Université Paris-Dauphine Année 2011-2012
Département de Mathématique

Examen NOISE, sujet B

Résoudre trois et uniquement trois exercices au choix.

Exercice 1

On considère la distribution de Gumbel de paramètres $(\mu,\, \beta) \in \mathbb{R} \times \mathbb{R}_+$, de densité

$$g(x;\mu,\beta) \,=\, \frac{1}{\beta}\, \exp\left\{-\frac{x-\mu}{\beta} - e^{-\left(\frac{x-\mu}{\beta}\right)}\right\}\,,$$

On remarque que le mode de la distribution de Gumbell est $m = \mu$. Dans la suite on considèrera $\mu = 1$ et $\beta = 2$.

1. Proposer un algorithme d'acceptation-rejet pour la simulation de $X \sim g$ reposant sur un échantillon de loi $\mathcal{N}(m, s^2)$ où m est le mode de g et s^2 est un paramètre choisi de sorte à maximiser le taux d'acceptation (on pourra le déterminer numériquement). Écrire la fonction `fAR(...)` ayant comme paramètres de sortie le vecteur des n réalisations ainsi que le taux d'acceptation.
2. Proposer une deuxième méthode de simulation reposant sur le principe d'inversion générique à partir de réalisations d'une loi $\mathcal{U}(0,1)$. Écrire la fonction `fIG(...)` ayant comme paramètres de sortie le vecteur des n réalisations.
3. Simuler deux n-échantillons `xAR` et `xIG` à l'aide des deux différentes méthodes, et, en utilisant la commande `par(mfrow=c(1,2))`, tracer les histogrammes des réalisations obtenues par `fAR()` et `fIG()` et y superposer la densité théorique f. [*Utiliser* `par(mfrow=c(1,1))` *pour revenir à la situation d'un graphique par fenêtre.*]
4. Donner le code `R` qui permet de tracer sur un même graphique les fonctions de répartition empirique et théorique.
5. Choisir l'algorithme de simulation paraissant le plus avantageux et donner une estimation Monte-Carlo de $\mathbb{E}[X]$ et de $\mathbb{E}\left[(X - \mathbb{E}[X])^2\right]$.

Exercice 2

Soit l'intégrale

$$I = \int_4^\infty \frac{1}{\sqrt{2\pi}} e^{-\frac{y^2}{2}}\, \mathrm{d}y$$

a. Montrer par l'exemple qu'une méthode de Monte-Carlo d'évaluation de I reposant sur la génération d'un n-échantillon de variables aléatoires gaussiennes $\mathcal{N}(0,1)$ ne peut pas fonctionner.

b. Proposer une méthode de Monte-Carlo d'approximation de I reposant sur un n-échantillon de variables aléatoires de loi exponentielle $\mathcal{E}(1)$ translatée de 4, soit

```
> x=rexp(n)+4
```

(La densité d'une loi exponentielle $\mathcal{E}(1)$ translatée de 4 est $g(x) = \exp\{-(x-4)\}$ sur $[4, \infty[$.) Fournir un intervalle de confiance à 95% sur I pour $n = 10^4$.

c. Utiliser l'échantillon produit dans la question b. pour construire une évaluation $\hat{\lambda}$ par Monte-Carlo de

$$\lambda = \int_4^\infty y e^{-\frac{y^2}{2}}\,\mathrm{d}y \Big/ \int_4^\infty e^{-\frac{y^2}{2}}\,\mathrm{d}y$$

d. Proposer une méthode de Monte-Carlo d'approximation de I reposant sur un n-échantillon de variables aléatoires de loi exponentielle $\mathcal{E}(1/\hat{\lambda})$ translatée de 4, soit

```
> x=rexp(n,1/lambda)+4
```

Fournir un intervalle de confiance à 95% sur I pour $n = 10^4$ et comparer à l'intervalle de confiance de la question b.

Exercice 3

Comme vu en cours, la statistique de la distance de Kolmogorov–Smirnov entre fonction de répartition empirique et fonction de répartition théorique n'a pas la même distribution si les paramètres de la fonction de répartition théorique sont estimés.

a. En générant 1000 échantillons $\mathcal{N}(0,1)$ de taille 50, représenter l'histogramme des p-values associées au test de Kolmogorov–Smirnov,

```
> ks.test(x,"pnorm")
```

et montrer que ces p-values ont une distribution uniforme $\mathcal{U}(0,1)$ par un test de Kolmogorov–Smirnov,

```
> ks.test(p,"punif")
```

b. Reprendre l'expérience en utilisant le test de normalité,

```
> ks.test(x,"pnorm",mean(x),sd(x))
```

c. Démontrer que la distance de Kolmogorov–Smirnov obtenue pour le test `ks.test(x,"pnorm",mean(x),sd(x))` est la même que celle obtenue pour le test

```
> ks.test((x-mean(x))/sd(x),"pnorm")
```

soit donc pour les données standardisées. En déduire qu'il suffit de calculer la distribution de cette distance de Kolmogorov–Smirnov sous la loi standard $\mathcal{N}(0,1)$.

d. Pour une taille d'échantillon donnée, $n = 50$, fournir un échantillon de 1000 distances de Kolmogorov–Smirnov à partir de `ks.test` pour des simulations d'échantillons $\mathcal{N}(0,1)$ de taille n et le test d'une adéquation à la loi $\mathcal{N}(\mu,\sigma^2)$ quand μ et σ sont estimés. En déduire le quantile à 95% et le moyen de tester la normalité d'un échantillon de taille $n = 50$ correctement.

