Université de Bordeaux **Mathématiques**
Préparation à l'agrégation **Année 2018/2019**

TP2 : Simulation de Variables Aléatoires

En pratique, de nombreuses lois peuvent être simulées avec des fonctions déjà implémentées dans Scilab. Néanmoins, vous devez savoir simuler les lois classiques en repartant à chaque fois de la loi uniforme. Ainsi, dans les exercices suivants, la seule fonction de simulation Scilab que vous utiliserez est `rand`. Celle-ci fait appel à un générateur de nombres pseudo-aléatoires ; elle permet de générer des nombres dont la distribution est proche d'une loi uniforme. De plus, plusieurs appels à la fonction `rand` (ou alors un appel à `rand(m,n)`) génèrent des nombres aléatoires qu'on peut considérer indépendants.

Exercice 1. Simulation de lois discrètes

1. Écrire une fonction `bern(p)` qui simule un nombre aléatoire de loi de Bernoulli $\mathcal{B}(p)$.

2. Modifier la fonction précédente en `bern(m,n,p)` de façon à générer une matrice $m \times n$ de nombres aléatoires indépendants suivant la loi $\mathcal{B}(p)$.

3. Écrire une fonction `binom(n,p)` qui simule un nombre aléatoire de loi binomiale $\mathcal{B}(n, p)$.

4. Écrire une fonction `geom(p)` qui simule un nombre aléatoire de loi géométrique $\mathcal{G}(p)$.

5. Écrire une fonction `unifdiscrete(n)` qui simule un nombre aléatoire de loi uniforme sur $\{1, 2, \dots, n\}$.

6. Simuler la loi p sur $\{-2, 3, 8\}$ donnée par $p(-2) = \frac{2}{3}$, $p(3) = \frac{1}{6}$ et $p(8) = \frac{1}{6}$.

7. Écrire une fonction `discrete(p)` qui simule un nombre aléatoire dont la loi discrète sur $\{1, \dots, n\}$ est donnée par les éléments `p(1), ... , p(n)` du paramètre `p`.

Exercice 2. Méthode d'inversion

Soit X une v.a.r. de fonction de répartition F. On définit la fonction quantile (ou "inverse généralisée" de F) par

$$F^{-}(u) = \inf\{\, x \in \mathbb{R} \mid F(x) \geqslant u \,\} \quad (u \in]0,1[) .$$

1. Montrer que F^{-} est bien définie, et que l'inf est en fait un min. Autrement dit,

$$\forall x \in \mathbb{R}, \forall u \in]0,1[, \quad u \leqslant F(x) \iff F^{-}(u) \leqslant x .$$

2. En déduire que si U est une loi uniforme sur $[0, 1]$, alors $F^{-}(U)$ admet F pour fonction de répartition.

3. Que vaut F^{-} si F réalise une bijection d'un intervalle ouvert I sur $]0, 1[$?

4. Écrire une fonction `randexp(m,n,lambda)` qui simule $m \times n$ nombres aléatoires de loi $\mathcal{E}(\lambda)$.

5. (Bonus) Écrire une fonction `cauchy` qui simule un nombre aléatoire de loi de Cauchy $\dfrac{dx}{\pi(1 + x^2)}$.

Exercice 3. Affichage de lois et lois empiriques

1. Afficher la densité gaussienne $x \mapsto \dfrac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\dfrac{-(x-m)^2}{2\sigma^2}\right)$.

2. Calculer et afficher la fonction de répartition de la loi $\mathcal{E}(\lambda)$.

3. À un n-échantillon $(x_1, \dots, x_n)$ de loi μ on associe la loi empirique et la fonction de répartition empirique

$$\widehat{\mu}_n = \frac{1}{n} \sum_{i=1}^{n} \delta_{x_i} \quad , \quad \widehat{F}_n(x) = \frac{1}{n} \sum_{i=1}^{n} \mathbf{1}_{x_i \leqslant x} .$$

a. En utilisant les fonctions `gsort` et `plot2d2`, écrire une fonction `fdremp(x)` qui trace la fonction de répartition empirique associé au n-échantillon stocké dans le vecteur `x`.

b. Vérifier ainsi la fonction `randexp` que vous avez écrite dans l'Exercice 2 : tirer un n-échantillon avec `randexp` et vérifier graphiquement que la fonction de répartition empirique $\widehat{F_n}$ est proche de la fonction de répartition attendue F. (Bonus : calculer l'erreur $\|F - \widehat{F_n}\|_\infty$ en norme uniforme.)

