Université de Kairouan **Année Universitaire 2015/2016**
ISMAI **M2 : IF**
Enseignant : Med Essaied Hamrita

TP1 : Simulation & Modélisation

Exercice 1 : Distribution de Pareto.
Si $\alpha > 0$ est une constante et X une $v.a$ avec $d.d.p$: $f(x) = \alpha x^{-(\alpha+1)}$, $x \geq 0$, alors X est dite une $v.a$ suivant la loi de Pareto de paramètre α.
1) Montrer que la fonction de répartition de cette loi est :

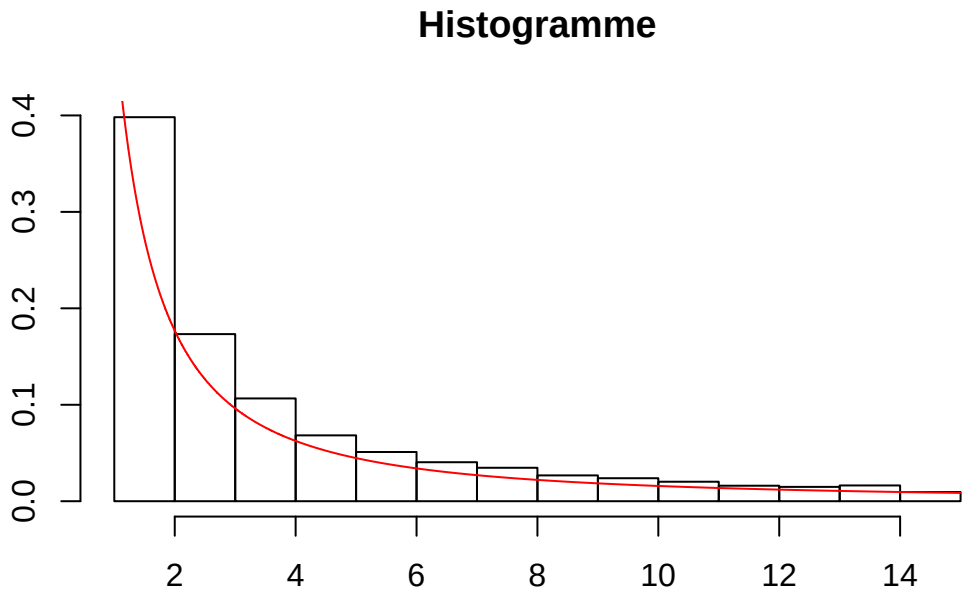
$$F(x) = 1 - x^{-\alpha}, \quad x \geq 1$$

2) Montrer que $F^{-1}(u) = (1-u)^{-1/\alpha}$ et écrire une fonction `R`, `rpareto(n,a)` qui génère une suite de $v.a$ suivant la loi de Pareto de paramètre a de taille n.

```
rpareto <- function(n,a){
     u <- runif(n)
     return((1-u)^(-1/a))
}
```

3) Produire une suite de 10000 observations issues de la loi de Pareto de paramètre $a = 0.5$. Représenter l'histogramme et calculer la moyenne empirique.
La figure suivante représente l'histogramme de l'échantillon fabriqué (limité seulement aux valeurs inférieures à 15 pour des raisons de lisibilité) avec la courbe de densité d'équation $f(x) = 0.5x^{-1.5}$ en surimpression. On l'obtient avec les instructions que voici :

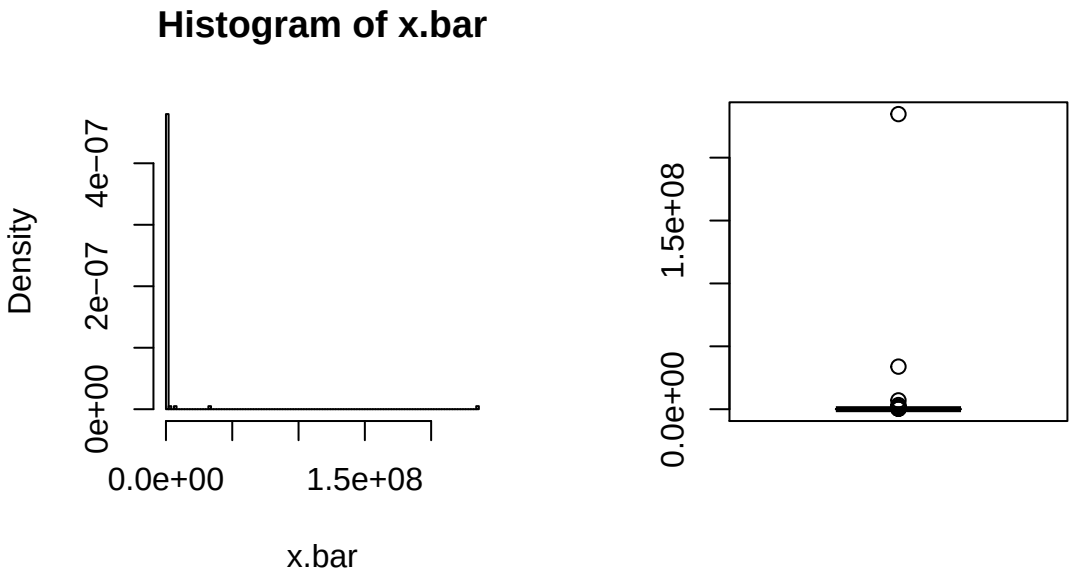
```
X1<-rpareto(10000,0.5)
hist(X1[X1<15],prob=T,xlab="",ylab="",main="Histogramme")
y<-sort(X1[X1<15])
lines(y,0.5*y^{-1.5},col="red")
```



```r
# La moyenne empirique
mean(X1)

## [1] 53371.3
```