Exercice 4

Soit une variable aléatoire X de densité

$$f(x) \;=\; C\,x\,e^{-4x^2/15},\;\; x \in \mathbb{R}_+\,. \tag{1}$$

1. Écrire une fonction `fAR1()` qui permette de simuler des réalisations de X par acceptation-rejet et qui retourne un n-échantillon $\{X_i\}_{i=1,n}$ et le taux d'acceptation. (On pourra montrer par exemple que $x\,\exp\{-2x^2/15\}$ est borné sur $\mathbb{R}_+$ et utiliser une loi normale appropriée et restreinte à $\mathbb{R}_+$. Ou bien utiliser une loi exponentielle.)

2. À l'aide de `fAR1()` generer $n = 10^4$ réalisations de $X \sim f$. Calculer la constante de normalisation C et donner un intervalle de confiance sur C au niveau 95%. théorique f.
3. Calculer la fonction de répartition F de X et écrire une deuxieme fonction `fAR2()` qui permette de simuler des réalisations de $X \sim F$ par inversion générique.
4. En utilisant la commande `par(mfrow=c(1,2))`, tracer les histogrammes des réalisations obtenues par `fAR1()` et `fAR2()` et en y superposant à chaque fois la densité théorique f.
5. Quelle comparaison entre `fAR1()` et `fAR2()` fait-elle sens ? Choisir l'algorithme que vous pensez être "le meilleur".
6. Utilisant cet algorithme, donner une estimation de Monte-Carlo de $\mathbb{E}[X]$, $\mathbb{E}[X^2]$ et $P(X > 2)$ et les intervalles de confiance correspondant au niveau 95%.

Exercice 5
On veut vérifier si le théorème limite

$$\sqrt{n}\left\{\hat{q}_n - F^{-1}(\alpha)\right\} \approx \mathcal{N}(0, \sigma^2) \tag{2}$$

où $\hat{q}_n$ est l'estimateur empirique du quantile $F^{-1}(\alpha)$ s'applique à un échantillon `x` donné. On prendra $\alpha = 0.11$ dans la suite de l'exercice.

a. Créer l'échantillon `x` par la commande

```
> x=c(rnorm(100,sd=10),rcauchy(50),rexp(20,5))
```

et tester la normalité de cet échantillon par `ks.test`.

b. Générer 2500 versions bootstrap $x^\star$ de l'échantillon `x` et calculer $\hat{q}_n(x^\star)$ pour chacun de ces échantillons bootstrap. En déduire des réalisations bootstrap de la différence $\sqrt{n}\{\hat{q}_n - F^{-1}(\alpha)\}$,

$$\delta^\star = \sqrt{n}\{\hat{q}_n(x^\star) - \hat{q}_n\}\,.$$

c. En utilisant ces 2500 réalisations bootstrap $\delta^\star$, utiliser `ks.test` pour tester si l'hypothèse de normalité du théorème limite avec variance libre (donc estimée) est bien vérifiée (au niveau 95%).

d. Tracer la fonction de répartition empirique associée à l'échantillon bootstrap des $\delta^\star$ et superposer la distribution normale estimée.

Exercice 6
Lorsque une densité de probabilité f est un mélange de lois normales

$$f(x) = p\ \varphi(x; 0, 1) + (1 - p)\varphi(x; 3, 1),$$

où $\varphi(\cdot; \mu, \sigma^2)$ correspond à la densité de la loi $\mathcal{N}(\mu, \sigma^2)$, on peut simuler un échantillon Z de distribution f et de taille `nech` de la manière suivante :

```
> p=0.6
> nech=1000
> Z=replicate(nech, rnorm(1, sample(c(0,3),prob=c(p,1-p)),1))
```

On s'intéresse à la quantité $P = \mathbb{P}(X < 2)$ quand $X \sim f$ et à son approximation par plusieurs méthodes de Monte Carlo.

a. Déterminer la valeur exacte de P :

1. par une intégration numérique en utilisant `integrate` ;
2. en utilisant le fait que f est un mélange de lois normales et en utilisant `pnorm`.

b. A partir d'un unique échantillon de taille `nech` simulé suivant f, et en remarquant que l'estimation de P correspond à l'estimation de la fonction de répartition de f en 2, donner un intervalle de confiance normal sur P à 95%.

c. Simuler $N = 500$ échantillons, chacun de taille `nech`, simulés suivant f. En déduire l'histogramme des N évaluations de P puis un intervalle de confiance à 95% sur P.

d. Tester si l'échantillon des N évaluations de P produits à la question **c.** est effectivement dans une famille normale en utilisant `ks.test`.