INSA – MS ESD Année 2019-2020
ROUFFIAC Jean-Eudes

STAT 1 : Sélection de variables et validation de modèles

Rapport de TP n°2

Titre : *Notions de base en estimation paramétrique et mise en œuvre avec le logiciel R*

Introduction

L'objectif de ce TP 2 est multiple :

- étudier par simulations le comportement des estimateurs ponctuel et par intervalle de la moyenne et de la variance dans le cadre d'échantillons gaussiens simulés.
- étudier par simulations le comportement des estimateurs ponctuels des coefficients d'asymétrie et d'aplatissement dans le cadre d'échantillons gaussiens simulés.
- étudier le comportement de l'estimateur du maximum de vraisemblance dans le cadre de la loi de Weibull.

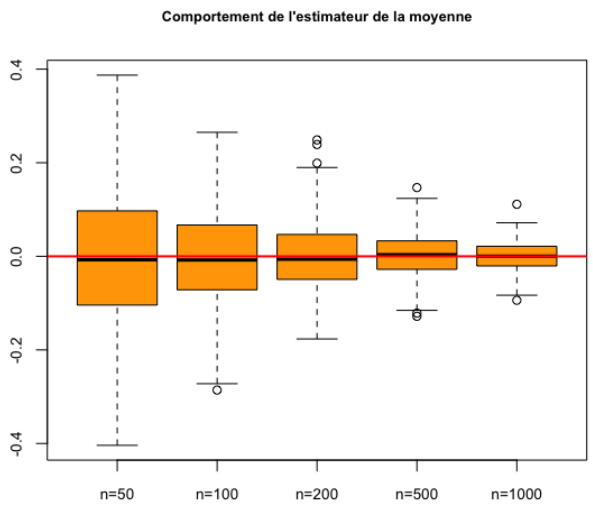
1 Estimation ponctuelle de la moyenne et de la variance

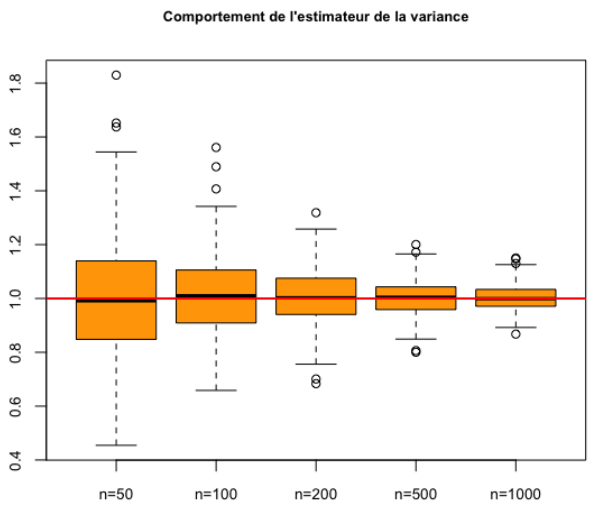
L'objet de cette première partie est d'étudier par simulations le comportement à distance finie de l'estimateur de la moyenne μ et de la variance σ d'une gaussienne. Pour cela, on commencera par simuler 400 échantillons gaussiens de taille 1000 avec $\mu = 0$ et $\sigma = 2$.
Grâce à la fonction rnorm, on stock les valeurs données par rnorm dans une matrice. Cette matrice a donc un format de 1000 lignes et 400 colonnes.

Pour chaque colonne de données, on construit la suite de moyennes succesives $\bar{x}_1, ..., \bar{x}_{1000}$ (avec $\bar{x}_n = (1/n)\sum_{j=1}^{n} x_j$) et la suite des variances successives $s_2^2, ..., s_2^{1000}$ (avec

$$s_n^2 = \frac{1}{n-1}\sum_{j=1}^{n}(x_j - \bar{x}_n)^2$$

on prendra $s_1^2 = 0$). On stocke les valeurs des moyennes et des variances successives dans deux matrices M et S2. On peut maintenant comparer le comportement des deux estimateurs en traçant les boites à moustaches des 400 estimations pour les tailles d'échantillons $n = 50, 100, 200, 500, 1000$.





