TP3 Statistiques

Matthias LAPU , Amael KREIS

Estimation du Maximum de vraisemblance et L'intervalle de confiance

Vraisemblance: La loi Bernoulli

Soit X une variable aléatoire de Bernoulli (`rbinom`) avec $p = 0.6$.

1. Simuler un échantillon i.i.d de taille $n = 10$. Quelle est une façon simple d'estimer p?

```
echantillon <- rbinom(10,1,0.6)
print(echantillon)
```

```
##  [1] 0 1 1 1 0 0 1 1 1 1
```

Une façon simple d'estimer p est de calculer la moyenne empirique de l'échantillon.Cependant cette méthode n'est pas très précise pour des échantillons de petites tailles.

```
print(mean(echantillon))
```

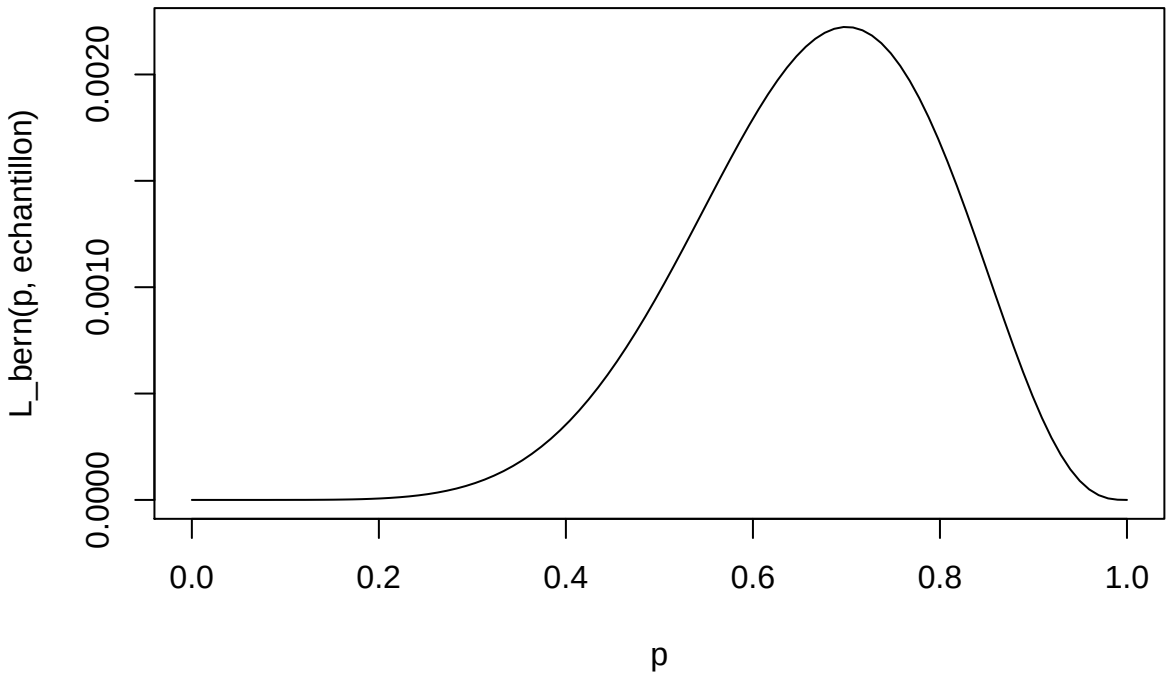
```
## [1] 0.7
```

2. Générer une fonction de vraisemblance, nommée `L_bern`, en fonction de (p, x), qui donne la vraisemblance d'un échantillon $x = (x_1, \ldots, x_n)$ pour une valeur donnée de p.

```
L_bern <- function(p,x) {
  n <- sum(x)
  return(p^n * (1-p)^(length(x) - n))
}
```

3. Pour votre échantillon, estimer la vraisemblance de l'échantillon pour n lois Bernoulli de paramètres p allant de 0 à 1. Tracez la courbe des valeurs calculées. Que remarquez-vous?

```
p <- seq(0,1,length.out = 100)
plot(p,L_bern(p,echantillon),type = "l",main="Vraisemblance d'un échantillon de Bernouili")
```



On remarque que la fonction se trouve régulièrement à x=p.Cela était prévisible que la moyenne empirique soit proche de la moyenne, en effet nous savons que c'est un estimateur convergent.

4. En utilisant la fonction `optim` de R, trouvez la valeur de p la plus probable d'avoir généré cet échantillon.

Attention : `optim` est par défaut une routine de **minimization**. Remarque : Avec la méthode de *L-BFGS-B* dans la fonction `optim`, vous pouvez traiter des contraintes sur le(s) paramètre(s), lorsque c'est nécessaire.

```
fn <- function(p){
  r <-L_bern(p,echantillon)
  return(1 - r)
}
optim(par = 0.5,fn,method = "L-BFGS-B",lower = 0,upper = 1)
```

```
## $par
## [1] 0.6999993
##
## $value
## [1] 0.9977764
##
## $counts
## function gradient
##        9        9
##
## $convergence
## [1] 0
##
## $message
## [1] "CONVERGENCE: REL_REDUCTION_OF_F <= FACTR*EPSMCH"
```

5. Tester avec des échantillons de taille allant de $n = 10$ à $n = 2000$ et comparer l'écart entre la valeur théorique attendue et la valeur obtenue. Que remarquez-vous? Comment combattre l'instabilité numérique due aux multiplications de probabilités?

```
res <- c()
for(i in seq(10,2000,10)){
  echantillon <- rbinom(i,1,0.6)
  res <- c(res,optim(par = 0.2,fn,method = "L-BFGS-B",lower = 0,upper = 1)[[1]])
}
```

On remarque qu'il est préférable d'utiliser la logvraisemblance, le log transforme les multiplications en addition, cela permet donc de réduire l'instabilité.

