TP Statistiques :
Compte-Rendu

Louis Chevant, David Sit, Xavier Milhaud

2 mars 2007

Table des matières

1 Propriétés d'estimateurs des paramètres de la loi gamma

On note $(\tilde{a}_n, \tilde{\lambda}_n)$ les estimateurs de (a, λ) par la méthode des moments (EMM) et $(\hat{a}_n, \hat{\lambda}_n)$ les estimateurs de maximum de vraisemblance (EMV).

1.1 Simulation d'un échantillon de taille 50 de loi gamma

La commande R pour effectuer un échantillon de taille 50 d'une loi gamma de parametre $a = 2$ et $\lambda = 4$ est : `echantillon <- rgamma(50,2,4)`. On obtient le resultat suivant :

```
 [1] 0.46331845 0.57329151 0.79118670 0.01091005 0.54772460 0.28467978 0.06399521
 [8] 1.29489197 0.62967189 0.80381732 0.72881839 0.63280851 0.61844639 0.33769750
[15] 0.13528189 0.95986827 0.28223123 0.40488010 0.46051527 0.65005093 0.35086963
[22] 0.30967330 0.25066484 0.67571296 0.83540287 0.56898892 1.00300667 0.31445297
[29] 0.11994899 0.09786629 0.31631179 0.26968969 0.89113564 0.61391493 0.38938988
[36] 0.06283447 0.75799614 0.26765639 0.39760205 0.25453143 1.04853626 0.22698139
[43] 0.13135179 0.37180946 0.41860935 1.61205043 0.16000014 0.69325456 0.21534892
[50] 0.45176097
```

Puis on calcule la moyenne , la variance et la médiane de cet chantillon grce respectivement aux fonctions R suivantes. Les résultats sont stockés dans des variables :
`moyenne<-mean(echantillon)`
`mediane<-median(echantillon)`
`variance<-var(echantillon)`.

Les valeurs trouvées sont :

```
> moyenne
[1] 0.4950288
> mediane
[1] 0.4117447
> variance
[1] 0.1072338
```

.

On peut avoir un résumé lgrement plus détaillé grce : `summary(echantillon)` qui donne :

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
0.01091 0.26820 0.41170 0.49500 0.66930 1.61200
```

Afin de tracer l'histogramme de cet chantillon, on utilise la rgle de Sturges. Pour $n = 50$, on obtient 7 classes. A l'aide du code qui se trouve en annexe, on obtient l'histogramme suivant :

```
> histo(echantillon)
```

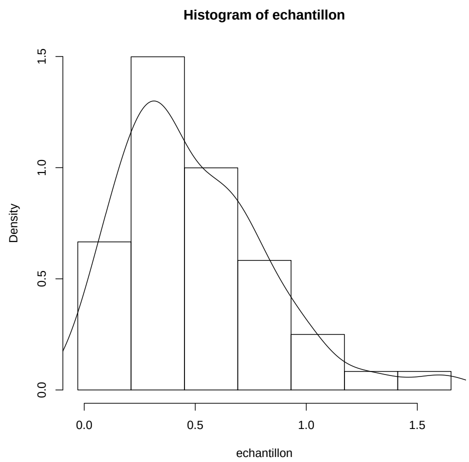

FIG. 1 – Histogramme classes de mme largeur de l'échantillon

On calcule l'EMV numériquement puisqu'il n'existe pas de solution explicite quand on essaie d'annuler les dérivées partielles de la log-vraisemblance. On va donc maximiser la log-vraisemblance grce la fonction optim. Il faut cependant faire attention au fait que `optim` minimise la solution, on entrera donc l'opposé de la log-vraisemblance pour maximiser les valeurs et obtenir les estimateurs. Pour la loi gamma , cette quantité vaut :

$$n * a * ln(\lambda) - n * ln\Gamma(a) - \lambda \sum_{i=1}^{n} x_i + (a-1) \sum_{i=1}^{n} ln(x_i)$$

On obtient le résultat suivant :

