Examen NOISE

On considère la distribution de Weibull généralisée de paramètres (k, λ, θ) de densité suivante:

$$g(x;k,\lambda,\theta) = \frac{k}{\lambda}\left(\frac{x-\theta}{\lambda}\right)^{k-1} \exp\left\{-\left(\frac{x-\theta}{\lambda}\right)^{k}\right\} \mathbb{I}_{x>\theta}$$

La fonction de répartition vaut alors:

$$G(x;k,\lambda,\theta) = \begin{cases} 1-\exp\left\{-\left(\frac{x-\theta}{\lambda}\right)^{k}\right\} & \text{si } x>\theta \\ 0 & \text{sinon} \end{cases}$$

Remarques Si $X \sim W(k,\lambda,\theta)$ alors

$$E[X] = \lambda\Gamma(1+\frac{1}{k})+\theta \quad , \quad Var[X] = \lambda^2\Gamma(1+\frac{2}{k}) - \left(\lambda\Gamma(1+\frac{1}{k})\right)^2 \quad q_2 = (\lambda \ln 2)^{1/k}+\theta$$

où Γ est la fonction Gamma (gamma dans R) et q_2 est la médiane. Attention, la fonction logarithme népérien (ln) est donnée par log dans R.

1 Simulation de réalisations de la loi de Weibull

1. **Méthode 1: à partir de réalisations d'une loi $\mathcal{U}(0,1)$:**
   - Proposer une méthode de simulation de cette loi reposant sur le *principe d'inversion générique*.
   - Ecrire une fonction rweibull1 ayant pour argument d'entrée n le nombre de réalisations et les paramètres (k,λ,θ) et fournissant en sortie le vecteur des n réalisations.
2. **Méthode 2: à partir de réalisations d'une loi $\mathcal{N}(0,1)$:**
   - On remarque que si $X \sim W(k,\lambda,\theta)$ alors $Y = 2(\frac{X-\theta}{\lambda})^k \sim \chi^2_2$. En déduire une deuxième méthode de simulation de réalisations d'une loi de Weibull de paramètres (k,λ,θ) reposant sur des simulations de lois normales centrées réduites[1]
   - Ecrire une fonction rweibull2 ayant pour argument d'entrée n le nombre de réalisations et les paramètres (k,λ,θ) et fournissant en sortie le vecteur des n réalisations.
3. **Comparaison des méthodes:** Démontrer analytiquement que les deux méthodes précédentes sont parfaitement équivalentes.

Le package stat comporte une fonction permettant de simuler des variables aléatoires suivant une loi de Weibull avec $\theta = 0$

Dans la suite on utilisera cette fonction.

```
> library(stat)
> X = theta + rweibull( ... \'a compl\'eter par vos soins)
```

[1] On rappelle que si $Z_1 \dots Z_n$ sont indépendantes et identiquement distribuées de loi $\mathcal{N}(0,1)$ alors $X = \sum_{i=1}^{n} Z_i^2$ est distribuée selon une loi du χ^2 à n degrés de liberté.

2 Utilisation des réalisations de la loi de Weibul

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

$$f(x) = C \times (x+2)(\cos(x+2))^2 \exp\{-(x+2)^2\}\mathbb{I}_{x>-2}$$

où C est la constante de normalisation.

1. Proposer un ensemble de paramètres (k, λ, θ) et une constante M tels que $\forall x > -2$,

$$(x+2)(\cos(x+2))^2 e^{-(x+2)^2} \leq M g(x; k, \lambda, \theta)$$

2. En déduire une méthode de simulation reposant sur un n-échantillon de loi de Weibull. On notera fAR la fonction correspondante.

3. 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 d'un échantillon de taille $n = 10000$ de loi de Weibull, permettant de calculer C.

2. Illustrer la convergence de votre méthode.

3. Founir un intervalle de confiance à 95% pour C

4 Fonction de répartition empirique

On suppose que $\theta = 0$, $\lambda = 2$, $k = 3$.

1. Tracer sur un même graphe les fonctions de répartition issues d'échantillons de tailles respectives $n = 30$, $n = 100$, $n = 1000$, $n = 10000$

2. Ajouter sur le graphe la fonction de répartition théorique.

3. Commentez votre graphique.

4. A partir de l'échantillon de taille $n = 10000$ estimer la probabilité :

$$p = P(X \leq q_2) = P(X \leq (\lambda \ln 2)^{1/k} + \theta)$$

Fournir un intervalle de confiance pour p et comparer à la vraie valeur.

5. A partir de l'échantillon de taille $n = 10000$, fournir une estimation du quantile à 63% i.e. q tel que $P(X < q) = 0.63$. Quel lien peut-on faire entre ce quantile et λ ?

5 Bootstrap

Télécharger le jeu de données lynx

```
> data(lynx)
> x = lynx
```

On suppose que les observations sont les réalisations d'un échantillon $\mathbf{X_n} = (X_1 \dots X_n)$ de taille $n = 114$ d'une variable aléatoire X. On suppose que les $X_i \sim W(k, \lambda, \theta)$ avec (k, λ, θ) inconnus.

Notons $Q_{0.5}(\mathbf{X_n})$ la médiane empirique de l'échantillon $\mathbf{X_n}$ et $Q_{0.63}(\mathbf{X_n})$ le quantile empirique de l'échantillon $\mathbf{X_n}$ à 63%. On s'intéresse aux trois statistiques d'échantillonnage suivantes:

$$T = \min_{i=1\dots n}(\mathbf{X_n}) \quad , \quad L = Q_{0.63}(\mathbf{X_n} - T) \quad \text{et} \quad K = \frac{\ln(L \ln 2)}{\ln(Q_{0.5}(\mathbf{X_n}) - T)}$$

T, L et K sont des estimateurs respectifs de θ, λ et k.

1. Donner un intervalle de confiance à 95% par bootstrap pour (k, λ, θ)
2. Afficher l'histogramme des données et tracer sur l'histogramme la densité de Weibull avec les paramètres estimés. Penser à régler le nombre de classes de l'histogramme de façon à optimiser la représentation.
3. Afficher également une estimation non-paramétrique de la densité, en décrivant votre choix de noyau.
4. Que pensez-vous de l'hypothèse selon laquelle les données observées suivent une loi de Weibull ?

6 Compréhension du code R

1. Dans le code ci-dessous, on cherche à simuler (x, y) tel que $x \sim \mathcal{N}(0, 1)$ et $y|x < -2 \sim \mathcal{N}(-2, 1)$, $y|x > -2 \sim \mathcal{N}(2, 1)$. Identifier les erreurs de programmation et corriger ce code pour obtenir le résultat voulu.

```
simulator=function(n==1){

   x=y=NUL
   for (t in 1:n){

     z=rnorm(1)
     if (z<-2) w=rnorm(-2)
     else w=rnorm(2)
     x=c(c,z)
     y=c(y,w)
     }
   list(x,y)
   }
```

   En déduire une approximation de la moyenne et de la variance de y.

2. 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.