6. Trouver deux intervalles de confiance de niveau 0.90 pour le paramètre p, d'après (i) l'inégalité de Bienaymé-Chebycheff et

   (ii) l'inégalité de Hoeffding et les comparer. Incluent-ils la valeur réelle ?

(i) : On étudie un échantillon avec de nombreuses valeurs. Nous savons que la moyenne empirique est un estimateur convergent de p. Prenons :

$$\bar{X} = \frac{1}{n}\sum_{i=1}^{n} X_i X_i \sim B(p)$$

Rappelons l'inégalité de Bienaymé-Tchebychev :

$$P(|X - \mu| \geq \delta) \leq \frac{V(X)}{\delta\check{s}}$$

Appliquons cette inégalité à la moyenne empirique :

$$P(|\bar{X} - p| \geq \delta) \leq \frac{p(1-p)}{\delta\check{s}n} P(|\bar{X} - p| < \delta) \geq 1 - \frac{p(1-p)}{\delta\check{s}n} P(\bar{X} - \delta < p < \bar{X} + \delta) \geq 1 - \frac{p(1-p)}{\delta\check{s}n}$$

On trouve donc notre intervalle de confiance, cependant p est encore dans la partie droite. Il est donc nécessaire de l'enlever pour continuer notre estimation. passons $p(1-p)$ sous forme canonique :

$$p(1-p) = p - p\check{s} = -(p - \frac{1}{2}) + \frac{1}{4}p(1-p) \leq \frac{1}{4}$$

Modifions l'inégalité :

$$P(\bar{X} - \delta < p < \bar{X} + \delta) \geq 1 - \frac{1}{\delta\check{s}4n}$$

Finalement nous avons donc :

$$\alpha = \frac{1}{\delta\check{s}4n} \rightarrow \delta = \frac{1}{2\sqrt{\alpha * n}}$$

Ainsi :

$$P(\bar{X} - \frac{1}{2\sqrt{\alpha * n}} < p < \bar{X} + \frac{1}{2\sqrt{\alpha * n}}) \geq 1 - \alpha$$

Finalement, nous avons donc :

$$I_{n,\alpha} = [\bar{X} \pm \frac{1}{2\sqrt{\alpha * n}}]$$

(ii)Inégalité de Hoeffding:

Rappelons l'inégalité :

$$P(|\bar{X} - p]| \geq \delta) \leq 2exp(-2n\delta^2)$$

Avec : $\alpha = 2exp(-2n\delta^2)$, on obtient l'intervalle de confiance suivante :

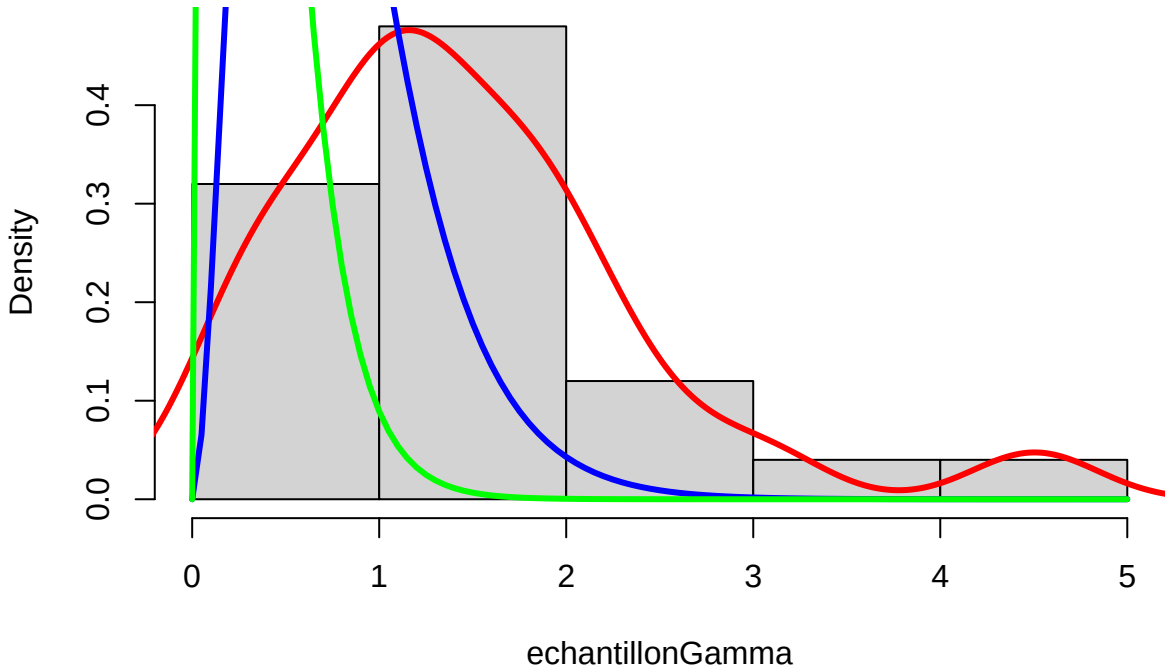
$$I_{n,\alpha} = [\bar{X} \pm \sqrt{\frac{1}{2n}\log(\frac{2}{\alpha})}]$$

Vraisemblance pour plusieurs paramètres: La Loi Gamma

Soit $X_1, \ldots, X_n$ un échantillon de n variables indépendantes de loi de Gamma(α, β) où $\theta = (\alpha, \beta)$ est inconnue. Simuler un échantillon i.i.d de taille $n = 25$ avec $\theta_0 = (2.5, 1.5)$.