```
> emv(echantillon)
[1] 1.925042 3.888621
```

Les EMV de cet échantillon sont donc : $\hat{a}_n = 1.925042$ et $\hat{\lambda}_n = 3.888621$. En ce qui concerne l'EMM, ils sont donnés pour une loi gamma par les formules suivantes :

$$\tilde{a}_n = \frac{X_n}{S_n^2} \text{ et } \tilde{\lambda}_n = \frac{X_n^2}{S_n^2}$$

et dont les valeurs numriques pour notre chantillon sont :

```
> emm(echantillon)
[1] 1.863162 3.198879
```

1.2 Simulation d'un nombre m d'échantillons de taille n

Pour cette question, nous avons implment une fonction `echantill` qui prend comme argument les paramètres n, m , a et λ. On stocke les EMV dans une matrice de taille (m,2) dont les colonnes représentent les $\hat{a}_n$ et $\hat{\lambda}_n$. Puis on calcule le biais pour 'a' en effectuant `mean(c1) - a`. De mme pour λ.
Ex pour $a = 2$, $\lambda = 4$, $n = 50$ et $m = 10$, on obtient :

```
> echantill(50,10,2,4)
Le biais de l'EMV de a vaut :  0.06248451
Le biais de l'EMV de lambda vaut :  0.2508922
```

Biais de $\hat{a}_n$	$n = 10$	$n = 50$	$n = 200$	$n = 500$	$n = 1000$
$m = 10$	0.428	0.128	-0.023	0.007	**0.008**
$m = 50$	0.469				
$m = 200$	0.5441				
$m = 500$	0.763				
$m = 1000$	**0.744**				**0.001**

Synthèse : ces tests nous permettent de montrer donc que les paramètres n et m n'ont pas la même influence : ainsi n a une action sur la convergence asymptotique du biais vers 0 tandis que m influe sur le fait d'avoir des estimateurs de a et λ proches entre différentes simulations d'échantillons.

1.3 Vérification empirique des propriétés de l'EMV

Le but de cette question est de vérifier empiriquement que l'EMV est un estimateur asymptotiquement sans biais du couple ($\hat{a}_n$,$\hat{\lambda}_n$).
Pour cela, il suffit de faire tendre la taille de l'échantillon vers l'infini et de constater que le biais tend vers 0. Un exemple de simulation d'une telle expérience est de prendre $n = 1000$. Nous obtenons alors un biais trs faibles comme le montre le tableau ci-dessus :

Pour vérifier que l'EMV est asymptotiquement gaussien, il suffit de vérifier qu'il suit une loi normale. C'est pourquoi, pour chaque paramètre $\hat{\lambda}_n$ et $\hat{a}_n$, on va tracer le graphe de probabilité. Si les points du nuage sont approximativement alignés, on admettra que l'EMV est asymptotiquement gaussien. En prenant $n = 50$, $m = 300$, $a = 2$ et $\lambda = 4$, on obtient les graphes suivants :

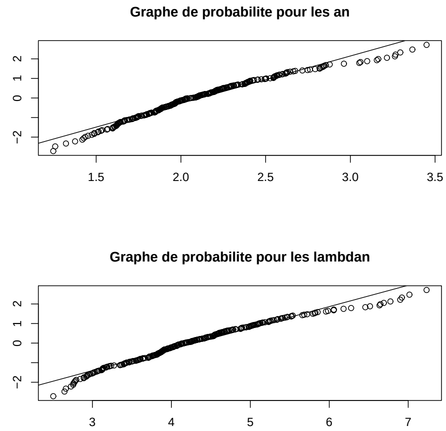

FIG. 2 – Graphe de probabilté de $\hat{\lambda}_n$ et $\hat{a}_n$

On s'aperoit que les points du nuage sont quasiment alignés, ce qui signifie que l'EMV est effectivement asymptotiquement gaussien.

1.4 Calcul de l'erreur quadratique moyenne de l'EMV

Comme on a déj calculé le biais auparavant , et qu'on sait que l'erreur quadratique moyenne d'un estimateur T_n s'écrit :

$$\mathbf{EQM}(T_n) = \textbf{Variance de l'estimateur} + \textbf{carré de son biais}$$

il suffit de reprendre le code fait pour calculer le biais de l'EMV de m échantillons de taille n. La variance est quant elle donnée par la commande `(n-1)/n*var(c)`. On obtient le résultat suivant grce la fonction erreur1 (cf. Annexe pour le code) :