Pour de faibles valeurs de n, les fluctuations sont importantes. Mais ces fluctuations se réduisent lorsque la taille de l'échantillon augmente. De même, la taille des boxplots diminuent quand la taille des échantillons augmentent. Donc plus on a des observations, meilleur est l'estimateur. La variances des estimateurs tend vers 0 quand n tend vers l'infini. Les estimateurs convergent vers la valeur théorique, à savoir 0 pour la moyenne et 1 pour la variance.

2 Intervalle de confiance pour la moyenne

Lorsqu'on construit un intervalle de confiance au niveau de confiance 95% et qu'on affirme que le paramètre inconnu appartient à cet intervalle de confiance, on prend un risque de 5% de se tromper.
On souhaite illustrer cela pour la moyenne d'un échantillon gaussien en construisant 400 intervalles de confiance et en comptabilisant l'appartenance ou non de la moyenne à chaque intervalle. L'objectif étant de comparer le pourcentage empirique de non-appartenance au risque théorique de 5%. Pour cela, on va construire pour chacun des échantillons précédemment simulés et pour une taille d'échantillon n = 50, 100, 200, 500, 1000, l'intervalle de confiance au niveau de confiance 95% du paramètre σ et comptabiliser l'appartenance ou non de σ à cet intervalle.

Valeur de N	50.0	100	200	500.0	1000.0
Nombre d'estimation en dehors de l'intervalle	22.0	20	28	26.0	22.0
en %	5.5	5	7	6.5	5.5

On remarque que le pourcentage d'estimation en dehors de l'intervalle est proche de 5% quelle que soit la valeur de n. On peut aussi noter que le % d'estimation en dehors de l'intervalle ne dépend pas de n.

3 Estimation ponctuelle des coefficients d'asymétrie et d'aplatissement

L'objet de cette partie est d'étudier par simulations le comportement à distance finie des estimateurs des coefficients d'asymétrie et d'aplatissement d'une gaussienne (qui valent respectivement 0 et 3).

Premièrement, on veut un estimateur pour chacun des paramètres β et γ suivants :
Soit X une variable aléatoire réelle d'espérance μ et de variance σ^2 (on note σ l'écart-type) possédant un moment d'ordre 4 fini, ie $\mathbb{E}(X^4) < \infty$. On appelle :
— coefficient d'asymétrie (skewness) de la variable X, la quantité définie par :

$$\beta = \mathbb{E}\left[\left(\frac{X-\mu}{\sigma}\right)^3\right] = \frac{\mathbb{E}[(X-\mu)^3]}{(\mathbb{E}[(X-\mu)^2])^{3/2}}$$

— coefficient d'aplatissement (kurtosis) de la variable X, la quantité définie par :

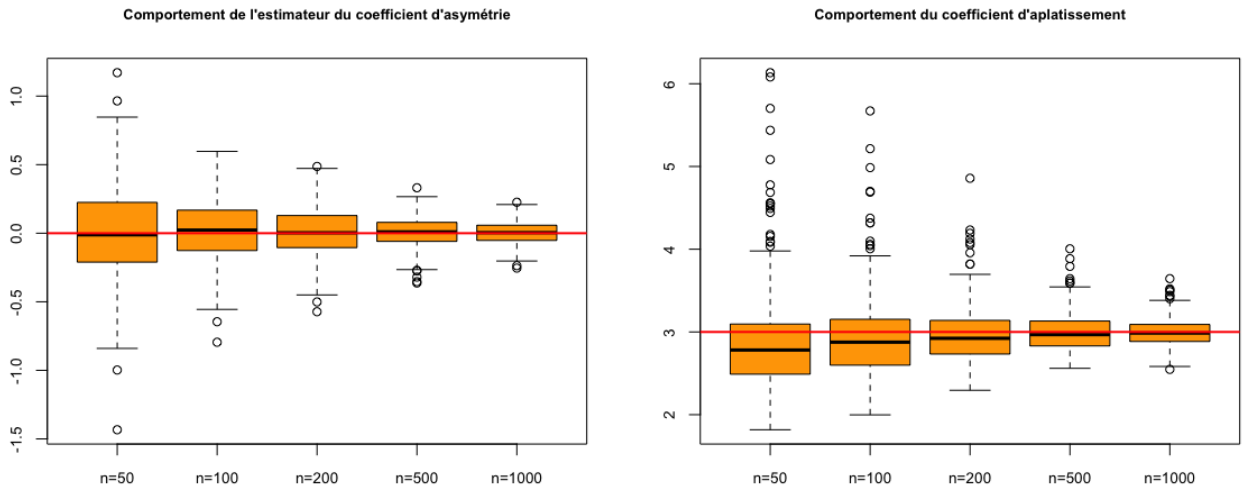
$$\gamma = \mathbb{E}\left[\left(\frac{X-\mu}{\sigma}\right)^4\right] = \frac{\mathbb{E}[(X-\mu)^4]}{(\mathbb{E}[(X-\mu)^2])^2}$$