7. Présentez l'histogramme des données simulées. Choisir trois paramètres candidats, disons, θ_0 (vrai) θ_1, θ_2. Comparer l'histogramme avec les densités candidates. Que remarquez-vous?

```
n = 25
theta0 <- c(2.5,1.5)
theta1 <- c(3,4)
theta2 <- c(2,6)
echantillonGamma <- rgamma(n,theta0[1],theta0[2])
hist(echantillonGamma,probability = TRUE) # le bon paramètre à estimer
lines(density(echantillonGamma),col="red",lwd=3)
curve(dgamma(x,shape = theta1[1], rate = theta1[2]),add=T,col="blue",lwd=3)
curve(dgamma(x,shape = theta2[1], rate = theta2[2]),add=T,col="green",lwd=3)
```



On remarque que l'échantillon est de trop petite taille pour avoir des informations étudiable.

8. Ecrire la log vraisemblance `logL_gamma`. Générez une fonction de log-vraisemblance avec les arguments (θ, x), qui donne la log vraisemblance d'un échantillon pour une valeur donnée de $\theta = (\alpha, \beta)$ et les donnéé $x = (x_1, \ldots, x_n)$. Pour votre échantillon, estimer la log-vraisemblance de paramètre $\theta = (\alpha, \beta)$, en faissant varier un paramètre à la fois.

La formule de la vraisemblance de la loi gamma est :

$$L(\alpha, \beta|x) = \frac{\beta^\alpha}{\Gamma(\alpha)} x^{\alpha-1} \exp(-\beta x)$$

La formule de la log-vraisemblance est donc :

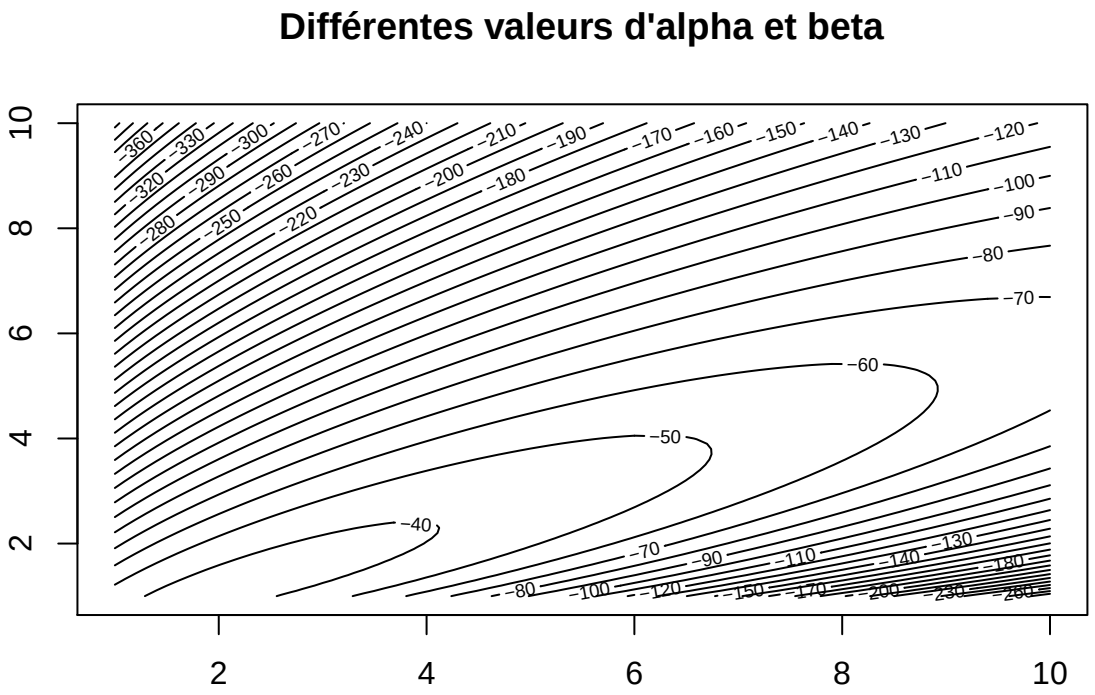
$$\frac{\partial L(\alpha, \beta|x)}{\partial x} = \log \mathcal{L}(\alpha, \beta|x) = \alpha \log \beta - \log \Gamma(\alpha) + (\alpha - 1) \log x - \beta x$$

```
logL_gamma_unit <- function(alpha,beta,x){
  return( alpha*log(beta) - log(gamma(alpha)) + (alpha-1)*log(x) - beta*x )
}

#on fait varier un paramètre et on calcule la somme
logL_gamma <- function(alpha,beta,v){
  sum <- 0
  for(i in v) {
    sum =  sum + logL_gamma_unit(alpha,beta,i)
  }
  return (sum)
}

sample <- rgamma(n, 2.5, 1.5)
```

```r
logLGamma <- function (alpha,beta){
  return (logL_gamma(alpha,beta,sample))
}

# il faut re-créer des fonctions
logLGammaV2 <- function (theta){
  return (logL_gamma(theta[1],theta[2],sample))
}
```

9. Tracer la courbe des valeurs ainsi calculées. Comme la fonction, $\ell(\theta)$, a deux arguments, vous pouvez réaliser des tracés de contour (`contour`). Pour plus de simplicité, il suffit de tracer comme une fonction unidimensionnelle en supposant que l'autre est fixe: $\ell(\alpha|\beta = \beta_0)$ avec quelque β_0 de votre choix et $\ell(\beta|\alpha = \alpha_0)$ pour quelque α_0 de votre choix. Que remarquez-vous?

```r
#on trace les courbes
x <- seq(1,10,0.1)
y <- seq(1,10,0.1)
contour(x,y,outer(x,y,logLGamma),nlevels = 50,main="Différentes valeurs d'alpha et beta")
```



10. Donner l'expression mathématique du vecteur Score (les dérivées premières) à laquelle l'EMV répond. En utilisant la fonction `optim`, trouver la valeur de θ la plus probable pour votre l'échantillon.

