Examen final du 11 janvier 2010
Séance de 11 heures à 13h45

Préliminaires

Cet examen est à réaliser sur ordinateur en utilisant le langage R et à rendre simultanément sur papier pour les réponses détaillées et sur fichier informatique pour les fonctions R utilisées. Les fichiers informatiques seront à sauvegarder suivant la procédure ci-dessous et seront pris en compte pour la note finale. Toute duplication de fichiers R sera pénalisée par un zéro. L'absence de document enregistré donnera lieu à une note nulle sans possibilité de contestation.
Pour cet examen, vous devez remettre vos fichiers en ligne sur Intercours, suivant les étapes:

1. Enregistrez d'abord vos fichiers sur l'ordinateur, sans utiliser d'accents ni d'espace, ni de caractères spéciaux.
2. Connectez-vous à Intercours http://intercours.dauphine.fr (ou http://www.ent.dauphine.fr et onglet "cours en ligne" - un clic sur l'image Intercours) Utilisez les identifiants de l'ENT (ceux de votre mail Dauphine)
3. Cliquez sur le cours intitulé "Examen (Christian Robert)" (dans la liste des cours à gauche)
4. Cliquez sur "Examen" au centre de la page
5. Vous allez maintenant soumettre vos fichiers. Pour cela, cliquez sur "Ajouter des pièces jointes" et sélectionnez votre premier fichier. Votre fichier appairait maintenant comme une pièce jointe en dessous du cadre "soumission". Si vous avez plusieurs fichiers à remettre, cliquez de nouveau sur "Ajouter des pièces jointes" pour sélectionner les suivants.
6. Une fois que vous aurez soumis vos fichiers, il ne sera plus possible de recommencer la procédure ou de modifier vos fichiers. Vérifiez que vos fichiers apparaissent bien comme des pièces jointes sous le cadre "soumission". Cliquez sur le bouton SOUMETTRE et OK. Un message de confirmation apparaît vous indiquant l'heure de la soumission.

Les documents disponibles sur votre compte informatique sont autorisés, ainsi que les documents papier du cours et l'aide en ligne de R. L'utilisation de tout service de messagerie ou de mail est interdite et, en cas d'utilisation avérée, se verra sanctionnée par une note nulle pour les deux parties. La copie papier de l'examen doit être rendue à la sortie de la salle informatique.

On considère la distribution de Student à deux degrés de liberté, non centrée de paramètre μ, densité

$$g(x;\mu) = \frac{1}{(2+(x-\mu)^2)^{3/2}}\,,$$

et fonction de répartition

$$G(x;\mu) = \frac{1}{2}\left(1 + \frac{x-\mu}{\sqrt{2+(x-\mu)^2}}\right)\,, \quad x \in \mathbb{R}\,.$$

On pourra utiliser dans la suite le fait que si Y est une variable aléatoire gaussienne centrée réduite et V est une variable aléatoire du χ^2 à deux degrés de liberté, alors

$$X = \frac{Y+\mu}{\sqrt{V/2}}$$

est une variable aléatoire de Student à deux degrés de liberté de paramètre μ. La moyenne de la densité $g(.;\mu)$ est approximativement μ.

1 Simulation de réalisations de la loi de Student

1. **Méthode 1**:
   (i) Proposer une méthode de simulation de cette loi reposant sur le principe d'inversion générique.
   (ii) Ecrire une fonction rstudent1 ayant pour argument d'entrée n le nombre de réalisations et le paramètre μ et fournissant en sortie le vecteur des n réalisations.
   (iii) Simuler un 1000-échantillon $X^{(1)}$ avec cette fonction. Montrer la pertinence de la fonction rstudent1 en traçant l'histogramme de $X^{(1)}$ ainsi que la vraie densité.

2. **Méthode 2**:
   (i) Déduire une deuxième méthode de simulation de réalisations d'une loi de Student à deux degrés de liberté et de paramètre μ reposant sur les simulations selon une loi normale centrée réduite et une loi du χ^2 (cf `rchisq`) à deux degrés de liberté.
   (ii) Ecrire une fonction rstudent2 fournissant en sortie le vecteur des n réalisations.
   (iii) Simuler un 1000-échantillon $X^{(2)}$ avec cette fonction. Montrer la pertinence de la fonction rstudent2 en traçant l'histogramme de $X^{(2)}$ ainsi que la vraie densité.

3. Dans la suite, on utilisera la fonction rt implémentée dans R pour générer des échantillons selon une loi de Student:

   ```
   > library(stats)
   > X = rt( ... à compléter par vos soins)
   ```

   Préciser les paramètres de cette fonction permettant de simuler un échantillon X de taille n dans la loi de Student à deux degrés de liberté et de paramètre μ.

