Université Paris Descartes
UFR de Mathématiques et Informatique

Introduction aux probabilités - Licence MIA 2e année - parcours Informatique
Contrôle continu du 16/12/2009 - Durée : 1 heure

Les documents ne sont pas autorisés. **Tous les codes demandés devront être écrits très soigneusement afin que chaque caractère soit lisible.**

Exercice 1

1. Ecrire la commande R permettant de tirer au sort 10 valeurs selon la loi binomiale de paramètres $n = 100$ et $p = 0.02$.
2. Dire pourquoi on peut approcher cette loi binomiale par une loi de Poisson de paramètre λ dont on donnera la valeur.
3. Ecrire la commande R permettant de tirer au sort 10 valeurs selon la loi de Poisson de paramètre λ.

Exercice 2 Une marche aléatoire consiste en l'expérience probabiliste suivante : on se place à la position origine d'un axe, puis à chaque étape on fait un pas de longueur 1 vers la droite ou la gauche, avec probabilité 1/2 pour chaque direction. On effectue n étapes au total. On note X_i la position du point à l'étape i (avec $X_0 = 0$).

1. Ecrire un programme R, en utilisant une boucle `for`, qui simule la marche aléatoire et construit un vecteur x de taille $n+1$, où $n = 10000$, correspondant aux positions successives obtenues.
2. Pour éviter d'utiliser des boucles, on propose de remplacer le programme précédent par celui-ci :

```
n = 10000
u = sample(c(-1,1),n,rep=TRUE)
x = cumsum(u)
x = c(0,x)
```

   Expliquer pourquoi ce programme permet aussi de simuler la marche aléatoire. (remarque : la fonction `cumsum` est décrite à la fin du sujet.)

Exercice 3 Considérons l'intégrale

$$I_1 = \int_0^1 \frac{1}{1+x^2} dx.$$

On peut calculer numériquement cette intégrale en effectuant les opérations suivantes :

– On simule un grand nombre n de variables X_i indépendantes et de loi uniforme sur $[0, 1]$.
– On calcule la moyenne empirique des variables $Y_i = \frac{1}{1+X_i^2}$:

$$\frac{1}{n} \sum_{i=1}^{n} \frac{1}{1+X_i^2}.$$

Cette méthode s'appelle méthode de Monte-Carlo.

1. Ecrire une fonction `MonteCarlo`, en langage R, prenant comme argument le nombre n et renvoyant la valeur approchée de I_1 obtenue avec la méthode de Monte-Carlo.
2. Modifier la fonction pour qu'elle renvoie une valeur approchée du nombre π.

3. Plus généralement, supposons que l'on cherche à calculer une intégrale du type
$$I = \int_{-\infty}^{+\infty} g(x)f(x)dx,$$
où f est une fonction de densité et g une fonction quelconque. La méthode de Monte-Carlo consiste alors à simuler n variables X_i indépendantes suivant la loi de densité f, puis à calculer la moyenne empirique $\frac{1}{n}\sum_{i=1}^{n} g(X_i)$. En effet, d'après la loi des grands nombres, cette moyenne empirique converge vers $E(g(X))$, et on peut voir facilement que $I = E(g(X))$.
On considère à présent l'intégrale
$$I_2 = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} \cos(x) e^{-\frac{x^2}{2}} dx.$$
Quelles fonctions f et g peut-on choisir pour que I_2 corresponde à l'intégrale I ? Quelle est la loi dont la densité est f ?
4. Réécrire la fonction `MonteCarlo` pour qu'elle renvoie une valeur approchée de I_2.

Annexe : quelques rappels de R

Lois prédéfinies en R

Loi	Appelation	Arguments	Valeurs par défaut
Binomiale de paramètre $(n, q) \in \mathbb{N}^* \times [0, 1]$	`binom`	`size=n` `prob=q`	
Binomiale négative de paramètre $(n, q) \in \mathbb{N}^* \times [0, 1]$	`nbinom`	`size=n` `prob=q`	
Géométrique de paramètre $q \in [0, 1]$	`geom`	`prob=q`	
Normale (ou Gaussienne) de paramètre $(\mu, \sigma) \in \mathbb{R} \times \mathbb{R}_+$	`norm`	`mean=`μ `sd=`σ	`mean=0` `sd=1`
Poisson de paramètre $\lambda \in \mathbb{R}_+$	`pois`	`lambda=`λ	
Uniforme sur $[min, max]$	`unif`	`min=`min `max=`max	`min=0` `max=1`

Exemples d'utilisation :
`> rpois(10,2.5)` renvoie dix tirages aléatoires suivant la loi de Poisson de paramètre 2.5,
`> dpois(4,2.5)` renvoie la valeur de $P(X = 4)$ où X suit la loi de Poisson de paramètre 2.5,
`> dnorm(-1,0.5,2)` renvoie $f(-1)$ où f est la densité de la loi normale $\mathcal{N}(\mu = 0.5, \sigma = 2)$.

Autres rappels utiles

- `sample(1 :4,5,rep=TRUE)` renvoie 5 nombres tirés au sort parmi $1, 2, 3$ ou 4 de façon équiprobable. La condition `rep=TRUE` traduit le fait qu'il s'agit d'un tirage avec remise.

- La fonction `cumsum` donne la somme cumulée du vecteur `x`. Par exemple, `cumsum(c(2,2,2,3))` renvoie le vecteur `[ 2 4 6 9 ]`