```
> erreur1(50,300,2,4)
L'EQM de l'EMV de a vaut :  0.2778088
L'EQM de l'EMV de lambda vaut :  1.157973
```

EQM de $\hat{a}_n$	$n = 10$	$n = 50$	$n = 200$	$n = 500$	$n = 1000$
$m = 10$	0.261	0.102	0.059	0.075	**0.012**
$m = 50$	2.516				
$m = 200$	3.888				
$m = 500$	3.062				
$m = 1000$	**3.107**				**0.011**

Synthèse : Ceci nous montre clairement que l'EQM de $\hat{a}_n$ est une fonction dcroissante de n mais pas de m. De plus, ces observations nous semblent suffisantes pour envisager une convergence vers 0 de l'EQM pour n grand.

1.5 Etude des mêmes critères pour l'EMM

Pour l'EMM, la méthode est la mme que celle utilisée pour l'EMV, on remplace juste l'appel de la fonction du calcul de l'EMV par celle du calcul de l'EMM.

Le biais trouvé par la fonction `echantill2` est :

```
> echantill2(50,10,2,4)
Le biais de l'EMM de a vaut :  0.2051064
Le biais de l'EMM de lambda vaut :  0.3613102
```

Biais de $\tilde{a}_n$	$n = 10$	$n = 50$	$n = 200$	$n = 500$	$n = 1000$
$m = 10$	0.890	-0.110	0.0309	-0.006	**0.008**
$m = 50$	0.997				
$m = 200$	0.997				
$m = 500$	0.960				
$m = 1000$	**0.988**				**0.011**

Ce tableau montre que l'EMM est un estimateur asymptotiquement sans biais de 'a'. En effet le biais tend vers 0 pour n tendant vers l'infini. De mme pour λ.

En ce qui concerne le comportement asymptotique gaussien, on trace le graphe de probabilités à l'aide de la fonction `nuage2` et on regarde si les points sont suffisamment alignés pour affirmer que les estimateurs EMM suivent une loi normale.

Comme le montre la figure, les points étant alignés, on peut conclure que l'EMM est asymptotiquement gaussien.

Pour l'erreur quadratique, on obtient :

```
> erreur2(50,300,2,4)
L'EQM de l'EMM de a vaut :  0.4892925
L'EQM de l'EMM de lambda vaut :  1.736835
```

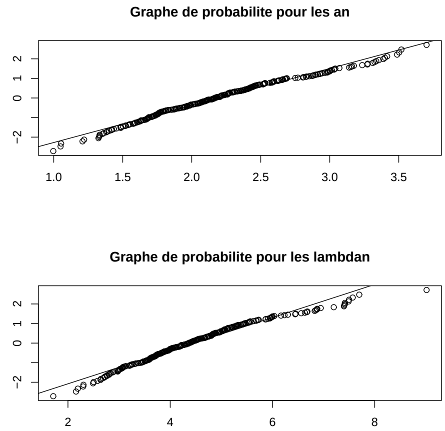

FIG. 3 – Graphe de probabilté de $\tilde{\lambda_n}$ et $\tilde{a_n}$

1.6 Comparaison des résultats des deux estimateurs

Prenons 300 échantillons de taille 50. On obtient les rsultats suivants :

```
> echantill(50,300,2,4)
Le biais de l'EMV de a vaut :  0.09409044
Le biais de l'EMV de lambda vaut :  0.2241013
> echantill2(50,300,2,4)
Le biais de l'EMM de a vaut :  0.1159625
Le biais de l'EMM de lambda vaut :  0.2662619
```

On se rend compte que le biais de l'EMM est plus grand que celui obtenu par l'EMV. Comme le meilleur estimateur est celui de plus petit biais, on peut conclure que c'est l'EMV $(\hat{a}_n, \hat{\lambda}_n)$ qui est le meilleur estimateur des deux. De plus le fait que l'EMV soit asymptotiquement sans biais et efficace fait que, si on a beaucoup de donnes, on est pratiquement certains que la mthode de vraisemblance est la meilleure mthode d'estimation possible.

Essayons avec des échantillons de taille beaucoup plus grande ($n = 500$) pour voir si l'on obtient toujours le mme résultat. Résultats pour $n = 500$:

```
> echantill(500,300,2,4)
Le biais de l'EMV de a vaut :  0.01141606
Le biais de l'EMV de lambda vaut :  0.02574626
> echantill2(500,300,2,4)
Le biais de l'EMM de a vaut :  0.01885884
Le biais de l'EMM de lambda vaut :  0.02737291
> erreur1(500,300,2,4)
```

```
L'EQM de l'EMV de a vaut :  0.02698536
L'EQM de l'EMV de lambda vaut :  0.1035687
> erreur2(500,300,2,4)
L'EQM de l'EMM de a vaut :  0.02374427
L'EQM de l'EMM de lambda vaut :  0.1155705
```

L'EMV reste toujours le meilleur si on ne prend en compte que le biais, qui est infrieur pour l'EMV par rapport l'EMM. Cependant on voit aussi que l'EQM de l'EMM est sensiblement gale celui de l'EMV. C'est pour cela qu'il faut relativiser les rsultats. Par exemple, un estimateur biais peut tre intressant si son erreur quadratique moyenne est infrieure la variance d'un estimateur sans biais.

2 Intervalles de confiance pour le paramètre λ

On suppose que le paramètre a est connu, fixé. Supposons ici $a = 2$

2.1 Calcul théorique de l'intervalle de confiance

Nous cherchons dans cette partie à donner un intervalle de confiance de seuil α pour λ. Le but est de trouver une fonction pivotale dépendante des observations (donc de l'échantillon) et de λ, qui suit une loi connue et dont le ou les paramètres ne dépendent pas de λ.

Un résultat connu stipule que la somme de deux lois Gamma reste une loi Gamma, de plus : Si X et Y sont des variables aléatoires indépendantes de lois respectives $G(\alpha, \lambda)$ et $G(\beta, \lambda)$ alors X+Y suit une loi $G(\alpha + \beta, \lambda)$.

Dans notre cas, la $\sum_{i=1}^{n} X_i$ suit donc une loi $G(na, \lambda)$

Une autre propriété fait la relation entre une loi Gamma particulière et la loi du chi-2 :

La loi $G\left(\frac{n}{2}, \frac{1}{2}\right)$ est appelée loi du chi-2 à n degés de liberté, notée χ^2_n.

Le but est donc de se ramener à une loi Gamma qui aurait pour paramètre $G\left(\mathbb{N}, \frac{1}{2}\right)$. C'est pourquoi nous avons également besoin du résultat suivant :

Si X est une variable aléatoire de loi $G(\alpha, \lambda)$ et a un réel strictement positif, alors aX suit une loi $G\left(\alpha, \frac{\lambda}{a}\right)$.

.

Ainsi, la fonction pivotale intéressante serait $Y = 2\lambda \sum_{i=1}^{n} X_i$ qui suivrait donc une loi $G\left(na, \frac{1}{2}\right)$ qui est en fait une loi du chi-2 de paramètre χ^2_{2an}.

En revanche, il faut bien sûr que $2an$ soit un entier puisqu'il s'agit de degrés de liberté !

L'intervalle de confiance cherché est donc du type $[Z_1; Z_2]$, ainsi nous cherchons $P(Z_1 \leq \lambda \leq Z_2) = 1 - \alpha$. Or on a : quels que soient les réels c et d, $0 < c < d$: $P(c \leq Y \leq d) = F_{\chi^2_{4n}}(d) - F_{\chi^2_{4n}}(c)$.

Comme il y a une infinité de faons possibles pour choisis a et b de sorte que cette probabilité soit égale à $1 - \alpha$, on décidé d'équilibrer en prenant : $F_{\chi^2_{4n}}(d) = 1 - \frac{\alpha}{2}$ et $F_{\chi^2_{4n}}(c) = \frac{\alpha}{2}$
Par la commande `qchisq` on peut avoir les valeurs de $z_{4n,\alpha}$. Donc les valeurs de c et d seront donnés par :

$$d = z_{2*a*n,\frac{\alpha}{2}} = \texttt{qchisq}(1 - \tfrac{\alpha}{2}, 2 * a * n)$$
$$c = z_{2*a*n,1-\frac{\alpha}{2}} = \texttt{qchisq}(\tfrac{\alpha}{2}, 2 * a * n)$$

Enfin comme

$$P(c \leq Y \leq d) = P(\frac{c}{2\sum_1^n X_i} \leq \lambda \leq \frac{d}{2\sum_1^n X_i}) = 1 - \alpha$$

il suffit de calculer les valeurs de $z_1 = \frac{c}{2*\sum_1^n X_i}$ et $z_2 = \frac{d}{2*\sum_1^n X_i}$ pour avoir les bornes de l'intervalle de confiance que l'on cherche :

$$IC = [\frac{z_{2an,1-\frac{\alpha}{2}}}{2\sum_i X_i}, \frac{z_{2an,\frac{\alpha}{2}}}{2\sum_i X_i}]$$

2.2 Vérification empirique du résultat

Pour cette question, on va calculer les intervalles de confiance grce à une fonction `genereintervalle` qui prendra respectivement en paramètre la taille des échantillons, un nombre m d'échantillons, a, la vraie valeur de λ et une valeur pour α,. Elle nous retournera le pourcentage d'intervalles contenant la vritable valeur de λ.

Ex pour $m = 30$, $n = 50$, $a = 3$, $\lambda = 2$ et $\alpha = 0.05$:

```
> genere_intervalles(50,30,3,2,0.05)
Le nombre d'intervalles comprenant le vritable lambda est : 28
Il y a 93.33333 % d'intervalles comprenant lambda
```

Ex pour $m = 200$, $n = 300$, $a = 3$, $\lambda = 2$:

```
> genere_intervalles(300,200,3,2,0.05)
Le nombre d'intervalles comprenant le vritable lambda est : 195
Il y a 97.5 % d'intervalles comprenant lambda
```

On observe bien que lorqu'on simule un grand nombre d'chantillons $(X_1, .., X_n)$ de loi $G(3, 2)$, on a une proportion gale $1 - \alpha$ des intervalles de confiance de seuil α obtenus contient la vraie valeur du paramtre.

3 Annexe : Code des fonction

3.1 Code pour la question 1.1

```
#Genere un echantillon de taille 50 de loi gamma
echantillon<-rgamma(50,2,4)

# Calcul de la moyenne, de la variance et de la mediane empirique de l'echantillon
moyenne<-mean(echantillon)
mediane<-median(echantillon)
variance<-49/50*var(echantillon)

#Affiche un resume de l'echantillon
summary(echantillon)


# Affiche un histogramme de meme classe de l'echantillon
histo<- function(echantillon)
{
n <- length(echantillon)  # nombre de donnees
k <- round(log2(n) + 1)   # nombre de classes(regle Sturges)
x1 <- min(echantillon)
xn <- max(echantillon)
etendue = (xn - x1)
a0 = x1 - 0.025*etendue
ak = xn + 0.025*etendue
pas = (ak - a0) / k
hist(echantillon, prob=T, breaks=seq(a0,ak,pas))
lines(density(echantillon))
}

# Fonction permettant de determiner les EMV numeriquement
emv<-function(x)
{
  logvraisemblance<-function(par,e=x)
  {
# retourne l'oppose de la log-vraisemblance
  a<-par[1]
  lambda<-par[2]
  n<-length(e)
  v<--n*a*log(lambda)+n*log(gamma(a))+lambda*sum(e)-(a-1)*sum(log(e))
  }
emv <- optim(c(1, 1), logvraisemblance)
emv <-emv$par
return(emv)
}


# Fonction permettant de determiner les EMM numeriquement
emm<-function(x) # Calcule l'EMM d'un vecteur x
```

```
{
n=length(x)
moy<-mean(x)
var<-(n-1)/n*var(x)

lambda<-moy/var
a<-moy^2/var

return(c(a,lambda))
}
```

3.2 Code pour la question 1.2

```
#Methode sumulant un nombre m d'enchantillons de taille n et calculant
#le biais de l'EMV

echantill<-function(n,m,a,lambda)
{
mat<-matrix(data=0,nrow=m,ncol=2)
for (i in 1:m){
mat[i,]<-emv(rgamma(n,a,lambda))
}