Le score est le gradient de la log-vraisemblance. Soit θ un vecteur des paramètres du modèle

$$S(\theta) = \nabla_\theta \mathcal{L}(\theta) = \begin{pmatrix} n(\log(\beta) - \frac{\psi_0(\alpha)}{\alpha}) + \sum_{i=1}^n \log(x_i) \\ \frac{n\alpha}{\beta} - \sum_{i=1}^n x_i \end{pmatrix}$$

Avec ψ la fonction dérivé de la fonction gamma.

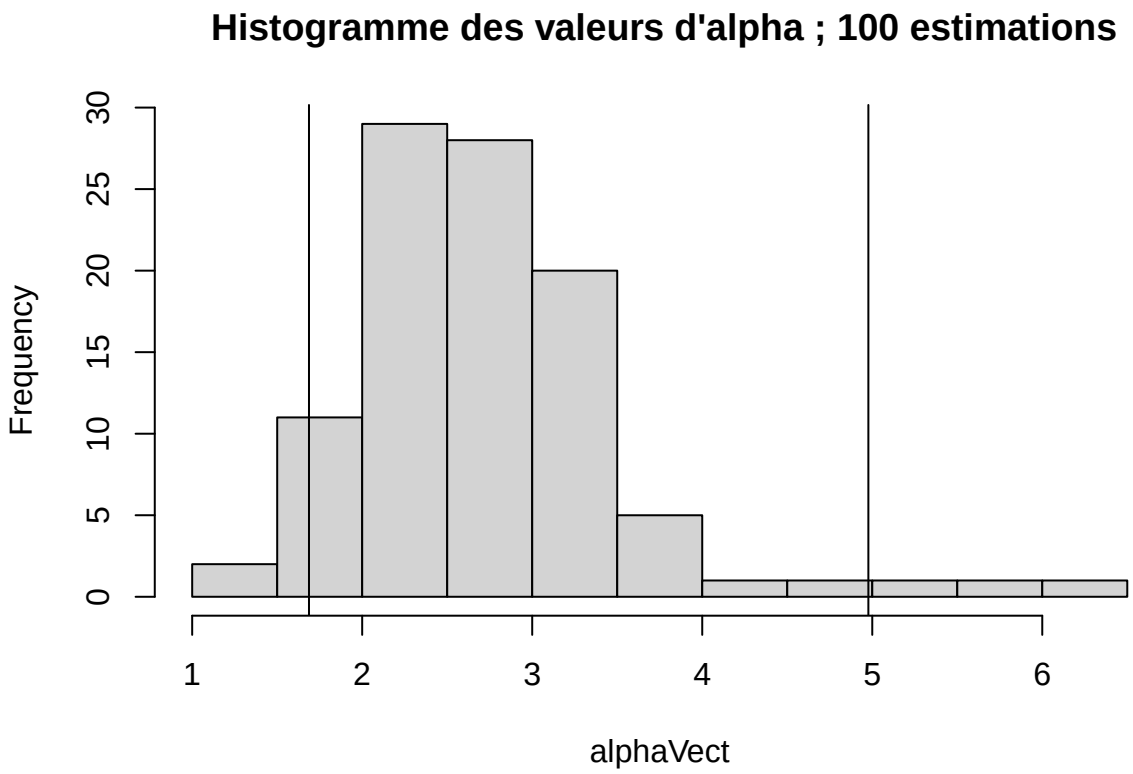
```r
x0 <- c(1,1)
# la fonction optim minimise il faut donc multiplier par -1 pour la maximiser
optim(par = x0, fn = logLGammaV2, method = "L-BFGS-B", control = list(fnscale = -1))
```

```
## $par
## [1] 2.156486 1.195333
##
## $value
## [1] -36.28977
##
## $counts
## function gradient
##       10       10
##
## $convergence
## [1] 0
##
## $message
## [1] "CONVERGENCE: REL_REDUCTION_OF_F <= FACTR*EPSMCH"
```