4. Une autre manière de visualiser la répartition du n-échantillon (x_i) est d'afficher un histogramme. Ceci est particulièrement utile lorsque la loi μ est absolument continue, auquel cas pour n grand, l'histogramme est comparable à la densité sous-jacente. Pour visualiser l'histogramme d'un n-échantillon `x`, on utilisera

```
histplot(nclasses,x)
```

Par défaut, `histplot` normalise l'histogramme de façon à ce que l'aire sous l'histogramme soit égale à 1. De plus, à nombre de classes donné, `histplot` choisit automatiquement les classes utilisées. Le nombre de classes peut être fixée à la main. Un choix standard est `nclasses` $= \lceil 4n^{1/4} \rceil$.

$\longrightarrow$ Vérifier de nouveau votre fonction `randexp` en superposant histogramme et densité théorique.

5. Pour visualiser des lois discrètes sur un ensemble fini de nombres, on utilise un diagramme en bâtons affiché avec `plot2d3`. Visualiser ainsi la loi binomiale $\mathcal{B}(n, p)$ pour différentes valeurs de n et p (on pourra utiliser la fonction `factorial`).

6. Si $(x_1, \ldots, x_n)$ est un n-échantillon d'une loi discrète à valeurs dans $\{a_1, \ldots, a_p\}$ avec $a_1 < \ldots < a_p$, on peut calculer sa loi empirique avec la commande

```
dhistx = histc(at,x,normalization=%t)
```

où `at(1)` contient $a_1 - 1$ et `at(2:$)` contient les valeurs $a_1, \ldots, a_p$. Le paramètre de normalisation fait en sorte que la somme des valeurs de `dhistx` soit égale à 1.

$\longrightarrow$ Vérifier ainsi la fonction `binom(n,p)` que vous avez écrite dans l'Exercice 1.

NB : Si vous préférez les diagrammes en barres, vous pourrez tester chez vous la fonction `bar`.

Exercice 4. Méthode de Box-Muller pour simuler $\mathcal{N}(0, 1)$

1. Soit S et Θ deux v.a. indépendantes de lois respectives $\mathcal{E}(\frac{1}{2})$ et $\mathcal{U}([0, 2\pi])$.

Montrer que $X = \sqrt{S}\cos(\Theta)$ et $Y = \sqrt{S}\sin(\Theta)$ sont indépendantes de loi $\mathcal{N}(0, 1)$.

2. En déduire une méthode de simulation de la loi $\mathcal{N}(0, 1)$ et l'implémenter.

3. Vérifier votre fonction en superposant histogramme et densité théorique.

Exercice 5. Méthode de rejet

Soient $B \subset A$ deux boréliens de $\mathbb{R}^2$ tels que $0 < \lambda(B) \leqslant \lambda(A) < \infty$.

Soit (X_n) une suite i.i.d. de loi $\mathcal{U}(A)$.

1. Montrer que $T = \min\{\, n \in \mathbb{N}^* \mid X_n \in B \,\}$ suit la loi $\mathcal{G}(p)$ où $p = \frac{\lambda(B)}{\lambda(A)}$.

2. Montrer que X_T suit la loi uniforme sur B, et est indépendante de T.

3. Construire un algorithme de simulation de la loi uniforme sur l'intérieur de l'ellipse d'équation $\frac{x^2}{4} + y^2 = 1$.

4. Implémenter cet algorithme, et tirer 1000 échantillons (x_i, y_i) de cette loi et les représenter avec

```
plot2d(x,y,-1)
```

puis superposer l'ellipse avec les commandes

```
theta = linspace(-%pi,%pi,1e3);
plot2d(2*cos(theta),sin(theta),2)
```

5. À B fixé, comment choisir A ? Commenter en terme de temps de calcul.

La bonne nouvelle de fin de TP : Bien entendu, il existe déjà dans Scilab des fonctions permettant de simuler la plupart des lois usuelles (`grand`) et aussi de calculer les fonctions de répartition et fonctions quantiles (par exemple `cdfnor` pour la loi normale, dont l'usage et la documentation requièrent une certaine attention). On trouvera aussi des compléments dans la boîte à outils *Stixbox* (par exemple les fonctions `distfun`$\star$ pour “distribution functions”).