#Pour chaque echantillon l, l'EMV est stockee \`a la ligne l de la matrice

c1<-numeric(m) #Creation d'un vecteur de taille m
c2<-numeric(m) #Creation d'un vecteur de taille m

for (i in 1:m){
c1[i]=mat[i,1]  #Le vecteur c1 stocke toutes les valeurs des a_n
c2[i]=mat[i,2]  #Le vecteur c2 stocke toutes les valeurs des lambda_n
}

biais1<-mean(c1)-a
biais2<-mean(c2)-lambda

cat("Le biais de l'EMV de a vaut : ", biais1,"\n")
cat("Le biais de l'EMV de lambda vaut : ", biais2,"\n")
```

3.3 Code pour la question 1.3

```
# On va tracer le nuage de points sur les EMV a_n pour montrer que qu'ils sont
# bien alignes et qu'ils suivent une loi normale, c'est a dire asymptotiquement gaussien


nuage<-function(n,m,a,lambda)
```

```
{
mat<-matrix(data=0,nrow=m,ncol=2)
for (i in 1:m){
mat[i,]<-emv(rgamma(n,a,lambda))
}

#Pour chaque echantillon l, l'EMV est stockee a la ligne l de la matrice

c1<-numeric(m) #Creation d'un vecteur de taille m
c2<-numeric(m) #Creation d'un vecteur de taille m

for (i in 1:m){
c1[i]=mat[i,1]  #Le vecteur c1 stocke toutes les valeurs des a_n
c2[i]=mat[i,2]  #Le vecteur c2 stocke toutes les valeurs des lambda_n
}

par(mfrow=c(2,1))
c1sort<-sort(c1)
plot(c1sort[1:(length(c1sort)-1)],qnorm(seq(1:(length(c1sort)-1))/length(c1sort)),xlab="",
                                   ylab="",main="Graphe de probabilite pour les an")

reg<-lm(qnorm(seq(1:(length(c1sort)-1))/length(c1sort))~c1sort[1:(length(c1sort)-1)])
abline(reg)

c2sort<-sort(c2)
plot(c2sort[1:(length(c2sort)-1)],qnorm(seq(1:(length(c2sort)-1))/length(c2sort)),xlab="",
                                  ylab="",main="Graphe de probabilite pour les lambdan")

reg<-lm(qnorm(seq(1:(length(c2sort)-1))/length(c2sort))~c2sort[1:(length(c2sort)-1)])
abline(reg)

}
```

3.4 Code pour la question 1.4

```
# Cette fonction va simuler m echantillons de taille n puis
# elle va calculer les erreurs quadratiques moyennes


erreur1<-function(n,m,a,lambda)
{
mat<-matrix(data=0,nrow=m,ncol=2)
for (i in 1:m){
mat[i,]<-emv(rgamma(n,a,lambda))
}

#Pour chaque echantillon l, l'EMV est stockee a la ligne l de la matrice

c1<-numeric(m) #Creation d'un vecteur de taille m
```

```
c2<-numeric(m) #Creation d'un vecteur de taille m

for (i in 1:m){
c1[i]=mat[i,1] #Le vecteur c1 stocke toutes les valeurs des a_n
c2[i]=mat[i,2] #Le vecteur c2 stocke toutes les valeurs des lambda_n
}

biais1<-mean(c1)-a
biais2<-mean(c2)-lambda

err1<-(n-1)/n*var(c1)+ biais1  # l'EQM est egale a la somme de la
err2<-(n-1)/n*var(c2)+ biais2  # variance et du caree de son biais

cat("L'EQM de l'EMV de a vaut : ", err1,"\n") # Affiche l'EQM des a_n
cat("L'EQM de l'EMV de lambda vaut : ", err2,"\n") # Affiche l'EQM des lambda_n

}
```

3.5 Code pour la question 1.5

```
#Methode simulant un nombre m d'echantillons de taille n et calculant
#le biais de l'EMM

echantill2<-function(n,m,a,lambda)
{
mat<-matrix(data=0,nrow=m,ncol=2)
for (i in 1:m){
mat[i,]<-emm(rgamma(n,a,lambda))
}

#Pour chaque echantillon l, l'EMM est stockee a la ligne l de la matrice

c1<-numeric(m) #Creation d'un vecteur de taille m
c2<-numeric(m) #Creation d'un vecteur de taille m

for (i in 1:m){
c1[i]=mat[i,1]  #Le vecteur c1 stocke toutes les valeurs des a_n
c2[i]=mat[i,2]  #Le vecteur c2 stocke toutes les valeurs des lambda_n