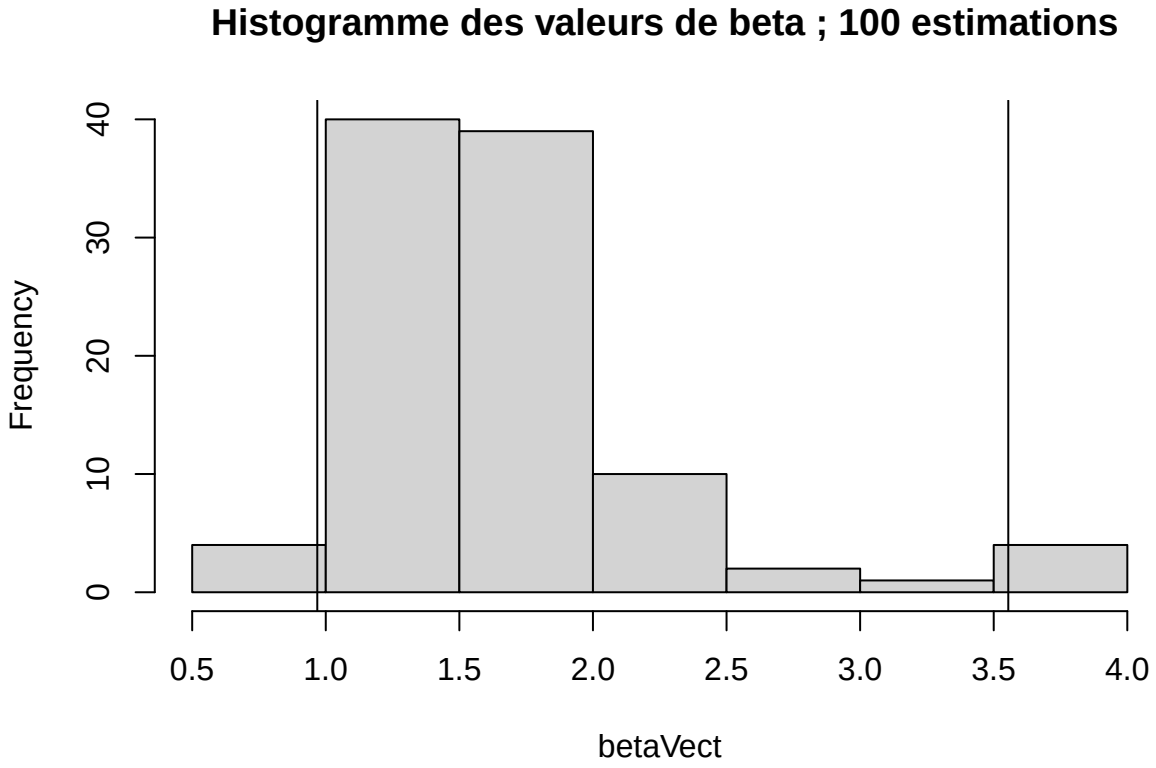
11. Répéter l'estimation 100 fois avec de nouveaux ensembles de données (échantillons) et tracer les estimations $\hat{\alpha}$ vs $\hat{\beta}$. Sont-elles indépendantes ? Trouver l'intervalle où 95% des estimations sont incluses pour chaque paramètre. Vous venez de trouver un intervalle de confiance empirique à 95% ! Visualisez vos résultats à l'aide d'un histogramme. Que remarquez-vous?

```r
# on fait 100 estimations
n=100
logLGammaV2 <- function (theta,echantillon){
  return (logL_gamma(theta[1],theta[2],echantillon))
}


x0 <- c(1,1)
list <- vector("list",100)
#remplit 100 échantillons de alpha et beta
for (i in 1:100){
    list[[i]] <- optim(par = x0, fn = logLGammaV2,echantillon=rgamma(25,2.5,1.5), method = "L-BFGS-B", c
}

alphaVect <- c()
betaVect <- c()
for(i in 1:100){

  alphaVect <- c(alphaVect,list[[i]][1] )
  betaVect <-  c(betaVect,list[[i]][2] )
}

#intervalle de confiance pour alpha et beta
confAlpha = quantile(alphaVect,c(0.025,0.975))
confBeta = quantile(betaVect,c(0.025,0.975))

#histogramme pour le vecteur alpha et le vecteur beta
```

```
hist(alphaVect,main=" Histogramme des valeurs d'alpha ; 100 estimations")
abline(v= confAlpha[1])
abline(v= confAlpha[2])
```




```
hist(betaVect,main=" Histogramme des valeurs de beta ; 100 estimations")
abline(v= confBeta[1])
abline(v= confBeta[2])
```



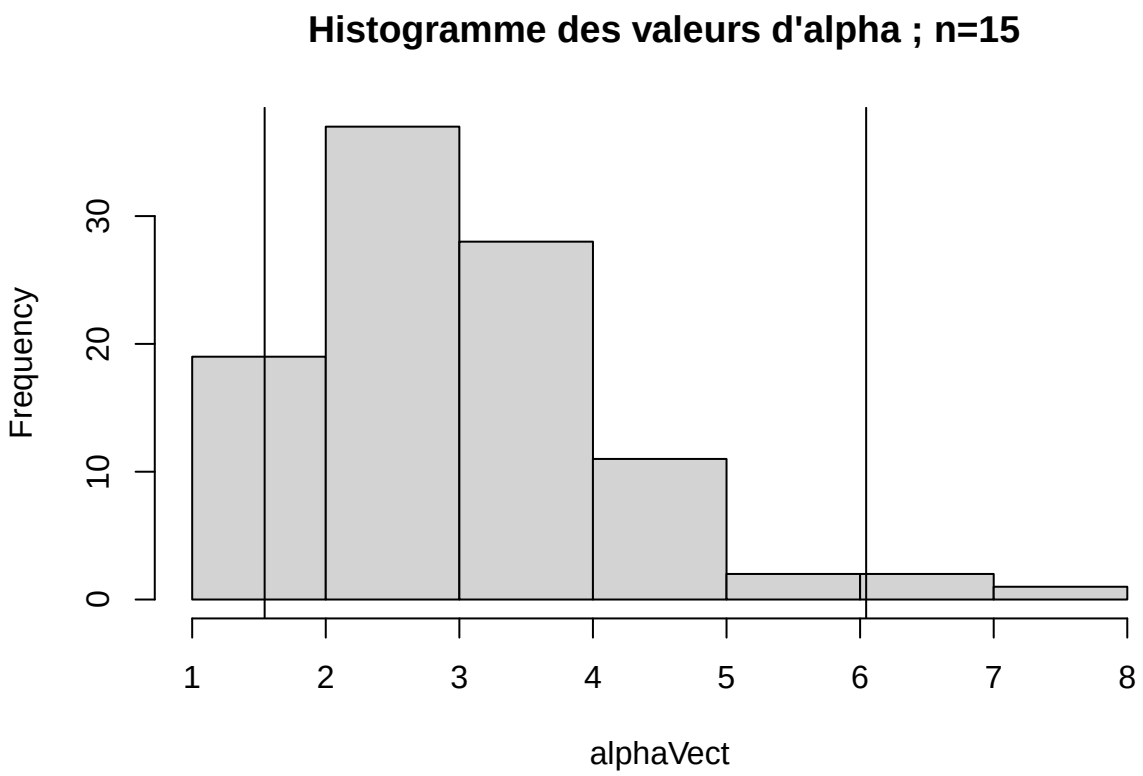
On remarque que les valeurs d'alpha et beta sont plus importante vers 2.5 et 1.5 .Ce sont les résultats que l'on souhaite originellement.