On propose des estimateurs pour ces deux paramètres :

$$\beta = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{X_i - \bar{X}_n}{\sqrt{S2_n}}\right)^3$$

$$\gamma = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{X_i - \bar{X}_n}{\sqrt{S2_n}}\right)^4$$

Pour analyser le comportement des estimateurs de skewness et kurtosis, nous allons procéder de la même façon que pour la partie 1 (simulation de 400 échantillons de taille 1000 d'une loi gaussienne).



L'examen des graphiques précédents conduit aux remarques suivantes :

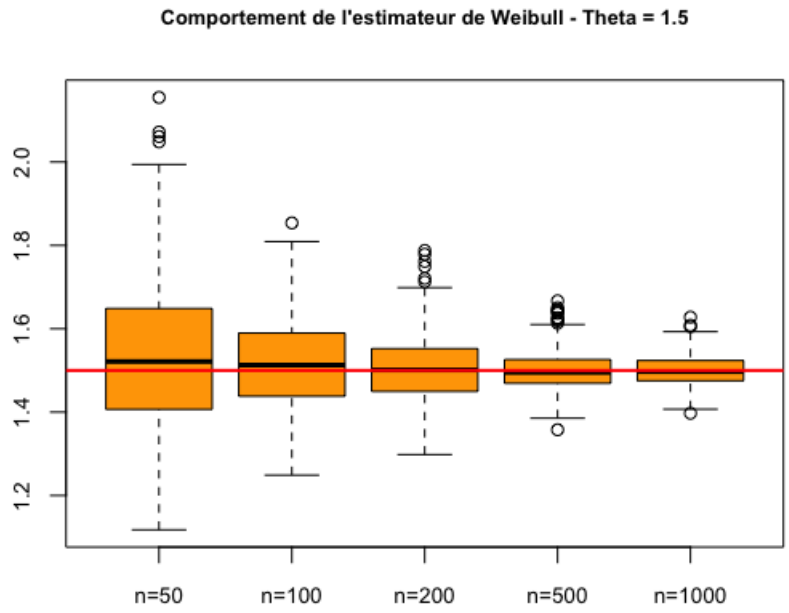
— On constate des fluctuations d'échantillonage importantes. En effet, pour $n = 50$, l'estimation du skewness varie entre -0.9 et 1 et celui du kurtosis entre 1.9 et 6. L'estimation des paramètres peut donc être très mauvaise pour des petites tailles d'échantillon. En revanche, ces fluctuations se réduisent lorsque la taille des échantillons augmentent.

— On constate que la taille des boites à moustaches se réduit avec l'augmentation de la taille de l'échantillon, ce qui montre que la variabilité de l'estimation diminue avec l'augmentation de la taille de l'échantillon. Cela illustre le fait que la variance de des estimateurs tend vers 0 quand n tend vers l'infini.

— On remarque pour le Kurtosis que l'estimation est légèrement biaisée pour les petites tailles d'échantillons car la moyenne des 400 estimations est plus petite que 3. Cela illustre le fait que l'estimateur n'est pas sans biais, mais asymptotiquement sans biais.