}

biais1<-mean(c1)-a
biais2<-mean(c2)-lambda

cat("Le biais de l'EMM de a vaut : ", biais1,"\n")
cat("Le biais de l'EMM de lambda vaut : ", biais2,"\n")
}


# Methode verifiant si l'EMM est asymptotiquement gaussien
```

```
nuage2<-function(n,m,a,lambda)
{
mat<-matrix(data=0,nrow=m,ncol=2)
for (i in 1:m){
mat[i,]<-emm(rgamma(n,a,lambda))
}

#Pour chaque echantillon l, l'EMM est stockee a la ligne l de la matrice

#print(mat)

c1<-numeric(m) #Creation d'un vecteur de taille m
c2<-numeric(m) #Creation d'un vecteur de taille m

for (i in 1:m){
c1[i]=mat[i,1]  #Le vecteur c1 stocke toutes les valeurs des a_n
c2[i]=mat[i,2]  #Le vecteur c2 stocke toutes les valeurs des lambda_n
}

par(mfrow=c(2,1))
c1sort<-sort(c1)
plot(c1sort[1:(length(c1sort)-1)],qnorm(seq(1:(length(c1sort)-1))/length(c1sort)),xlab="",
                                  ylab="",main="Graphe de probabilite pour les an")

reg<-lm(qnorm(seq(1:(length(c1sort)-1))/length(c1sort))~c1sort[1:(length(c1sort)-1)])
abline(reg)

c2sort<-sort(c2)
plot(c2sort[1:(length(c2sort)-1)],qnorm(seq(1:(length(c2sort)-1))/length(c2sort)),xlab="",
                                  ylab="",main="Graphe de probabilite pour les lambdan")

reg<-lm(qnorm(seq(1:(length(c2sort)-1))/length(c2sort))~c2sort[1:(length(c2sort)-1)])
abline(reg)

}


# Cette fonction va simuler m echantillons de taille n puis
# elle va calculer les erreurs quadratiques moyennes

erreur2<-function(n,m,a,lambda)
{
mat<-matrix(data=0,nrow=m,ncol=2)
for (i in 1:m){
mat[i,]<-emm(rgamma(n,a,lambda))
}

#Pour chaque echantillon l, l'EMV est stockee a la ligne l de la matrice
```

```
#print(mat)

c1<-numeric(m) #Creation d'un vecteur de taille m
c2<-numeric(m) #Creation d'un vecteur de taille m

for (i in 1:m){
c1[i]=mat[i,1] #Le vecteur c1 stocke toutes les valeurs des a_n
c2[i]=mat[i,2] #Le vecteur c2 stocke toutes les valeurs des lambda_n
}

biais1<-mean(c1)-a
biais2<-mean(c2)-lambda

err1<-(n-1)/n*var(c1)+ biais1  # l'EQM est egale a la somme de la
err2<-(n-1)/n*var(c2)+ biais2  # variance et du caree de son biais

cat("L'EQM de l'EMM de a vaut : ", err1,"\n") # Affiche l'EQM des a_n
cat("L'EQM de l'EMM de lambda vaut : ", err2,"\n") # Affiche l'EQM des lambda_n
}
```

3.6 Code pour la question 2.2

```
genere_intervalles<-function(n,m,a,lambda,alpha)
{
nbr<-0
mat<-matrix(data=0,nrow=m,ncol=2)
for (i in 1:m){
x<-rgamma(n,a,lambda)
d<-qchisq(1-alpha/2,2*a*length(x))
c<-qchisq(alpha/2,2*a*length(x))
somme<-sum(x)
u<-c/(2*somme)
v<-d/(2*somme)

mat[i,1]<-u
mat[i,2]<-v
}

for (i in 1:m){
if(((lambda-mat[i,][1])>=0 ) & ((mat[i,][2]-lambda)>=0)){
nbr<-nbr+1 #on incremente le compteur si l'intervalle contient lambda
}
}
cat("Le nombre d'intervalles comprenant le vritable lambda est :", nbr , "\n")
pourc<-nbr*100/m
cat("Il y a", pourc, "% d'intervalles comprenant lambda \n")
}
```