12. Tester avec des échantillons de taille $n = 15$ et $n = 100$ et comparer avec les résultats précédents. Quel est l'effet de la taille de l'échantillon ?

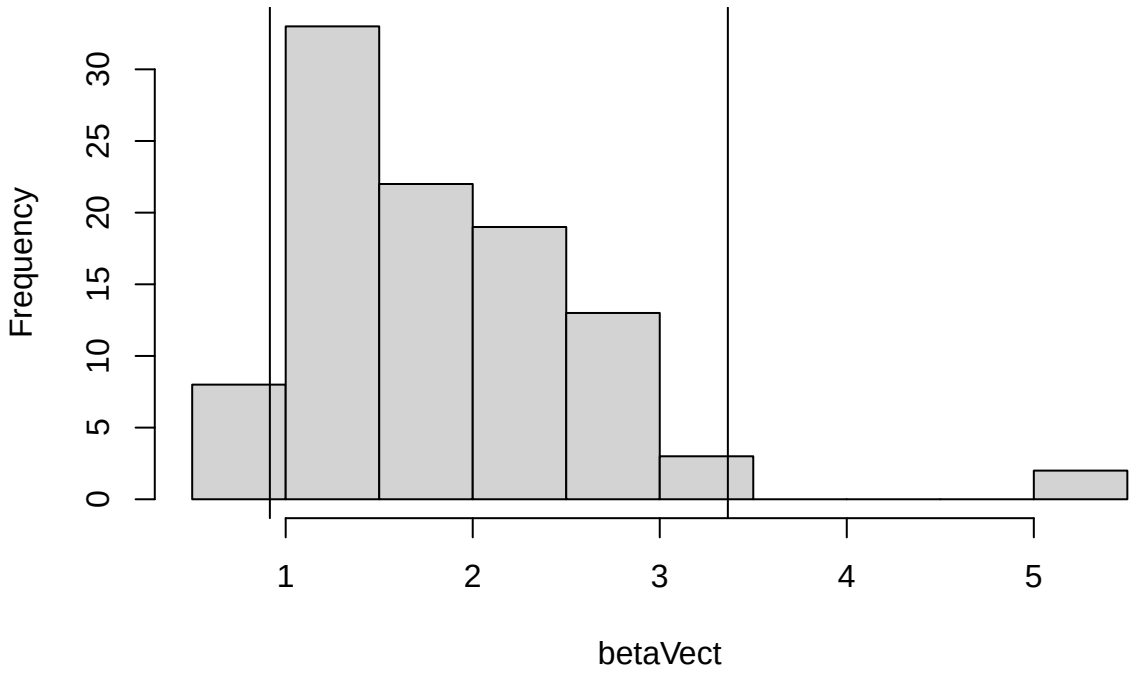
Avec un échantillon n=15 :

```
# on fait 100 estimations
n=100
logLGammaV2 <- function (theta,echantillon){
  return (logL_gamma(theta[1],theta[2],echantillon))
}


x0 <- c(1,1)
list <- vector("list",n)
for (i in 1:n){
    list[[i]] <- optim(par = x0, fn = logLGammaV2,echantillon=rgamma(15,2.5,1.5), method = "L-BFGS-B", c
}

alphaVect <- c()
betaVect <- c()
for(i in 1:n){

  alphaVect <- c(alphaVect,list[[i]][1] )
  betaVect <-  c(betaVect,list[[i]][2] )
}
```

```r
#intervalle de confiance pour alpha et beta
confAlpha = quantile(alphaVect,c(0.025,0.975))
confBeta = quantile(betaVect,c(0.025,0.975))

#histogramme pour le vecteur alpha et le vecteur beta

hist(alphaVect,main=" Histogramme des valeurs d'alpha ; n=15")
abline(v= confAlpha[1])
abline(v= confAlpha[2])
```