4 Estimateur du maximum de vraisemblance dans le cadre de la loi de Weibull standard

L'objectif de cette partie est de programmer l'estimateur du maximum de vraisemblance du paramètre θ de la loi de Weibull standard. Pour cela, on simule 400 échantillons de taille $n = 1000$ d'une loi de Weibull standard de paramètre $\theta = 1.5$. On s'intéresse alors au comportement de l'estimateur du maximum de vraisemblance pour différentes tailles d'échantillon. Le calcul des estimations est obtenu en 8 itérations en moyenne pour une précision à 10^{-6}.



L'examen du graphiques précédent conduit aux remarques suivantes :

- On constate des fluctuations d'échantillonage importantes. En effet, pour $n = 50$, l'estimation du paramètre varie entre 1.1 et 2.2. L'estimation du paramètre peut donc être très mauvaise pour des petites tailles d'échantillon. En revanche, ces fluctuations se réduisent lorsque la taille des échantillons augmentent.
- On constate que la taille des boites à moustaches se réduit avec l'augmentation de la taille de l'échantillon, ce qui montre que la variabilité de l'estimation diminue avec l'augmentation de la taille de l'échantillon. Cela illustre le fait que la variance de l'estimateur tend vers 0 quand n tend vers l'infini.
- On remarque que l'estimation est légèrement biaisée pour les petites tailles d'échantillons car la moyenne des 400 estimations est plus grande que 1.5. Cela illustre le fait que l'estimateur n'est pas sans biais, mais asymptotiquement sans biais.

5 Estimateur du maximum de vraisemblance dans le cadre de la loi de Weibull générale

La loi de Weibull générale dépend de deux paramètres. Le but de cette partie est de calculer l'estimateur du maximum de vraisemblance des deux paramètres de la loi de Weibull en utilisant l'algorithme itératif de Newton-Raphson.

La densité de la loi de Weibull est donnée, pour $x > 0$, par :

$$f(X) = \frac{\theta}{\lambda}\left(\frac{x}{\lambda}\right)^{\theta-1} e^{-(x/\lambda)^{\theta}}$$

où :
— $\theta > 0$ est le paramètre de forme
— $\lambda > 0$ est le paramètre d'échelle
Dans un premier temps, nous écrivons la vraisemblance L :

$$L(x,\theta,\lambda) = \frac{\theta^n}{\lambda^n}\left[\prod_{j=1}^{n}\left(\frac{x_j}{\lambda}\right)^{\theta-1}\right]\exp\left(-\frac{1}{\lambda^{\theta}}\sum_{j=1}^{n}x_j^{\theta}\right)$$

puis la log-vraisemblance LL :

$$LL(x,\theta,\lambda) = \log L(x,\theta,\lambda) = n\log(\theta) - n\log(\lambda) + (\theta-1)\sum_{j=1}^{n}\log\left(\frac{x_j}{\lambda}\right) - \sum_{j=1}^{n}\left(\frac{x_j}{\lambda}\right)^{\theta}$$

L'estimation du maximum de vraisemblance des paramètres θ et λ est donnée par :

— $\theta_n = arg\min_{\theta\in\mathbb{R}^+}(-LL(x,\theta,\lambda))$
— $\lambda_n = arg\min_{\lambda\in\mathbb{R}^+}(-LL(x,\theta,\lambda))$

On pose $g(\theta,\lambda) = -LL(x,\theta,\lambda)$.