2 Utilisation des réalisations de la loi de Student

La densité de Student g peut-elle être utilisée pour générer par acceptation rejet un échantillon de chacune des lois suivantes?

1. Gaussienne,
2. Exponentielle,
3. de Cauchy.

Soit f la densité de probabilité définie de la façon suivante:

$$f(x) = C \times \frac{1+\cos(2x)}{1+(x-\pi)^4}$$

où C est la constante de normalisation.
On cherche un paramètre μ et une constante $M > 0$ tels que $\forall x$,

$$\frac{1+\cos(2x)}{1+(x-\pi)^4} \leq M g(x;\mu)\,.$$

1. Justifier pourquoi la majoration précédente est possible.
2. Tracer les graphes des fonctions $x \to \frac{1+\cos(2x)}{1+(x-\pi)^4}$ et $x \to g(x;\mu)$ pour certaines valeurs de μ. Donner (et justifier) la valeur du paramètre μ la mieux adaptée.
3. Déterminer ensuite M. On pourra utiliser R (aucune justification est demandée), par exemple en traçant la courbe du rapport de deux fonctions.
4. En déduire une méthode de simulation reposant sur un n-échantillon X, $n = 10000$, de loi de Student généré à l'aide de la commande `rt`. On notera fARStudent la fonction correspondante.
5. Illustrer graphiquement la pertinence de votre méthode de simulation.

3 Calcul d'une intégrale

On cherche à calculer la constante de normalisation C.

1. Proposer une méthode de Monte Carlo reposant sur l'utilisation du n-échantillon X de loi de Student permettant de calculer C (on fera intervenir le taux d'acceptation).
2. Fournir un intervalle de confiance à 95% pour C (par le TCL).
3. Illustrer la convergence de votre méthode.

4 Fonction de répartition empirique

On utilise à partir d'ici le paramètre $\mu = 0$.

1. Tracer sur un même graphe les fonctions de répartition issues d'échantillons de tailles respectives $n = 50$, $n = 500$, $n = 5000$, $n = 50000$.
2. Ajouter sur le graphe la fonction de répartition théorique.
3. Commentez votre graphique.
4. A partir du 50000-échantillon, estimer le quantile de niveau 25%. Fournir un intervalle de confiance par bootstrap (on réalisera $B = 1000$ tirages bootstrap).

5 Bootstrap

1. Récupérer l'échantillon x par :

```
library(utils)
data(Nile)
x=Nile
```

2. Commenter rapidement ces données (que représentent-elles, taille de l'échantillon, moyenne, variance).
3. On suppose que les observations sont les réalisations d'un échantillon $\mathbf{X}_n = (X_1 \dots X_n)$ de taille $n = 100$ d'une variable aléatoire X. On suppose que les X_i suivent une loi de Student $X_i \sim g(.; \mu, \nu)$, de paramètres μ (centrage) et ν (degré de liberté). Soient les deux estimateurs suivants de μ et ν:

$$\hat{\mu} = 0.95 \times \frac{1}{n} \sum_i X_i, \quad \hat{\nu} = \frac{2\widehat{v}}{\widehat{v} - (1 + \hat{\mu}^2)},$$

où on a $\widehat{v} = \frac{1}{n} \sum_i X_i^2$.

4. Donner des intervalles de confiance à 95% par bootstrap pour μ et pour ν (on réalisera $B = 1000$ tirages bootstrap).
5. Afficher l'histogramme des données. Penser à régler le nombre de classes de l'histogramme de façon à optimiser la représentation. Tracer sur l'histogramme la densité de Student avec les paramètres estimés.
6. Afficher sur un nouveau graphique une estimation non-paramétrique de la densité, en décrivant votre choix de noyau. On utilisera la fonction `density`.
7. Que pensez-vous de l'hypothèse selon laquelle les données observées suivent une loi de Student ?

6 Compréhension du code R

Dans le code ci-dessous, on cherche à obtenir la distribution bootstrap du maximum d'un echantillon de `2*n` points, lorsqu'on observe `n=83` valeurs d'un échantillon dénoté `obs`. Identifier les erreurs de programmation et corriger ce code pour obtenir une évaluation correcte de la moyenne et de la variance de ce maximum.

```
bootmin=function(B=10^4){

   boot=matrix(0,ncol=n,row=B)
   for (T in 1:B)
     boot[T,]=sample(ord,2*n,rep=T)
     }

   temp=apply(boot,1,sort)
   bootmax=temp[,2n]

   list(mean(samax),var(samax)
   }
```

En simulant `obs` par `obs=rnorm(n)`, en déduire les valeurs numériques de cette moyenne et de cette variance.