u=rand(1,1);
        x=floor(6*u)+1;
        T=T+1;
    end
    // disp(T);
    mT=mT+T;
end
mT=mT/mc; disp('Temps moyen du premier six : '+string(mT));
```

Modifiez ce programme pour qu'il détermine également la valeur moyenne de la somme des dés jusqu'à l'obtention du premier 6 ainsi que $\mathbb{P}(T > \mathbb{E}[T] + r)$ où r est un entier de votre choix ; normalement, vous avez constaté que $\mathbb{E}[T] = 6$.

5. Ruine du joueur.

James Bond arrive dans un Casino (Royale) avec a euros en poche. Il joue à la roulette en pariant à chaque fois un euro sur « pair ». Sa fortune S évolue au cours du temps de la manière suivante :

$$S_0 = a, \qquad S_n = a + \sum_{k=1}^{n} X_k, \quad n \geq 1,$$

où les $(X_n)_{n\geq 1}$ sont i.i.d. suivant la loi $\mathbb{P}(X_1 = 1) = 1 - \mathbb{P}(X_1 = -1) = p \in]0, 1[$. Le jeu lui est favorable si $p > 1/2$, défavorable si $p < 1/2$; si $p = 1/2$ le jeu est équitable.

Il décide de s'arrêter de jouer dès qu'il aura gagné b euros à moins bien sûr qu'il ne soit ruiné avant. On s'intéresse donc à la quantité suivante :

$$T = \inf\{n \geq 1 : S_n = 0 \text{ ou } S_n = a + b\}.$$

Écrire dans le fichier `bond.sce` un programme permettant d'estimer les quantités : $\mathbb{P}(S_T = 0)$ (James Bond est ruiné), $\mathbb{P}(S_T = a+b)$ (James Bond gagne la partie) ainsi que la durée moyenne de la partie à savoir $\mathbb{E}[T]$.

La théorie donne les valeurs suivantes : pour $p = 1/2$,

$$\mathbb{P}(S_T = 0) = \frac{b}{a+b}, \quad \mathbb{P}(S_T = a+b) = \frac{a}{a+b}, \quad \mathbb{E}[T] = ab,$$

pour $p \neq 1/2$, notant $r = (1-p)/p$,

$$\mathbb{P}(S_T = 0) = \frac{r^{a+b} - r^a}{r^{a+b} - 1}, \quad \mathbb{P}(S_T = a+b) = \frac{r^a - 1}{r^{a+b} - 1}, \quad \mathbb{E}[T] = \frac{1}{2p-1}\left(\frac{(a+b)(1-r^a)}{1-r^{a+b}} - a\right).$$

Retrouvez-vous ces valeurs ?

À la roulette, il y a 37 numéros : le 0 est vert, 18 pairs rouges et 18 impairs noirs. J. Bond arrive avec 100 euros et veut gagner 10 euros. Quelle est la probabilité qu'il y parvienne ? Combien de fois va-t-il devoir miser ?

6. Illustration du Théorème Central Limite.

Rappelons que, lorsque $(X_i)_{i\geq 1}$ est une suite de v.a. i.i.d. avec $\mu = \mathbb{E}[X_1]$ et $\sigma^2 = \mathbb{V}(X_1)$,

$$Y_n = \frac{\sqrt{n}}{\sqrt{\sigma^2}}\left(\frac{1}{n}\sum_{i=1}^{n} X_i - \mu\right) = \frac{\sum_{i=1}^{n} X_i - n\mu}{\sqrt{n\sigma^2}}$$

converge en loi vers G de loi normale centrée réduite c'est à dire de densité $x \longmapsto e^{-x^2/2}/\sqrt{2\pi}$.

Utilisez la méthode des histogrammes, pour illustrer cette convergence. Pour n grand, on pourra considérer m réalisations de Y_n puis tracer l'histogramme des Y_n et la densité de G. Faites le test pour X_1 suivant la loi de Bernoulli $\mathcal{B}(p)$, puis pour la loi uniforme sur $[-2, +2]$.

7. Promenade sur le tore.

On considère l'ensemble $I = \{0, 1, \ldots, 9\}$ en convenant que le successeur de 9 est 0. Un mobile partant de 0 se déplace sur cet ensemble avec la règle suivante : à chaque instant, il reste sur place avec probabilité $0 < p < 1$ et il avance d'une place avec probabilité $1-p$. On note X_n sa position à l'instant n.

1. Écrire la fonction Scilab `y = function suivant(x,p)` donnant la position aléatoire suivante de x.

2. Tracer une trajectoire issue de 0 aux instants 1, 2, ..., 100.

3. On note T le temps de retour au point 0 :

$$T = \inf\{n \geq 1 : X_n = 0\}.$$

   Écrire la fonction Scilab `function T=Retour(p)` simulant T. Estimer la moyenne et la variance de T. Ces quantités dépendent-elles de p ?

4. Simuler la position du mobile à l'instant n et tracer l'histogramme pour 1000 tirages. Prenez $n = 20$, $n = 100$, $n = 250$. Qu'en déduisez-vous ?

8. Le collectionneur.

Chaque paquet de céréales de marque X, contient en cadeau (empoisonné) un autocollant. Il y a m autocollants différents. Le problème est de savoir combien de paquets de céréales sont nécessaires en moyenne pour obtenir tous les autocollants.

Créer un fichier `collection.sce` qui permet d'obtenir le nombre `N` de paquets nécessaires à l'obtention de la collection complète. On supposera que les autocollants se trouvant dans les paquets de céréales sont pris au hasard parmi les m autocollants possibles.

Modifiez votre programme afin qu'il fournisse une approximation du nombre moyen de paquets de céréales nécessaire. Comparez votre résultat à la valeur théorique :

$$\mathbb{E}[N] = 1 + m \sum_{k=1}^{m-1} \frac{1}{k}.$$

On prendra `m=50` pour les applications numériques.

On suppose à présent qu'un enfant collectionne les images des joueurs des équipes françaises de ligue 1. Chaque paquet contient `nv=4` images, les « doublons » étant bien évidemment permis. Sachant que le prix d'un paquet de `nv=4` vignettes est de 0,20 euros, combien faut-il dépenser en moyenne pour remplir l'album qui comporte 400 images ?

9. Polynômes de Berstein.

Soient $(U_i)_{i\geq 1}$ une suite de variables aléatoires i.i.d. de loi uniforme sur $[0,1]$ et $f : [0,1] \longrightarrow \mathbf{R}$ une fonction continue. On note $B_n(f)$ le polynôme de Bernstein

$$B_n(f)(x) = \mathbb{E}\left[f\left(M_n(x)\right)\right], \qquad M_n(x) = \frac{1}{n}\sum_{k=1}^{n} \mathbf{1}_{U_i \leq x},$$

et on rappelle que $B_n(f)$ converge vers f uniformément sur $[0,1]$.

1. En choisissant la fonction $f(x) = \sqrt{x}$, illustrer cette convergence. On pourra tracer sur le même graphique f ainsi que $B_n(f)$ pour différentes valeurs de n.

2. Pour $f(x) = \sqrt{x}$ puis $f(x) = |x|$ tracer le graphe $n\,|f - B_n(f)|$. Quelle conclusion en tirez-vous ?