L'équation

$$\nabla(g) = 0$$

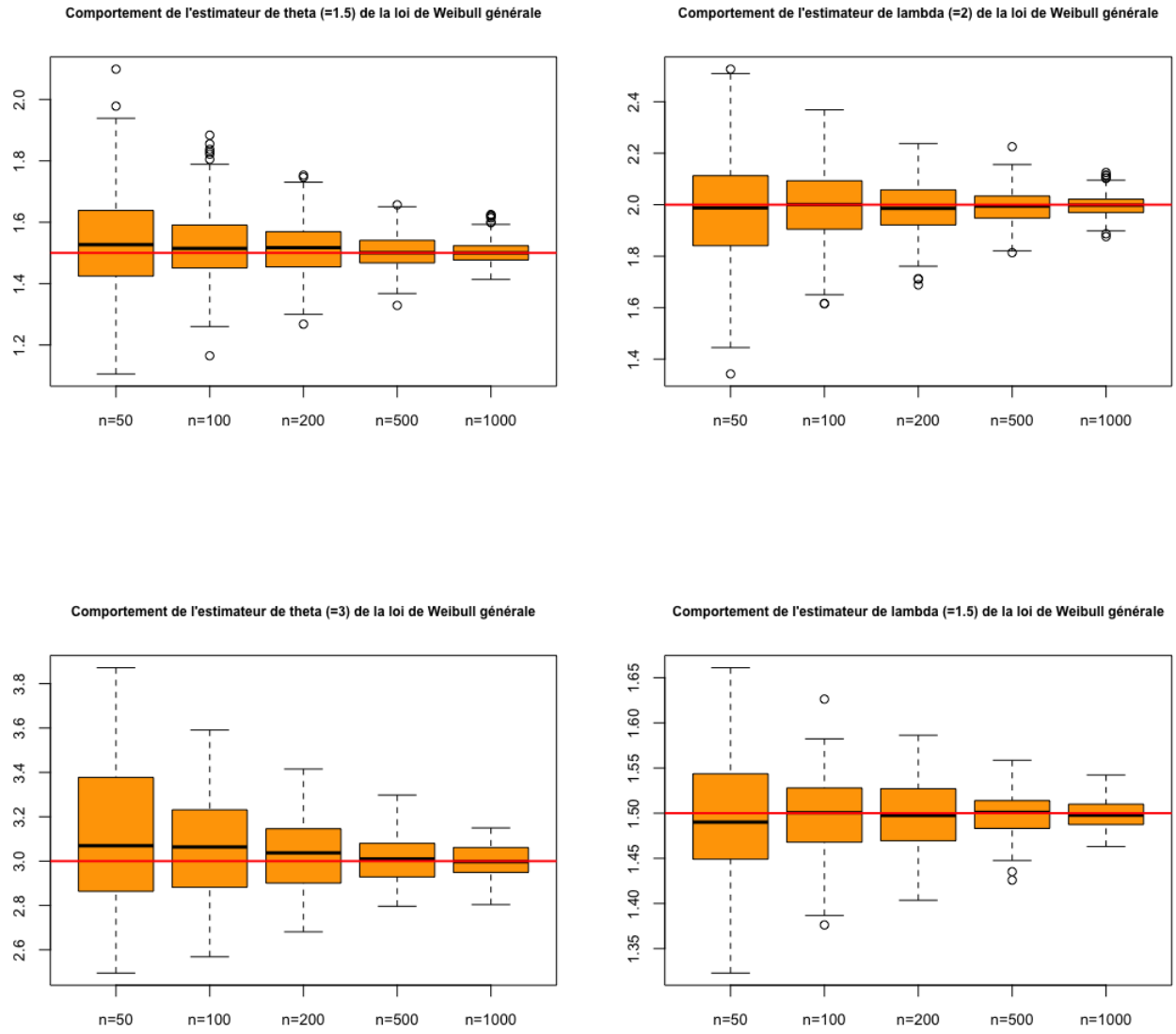
n'admet pas de solution explicite. Nous allons donc utiliser l'algorithme de Newton-Raphson pour obtenir une estimation des paramètres θ et λ. Pour ce faire, nous devons expliciter les dérivées partielles et secondes afin d'écrire le gradient et la matrice hessienne de la fonction g, qui seront utilisées dans l'algorithme. Voici ci-dessous le gradient et la matrice hessienne de la fonction g.