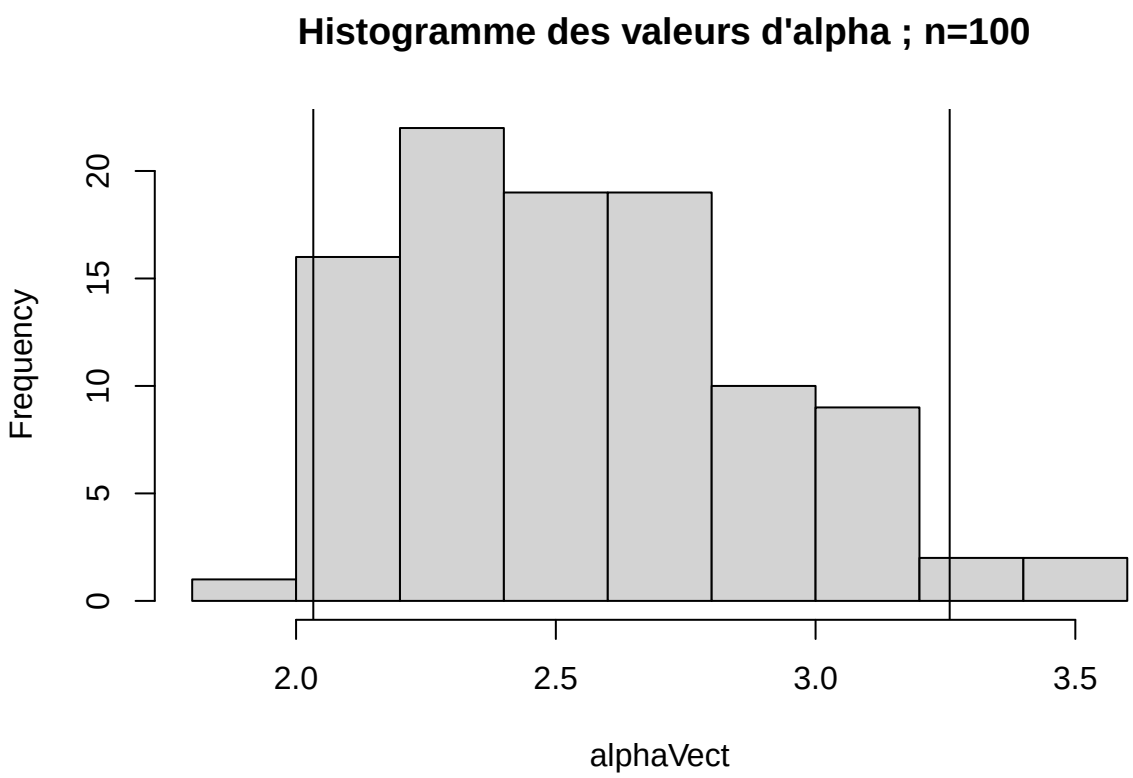

```r
hist(betaVect,main=" Histogramme des valeurs de beta ; n=15")
abline(v= confBeta[1])
abline(v= confBeta[2])
```



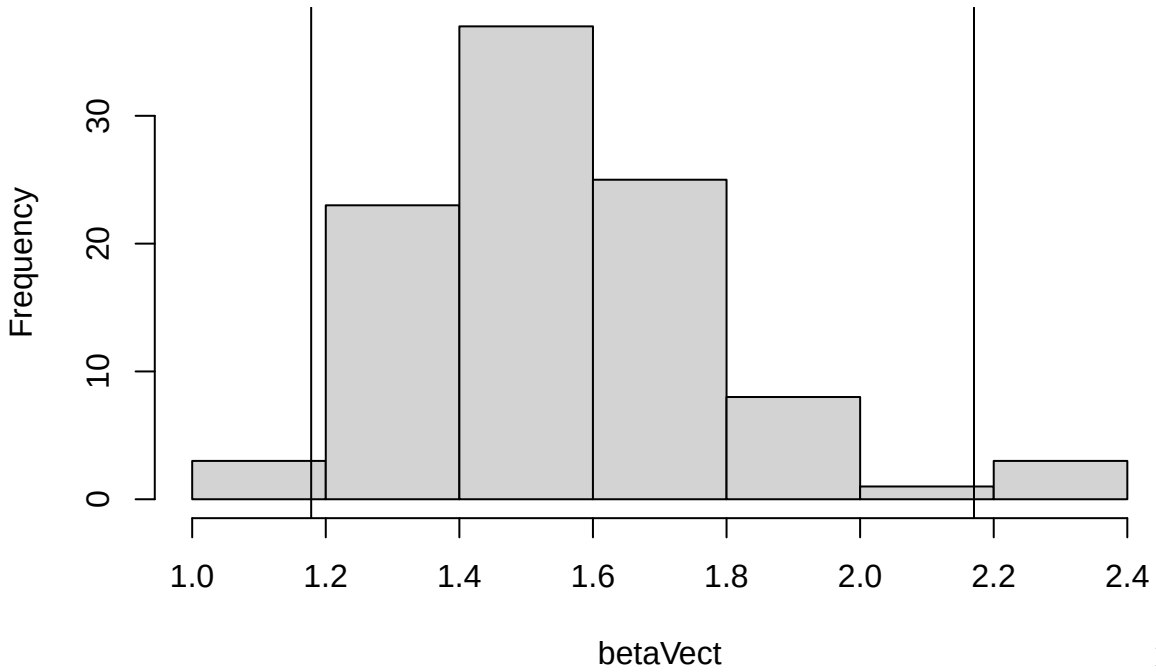
Avec un échantillon n=100

```
# on fait 100 estimations
n=100
logLGammaV2 <- function (theta,echantillon){
  return (logL_gamma(theta[1],theta[2],echantillon))
}


x0 <- c(1,1)
list <- vector("list",n)
for (i in 1:n){
    list[[i]] <- optim(par = x0, fn = logLGammaV2,echantillon=rgamma(100,2.5,1.5), method = "L-BFGS-B",
}

alphaVect <- c()
betaVect <- c()
for(i in 1:n){

  alphaVect <- c(alphaVect,list[[i]][1] )
  betaVect <-  c(betaVect,list[[i]][2] )
}

#intervalle de confiance pour alpha et beta
confAlpha = quantile(alphaVect,c(0.025,0.975))
confBeta = quantile(betaVect,c(0.025,0.975))

#histogramme pour le vecteur alpha et le vecteur beta

hist(alphaVect,main=" Histogramme des valeurs d'alpha ; n=100")
```

```
abline(v= confAlpha[1])
abline(v= confAlpha[2])
```




```
hist(betaVect,main=" Histogramme des valeurs de beta ; n=100")
abline(v= confBeta[1])
abline(v= confBeta[2])
```



L'augmentation de la taille de l'échantillon permet de se rapprocher de la bonne valeur θ, en effet on peut apercevoir l'intervalle de confiance réduire.