Répéter ceci M fois et étudier la variabilité de la moyenne empirique.

```r
M=100
pp<-(matrix(0,nc=100,nr=10000))
for(i in 1:M){
   pp[,i]<-rpareto(10000,0.5)
}
x.bar<-colMeans(pp)
par(mfrow=c(1,2))
hist(x.bar, 100, prob=T)
boxplot(x.bar)
```



4) Calculer les moyennes empiriques basées sur les simulations lorsque $n = 10^2$, $n = 10^3$, $n = 10^4$, $n = 10^5$ et discuter le résultat.

```r
N<-c(10^2,10^3,10^4,10^5)
x11<-rpareto(N[1],0.5)
x12<-rpareto(N[2],0.5)
x13<-rpareto(N[3],0.5)
x14<-rpareto(N[4],0.5)
XBar<-c(mean(x11),mean(x12),mean(x13),mean(x14))
XBar

## [1]    292.3259   1704.6162   3660.9848 552126.4500
```

Exercice 2 : Loi exponentielle

Soit $(X_1, X_2, 0 \ldots, X_n)$ un échantillon provenant d'une loi de densité exponentielle de paramètre θ, notée $\varepsilon(\theta)$.

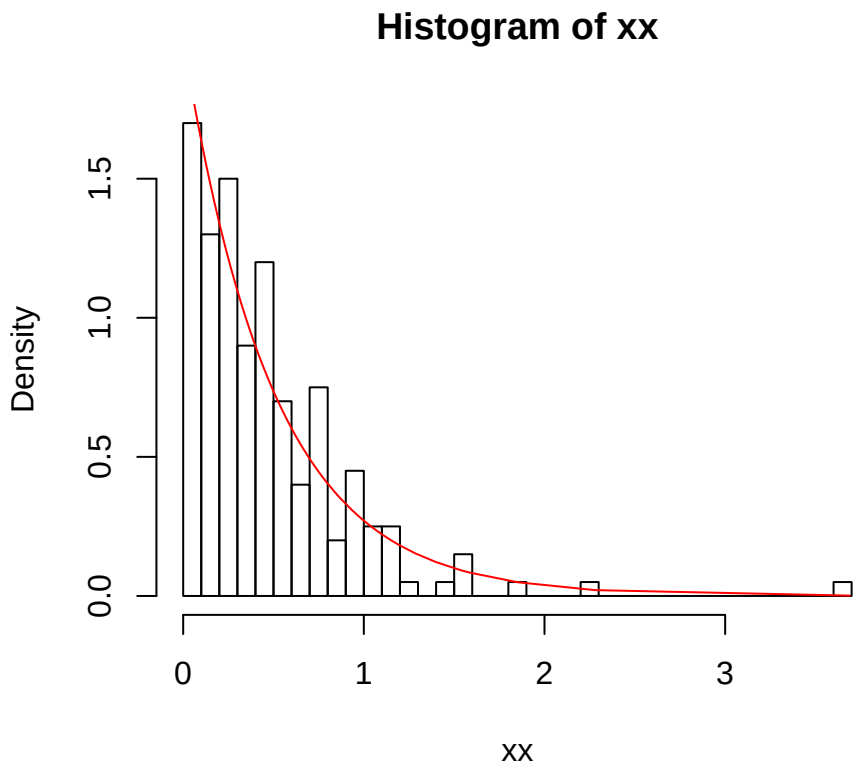
1) Déterminer la fonction de répartition de la $v.a.$ X. En déduire l'expression permettant de simuler cette variable en suivant la méthode d'inversion.

2) Écrire une fonction `R` qui simule une loi exponentielle. On note cette fonction `Rexp`.

```r
Rexp<-function(n,lambda){
  u<-runif(n)
  X<-(-1/lambda)*log(u)
  return(X)
}
```

3) Générer une réalisation d'un échantillon de taille $n = 200$ et de loi $\varepsilon(\theta_0)$ pour $\theta_0 = 2$.

```r
set.seed(123)
xx<-Rexp(200,2)
hist(xx,breaks=50,prob=T)
yy<-sort(xx)
lines(yy,2*exp(-2*yy),col="red")
```