$$\nabla(g) = \begin{pmatrix}\frac{\partial g}{\partial\theta}\\ \frac{\partial g}{\partial\lambda}\end{pmatrix} = \begin{pmatrix}-\frac{n}{\lambda} - \sum_{j=1}^{n}\log\left(\frac{x_j}{\lambda}\right) + \sum_{j=1}^{n}\log\left(\frac{x_j}{\lambda}\right)\left(\frac{x_j}{\lambda}\right)^{\theta}\\ \frac{\theta}{\lambda}\left(n - \sum_{j=1}^{n}\log\left(\frac{x_j}{\lambda}\right)^{\theta}\right)\end{pmatrix}$$

$$H(g) = \begin{pmatrix}\frac{n}{\theta^2} + \sum_{j=1}^{n}\log^2\left(\frac{x_j}{\lambda}\right)\left(\frac{x_j}{\lambda}\right)^{\theta} & \frac{n}{\lambda} - \frac{1}{\lambda}\sum_{j=1}^{n}\left(\frac{x_j}{\lambda}\right)^{\theta} - \frac{\theta}{\lambda}\sum_{j=1}^{n}\log\left(\frac{x_j}{\lambda}\right)\left(\frac{x_j}{\lambda}\right)^{\theta}\\ \frac{n}{\lambda} - \frac{1}{\lambda}\sum_{j=1}^{n}\left(\frac{x_j}{\lambda}\right)^{\theta} - \frac{\theta}{\lambda}\sum_{j=1}^{n}\log\left(\frac{x_j}{\lambda}\right)\left(\frac{x_j}{\lambda}\right)^{\theta} & -\frac{\theta}{\lambda^2}\left(n - \sum_{j=1}^{n}\left(\frac{x_j}{\lambda}\right)^{\theta}\right) + \frac{\theta^2}{\lambda^2}\left(\sum_{j=1}^{n}\left(\frac{x_j}{\lambda}\right)^{\theta}\right)\end{pmatrix}$$

L'écriture dans R du gradient et de la matrice hessienne de la fonction g ainsi que l'algorithme de Newton-Raphson se trouve en annexe.

On simule deux fois 400 échantillons de taille 1000 d'une loi de Weibull de paramètres $(\theta = 1.5, \lambda = 2)$ et $(\theta = 3, \lambda = 1.5)$, grâce à la fonction rweibull de R. On s'intéresse au comportement des estimateurs pour différentes tailles d'échantillon. On notera que le calcul des estimations est obtenu en une quinzaine d'itérations en moyenne, pour une précision à 10^{-6}.



L'examen des graphiques précédents conduit aux remarques suivantes (nous commenterons les résultats obtenus dans le cas des simulations pour la loi de Weibull de paramètres $(\theta = 3, \lambda = 1.5)$, les commentaires pour les paramètres $(\theta = 1.5, \lambda = 2)$ étant les mêmes. Cette deuxième simulation de la loi de Weibull était pour s'assurer de la bonne exécution de notre algorithme pour différentes valeurs des paramètres θ et λ) :

— On constate des fluctuations d'échantillonage importantes. En effet, pour $n = 50$, l'estimation de θ varie entre 2.4 et 3.8 et celle de λ entre 1.3 et 1.70. L'estimation des

paramètres peut donc être très mauvaise pour des petites tailles d'échantillon. En revanche, ces fluctuations se réduisent lorsque la taille des échantillons augmentent.

— On constate que la taille des boites à moustaches se réduit avec l'augmentation de la taille de l'échantillon, ce qui montre que la variabilité de l'estimation diminue avec l'augmentation de la taille de l'échantillon. Cela illustre le fait que la variance de l'estimateur tend vers 0 quand n tend vers l'infini.

— On remarque que l'estimation est légèrement biaisée pour les petites tailles d'échantillons car la moyenne des 400 estimations est plus grande que 3 pour θ et plus petite que 1.5 pour λ. Cela illustre le fait que l'estimateur de θ et λ n'est pas sans biais, mais asymptotiquement sans biais.