Normalité asymptotique de l'EMV et l'intervalle de confidence

En pratique, nous n'avons qu'un seul ensemble de données avec des paramètres $\theta = (\theta_1, \ldots, \theta_p)$ inconnus, la simulation de multiples échantillons permettant de construire la distribution de l'estimateur n'est donc plus possible. Dans ce cas, on peut construire des intervalles basés sur une approximation asymptotique de la distribution, en utilisant le fait que $\hat{\theta} \approx \mathcal{N}(\theta, I_n(\theta)^{-1})$.

L'évaluation de l'information de Fisher directe nécessite de calculer le hessien H et son espérance analytiquement en différenciant l'opposé de la log-vraisemblance deux fois par rapport aux paramètres (pour obtenir la matrice complète), puis d'inverser explicitement la matrice, et enfin de remplacer θ par $\hat{\theta}$.

13. Pour revenir à l'exemple du Gamma ($n = 25$), trouver l'information de Fisher et estimer la covariance asymptotique. A partir de là, construire un intervalle de confiance asymptotique de niveau 0.95. Comparer avec la solution de 11.

$$\mathcal{L}(\alpha, \beta|x) = \alpha \log \beta - \log \Gamma(\alpha) + (\alpha - 1) \log x - \beta x$$

Calcul du Hessien:

$$\frac{\partial \mathcal{L}(\alpha, \beta|x)}{\partial \alpha} = \log \beta - \frac{\partial \log \Gamma(\alpha)}{\partial \alpha} + \log(x) \qquad \frac{\partial^2 \mathcal{L}(\alpha, \beta|x)}{\partial \alpha^2} = -\frac{\partial^2 \log \Gamma(\alpha)}{\partial \alpha^2}$$

$$\frac{\partial \mathcal{L}(\alpha, \beta|x)}{\partial \beta} = \frac{\alpha}{\beta} - x \qquad \frac{\partial^2 \mathcal{L}(\alpha, \beta|x)}{\partial^2 \beta} = -\frac{\alpha}{\beta^2}$$

$$\frac{\partial^2 \mathcal{L}(\alpha, \beta|x)}{\partial \alpha \partial \beta} = \frac{1}{\beta}$$

Donc

$$I(\alpha,\beta) = \begin{pmatrix} \frac{\partial^2 \log\Gamma(\alpha)}{\partial\alpha^2} & -\frac{1}{\beta} \\ -\frac{1}{\beta} & \frac{\alpha}{\beta^2} \end{pmatrix}$$

De plus, comme $\hat{\theta} \approx \mathcal{N}(\theta, I_n(\theta)^{-1})$, On a

$$\sqrt{n}(\hat{\theta} - \theta^*) \xrightarrow{\mathcal{L}} \mathcal{N}(0, I_1^{-1}(\theta^*))$$

Donc $Var(\hat{\theta}) = I_1^{-1}(\theta)$

On sait que $I(\theta) = nI_1(\theta)$

D'où

$$I_1(\theta)^{-1} = \frac{1}{n} I(\theta) = \frac{1}{det(I(\theta))} \begin{pmatrix} \frac{\partial^2 \log\Gamma(\alpha)}{\partial\alpha^2} & -\frac{1}{\beta} \\ -\frac{1}{\beta} & \frac{\alpha}{\beta^2} \end{pmatrix}$$

On commence par calculer le determinant:

$$det(I(\theta)) = \frac{\partial^2 \log\Gamma(\alpha)}{\partial\alpha^2} \times \frac{\alpha}{\beta^2} - \frac{1}{\beta^2} = \frac{1}{\beta^2}\left(\frac{\partial^2 \log\Gamma(\alpha)}{\partial\alpha^2} - 1\right)$$

Puis on trouve la covariance asymptotique:

$$Var(\hat{\theta}) = \frac{\beta^2}{\frac{\partial^2 \log\Gamma(\alpha)}{\partial\alpha^2} - 1} \begin{pmatrix} \frac{\alpha}{\beta^2} & \frac{1}{\beta} \\ \frac{1}{\beta} & \frac{\partial^2 \log\Gamma(\alpha)}{\partial\alpha^2} \end{pmatrix}$$

En général, il n'est pas possible d'évaluer l'information de Fisher de manière analytique, ou cela peut prendre trop de temps. Dans ce cas, on utilise l' "information observée" (sans ésperance), $I_O(\hat{\theta})^{-1} = -H(\hat{\theta})^{-1}$.

Nous pouvons utiliser l'option 'hessian=TRUE' dans `optim` pour obtenir la matrice hessienne et estimer la fonction de covariance par son inverse.

14. Estimer la covariance asymptotique avec l'information observée et construire un intervalle de confiance asymptotique de niveau 0.95.

```
hessien <- optim(par = x0, fn = logLGammaV2,echantillon=rgamma(25,2.5,1.5), method = "L-BFGS-B", contro
print(-solve(hessien[[6]]))
```

```
##           [,1]      [,2]
## [1,] 0.7416367 0.3523245
## [2,] 0.3523245 0.1962351
```

15. Nous utilisons la simulation pour comparer la performance de ces deux estimateurs. Simuler plusieurs fois un nouvel ensemble de données, construiser les deux intervalles de confiance et compter combien de fois cet intervalle contient les vraies valeurs. Quelle est votre conclusion?