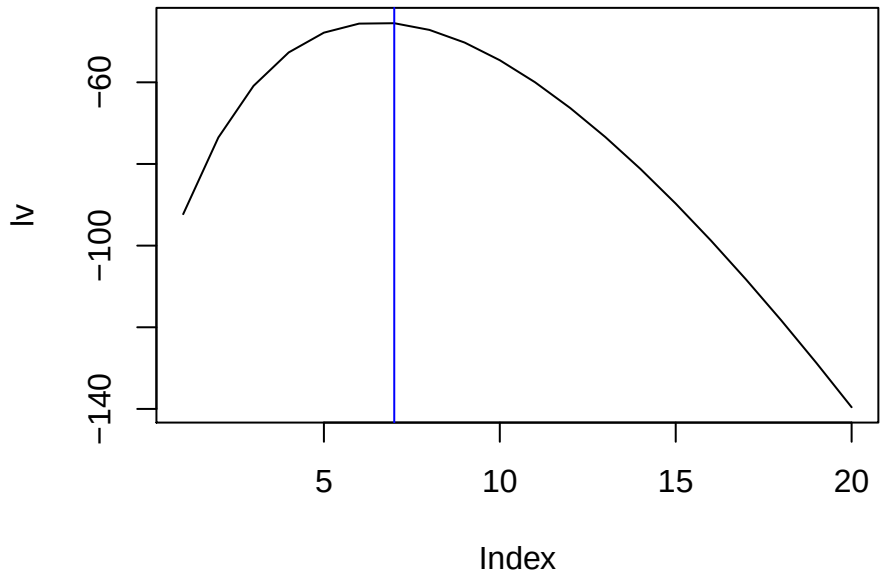


4) Créer une fonction appelée `LV` de la variable θ qui calcule la log vraisemblance en le paramètre θ pour l'échantillon précédant.

```r
LV<-function(x,theta){
  n<-length(x)
  return(n*log(theta)-theta*sum(x))
}
```

5) Varier θ et représenter graphiquement la fonction LV.

```r
theta<-seq(1,5,len=20)
lv<-LV(xx,theta)
plot(lv,type="l")
abline(v=which.max(lv),col="blue")
```

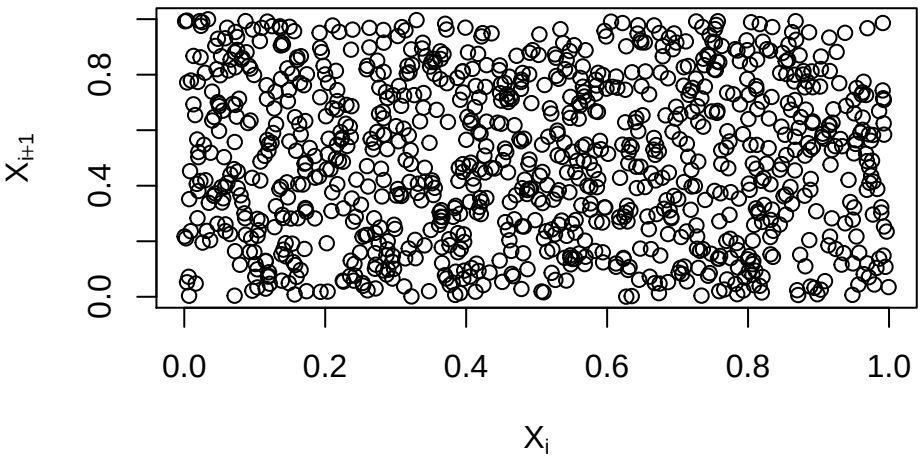


5) Chercher l'EMV de θ pour l'échantion précédent.

```
(theta.chap<-theta[which.max(lv)])

## [1] 2.263158

1/mean(xx)

## [1] 2.16699
```

Exercice 3 : Étant donné un vecteur des nombres psudo-aléatoires $x_1, x_2, \ldots, x_n$, on peut tracer la courbe (x_i, x_{i+1}) pour $i = 1, 2, \ldots, n-1$ afin de tester l'indépendance des x_i.
Construire un tel graphique en créant des nombres pseudo-aléatoires uniformes.

```
n<-1000
u<-runif(n)
plot(u[1:(n-1)],u[2:n],xlab=expression(X[i]),ylab=expression(X[i+1]))
```



Exercice 4 : Soit la densité f d'une variable aléatoire définie par :

$$f(x) = \begin{cases} \frac{1}{x^2}, & \text{si } x > 1 \\ 0, & \text{sinon.} \end{cases}$$

1) Déterminer la fonction de répartition F de la variable X.
2) Écrire une fonction `R` pour simuler cette variable par la transformation inverse.
3) Construire un histogramme de 10000 réalisations et la densité f sur le même graphique pour tester votre programme.

```
###############
# Question 2
#
simf<-function(n){
  u<-runif(n)
  return(1/(1-u))
}
################
# Question 3
#
X<-simf(10000)
hist(X,freq=FALSE,breaks=seq(0, max(X)+1, 0.1),xlim=c(0,10),
     ylim=c(0,1))
x<-seq(0,10,len=1000)
f<-ifelse(x>1,1/x^2,0)
lines(x,f,col=2)
```