Annexe

```
#Fonction g. Demandee dans l'enonce, mais ne servira pas dans
   notre algorithme
g <- function(x, theta, lambda){
  n = length(x)
  return(-n*log(theta) + n * log(theta) - (theta - 1) * sum(log(x
     /lambda)) + sum((x/lambda)^theta))
}

#Gradient de la fonction g a partir des derivees partielles
   calculees
g_gradient <- function(x, theta, lambda){
  n = length(x)
  gradient = matrix(NA, nrow=2, ncol=1)
  gradient[1,1] = - n/theta - sum(log(x/lambda)) + sum(log(x/
     lambda)*(x/lambda)^theta)
  gradient[2,1] = (theta/lambda) * (n - sum((x/lambda)^theta))

  return(gradient)
}


#Matrice hessienne de la fonction g a partir des derivees
   partielles secondes calculees
g_hessienne <- function(x, theta, lambda){
  n = length(x)
  hessienne = matrix(NA, nrow=2, ncol=2)
  hessienne[1,1] = (n/theta*theta) + sum(log(x/lambda)*log(x/
     lambda)*(x/lambda)^theta)
  hessienne[1,2] = n/lambda - (1/lambda)*sum((x/lambda)^theta) -
     (theta/lambda)*sum(log(x/lambda)*(x/lambda)^theta)
  hessienne[2,1] = n/lambda - (1/lambda)*sum((x/lambda)^theta) -
     (theta/lambda)*sum(log(x/lambda)*(x/lambda)^theta)
  hessienne[2,2] = (-theta/lambda^2)*(n - sum((x/lambda)^theta))
     + (theta^2/lambda^2) * sum((x/lambda)^theta)

  return(hessienne)
}


#Fonction qui renvoie une estimation des parametres grace a l'
   algorithme de Newton-Raphson
est_teta_lambda <- function(x){
  z = matrix(data = 1, nrow = 2, ncol = 1)
  z1 = matrix(data = 0, nrow = 2, ncol = 1)
  k = 0
  z1 = z - solve(g_hessienne(x, z[1,1], z[2,1])) %*% g_gradient(x
     , z[1,1], z[2,1])
  while ((norm(z1 - z)) > 0.000001)  {
```

```
    z = z1
    z1 = z - solve(g_hessienne(x, z[1,1], z[2,1])) %*% g_gradient
       (x, z[1,1], z[2,1])
    k = k + 1

  }
  print(k)  #Affichage du nombre d'iterations
  return(z1)
}

#Parametres
theta = 1.5
lambda = 2

#Simulation de 400 echantillons de taille 1000
B <- 400 ; N <- 1000
dataW <- matrix(NA, ncol=B, nrow=N)
for(b in 1:B){
  dataW[,b] = rweibull(N,theta,lambda)
}


#Initialisation des matrices qui vont contenir les 4 estimations
   pour n= 50, 100, 200, 500, 1000
theta_general = matrix(NA, nrow=5, ncol=B)
lambda_general = matrix(NA, nrow=5, ncol=B)

valeur_n = list(50, 100, 200, 500, 1000)
for(j in 1:B){
  indice = 1
  for(i in valeur_n){
    resultat = est_teta_lambda(dataW[1:i,j]) #R cup re les
       esimations. Matrice de taille 2x1.
    theta_general[indice, j] = resultat[1] #Matrice qui contient
       les estimations de theta pour les diff rentes tailles d'
         chantillon
    lambda_general[indice, j] = resultat[2] #Matrice qui contient
        les estimations de lambda pour les diff rentes tailles d'
         chantillon
    indice = indice + 1 #Indice qui permet de remplir les
       matrices de r sultat
  }

}


#Affichage du boxplot Theta
boxplot(t(theta_general[c(1,2,3,4,5),]),
        names=c("n=50","n=100","n=200","n=500","n=1000"), col = "
           orange", cex.axis = 0.85, cex.main = 0.75, main = "
           Comportement␣de␣l'estimateur␣de␣theta␣de␣la␣loi␣de␣
```

```
            Weibull␣general")
abline(h = theta, col="red", lwd = 2)

#Afichage du boxplot lambda
boxplot(t(lambda_general[c(1,2,3,4,5),]),
        names=c("n=50","n=100","n=200","n=500","n=1000"), col = "
            orange", cex.axis = 0.85, cex.main = 0.75, main = "
            Comportement␣de␣l'estimateur␣de␣lambda␣de␣la␣loi␣de␣
            Weibull␣general")
abline(h = lambda, col="red", lwd = 2)
```