

Projet de Statistiques

Projet 1 : Loi Beta avec deux paramètres inconnus

NAGUI BECHICHI ET ANTOINE ISNARDY

6 janvier 2016

Table des matières

Question 1 : Estimateur des moments du couple (α, β)	3
Question 2 : L'« <i>EMV</i> » du couple (α, β) ne se calcule que numériquement	4
Question 3 : Comparaison graphique des estimateurs	6
Question 4 : test de Wald de l'hypothèse $\alpha = \beta$	8
Annexe : Code <i>R</i>	11

Question 1 : Estimateur des moments du couple (α, β)

Soient (X_1, \dots, X_n) iid $\hookrightarrow Beta(\alpha, \beta)$, avec $\theta = (\alpha, \beta) \in (\mathbb{R}+^*)^2$
La densité d'une loi *Beta* est donnée par :

$$\forall x \in]0, 1[; f(x) = \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{\alpha-1} (1-x)^{\beta-1} \quad (1)$$

En notant la constante de normalisation : $B(\alpha, \beta) = \int_0^1 x^{\alpha-1} (1-x)^{\beta-1} dx$, on obtient :

$$\forall x \in]0, 1[; f(x) = \frac{1}{B(\alpha, \beta)} x^{\alpha-1} (1-x)^{\beta-1} \quad (2)$$

Calculons les moments d'une loi *Beta* (α, β)

$$\begin{aligned} E_\theta[X] &= \int_0^1 \frac{1}{B(\alpha, \beta)} x^\alpha (1-x)^{\beta-1} dx \\ &= \frac{B(\alpha+1, \beta)}{B(\alpha, \beta)} = \frac{\Gamma(\alpha+1)\Gamma(\beta)}{\Gamma(\alpha+1+\beta)} \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} \\ E_\theta[X] &= \frac{\alpha\Gamma(\alpha)\Gamma(\beta)\Gamma(\alpha+\beta)}{(\alpha+\beta)\Gamma(\alpha+\beta)\Gamma(\alpha)\Gamma(\beta)} \end{aligned}$$

Ainsi, on trouve : $\boxed{E_\theta[X] = \frac{\alpha}{\alpha+\beta}}$

$$\begin{aligned} E_\theta[X^2] &= \int_0^1 \frac{1}{B(\alpha, \beta)} x^{\alpha+1} (1-x)^{\beta-1} dx = \frac{\beta(\alpha+2, \beta)}{\beta(\alpha, \beta)} \\ &= \frac{\Gamma(\alpha+2)\Gamma(\beta)}{\Gamma(\alpha+2+\beta)} \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} \\ &= \frac{(\alpha+1)\alpha\Gamma(\alpha)\Gamma(\beta)\Gamma(\alpha+\beta)}{(\alpha+\beta+1)(\alpha+\beta)\Gamma(\alpha+\beta)\Gamma(\alpha)\Gamma(\beta)} \\ E_\theta[X^2] &= \frac{\alpha(\alpha+1)}{(\alpha+\beta)(\alpha+\beta+1)} \end{aligned}$$

Ainsi, à l'aide de la formule de Koenig Huygens,

$$\begin{aligned}
Var_{\theta}[X] &= E_{\theta}[X^2] - E_{\theta}[X]^2 \\
&= \frac{\alpha(\alpha+1)}{(\alpha+\beta)(\alpha+\beta+1)} - \frac{\alpha^2}{(\alpha+\beta)^2(\alpha+\beta+1)}
\end{aligned}$$

On trouve alors $Var_{\theta}[X] = \frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)}$

On estime θ par la méthode des moments :

$$\begin{aligned}
\left\{ \begin{array}{l} \bar{X}_n = \frac{1}{n} \sum X_i \\ V_n = \frac{1}{n-1} \sum (X_i - \bar{X}_n)^2 \end{array} \right\} &\iff \left\{ \begin{array}{l} \alpha = \frac{\beta \bar{X}_n}{1 - \bar{X}_n} \\ V_n = \frac{\beta^2 \bar{X}_n / (1 - \bar{X}_n)}{\left(\frac{\beta \bar{X}_n + \beta - \beta \bar{X}_n}{1 - \bar{X}_n} \right)^2 \left(\frac{\beta \bar{X}_n + \beta - \beta \bar{X}_n + 1 - \bar{X}_n}{1 - \bar{X}_n} \right)} = \frac{\bar{X}_n (1 - \bar{X}_n)^2}{\beta + 1 - \bar{X}_n} \end{array} \right. \\
&\iff \left\{ \begin{array}{l} \beta = \frac{\bar{X}_n (1 - \bar{X}_n)^2 - V_n (1 - \bar{X}_n)}{V_n} \\ \alpha = \left(\frac{\bar{X}_n (1 - \bar{X}_n)}{V_n} - 1 \right) \bar{X}_n \end{array} \right\} \iff \boxed{\begin{array}{l} \hat{\beta} = (1 - \bar{X}_n)^2 \left(\frac{\bar{X}_n (1 - \bar{X}_n)}{V_n} - 1 \right) \\ \hat{\alpha} = \left(\frac{\bar{X}_n (1 - \bar{X}_n)}{V_n} - 1 \right) \bar{X}_n \end{array}}
\end{aligned}$$

Question 2 : L'«EMV» du couple (α, β) ne se calcule que numériquement

Nous allons écrire la vraisemblance du modèle et montrer que l'estimateur du maximum de vraisemblance n'a pas de forme explicite.

$$L(X, \theta) = \prod_{i=1}^n f(x_i, \alpha, \beta) = \prod_{i=1}^n \frac{1}{\beta(\alpha, \beta)} x_i^{\alpha-1} (1-x_i)^{\beta-1} 1(x_i)_{0 < x_i < 1} \quad (3)$$

On a donc $L(X, \theta) = \beta(\alpha, \beta)^{-n} \prod_{i=1}^n x_i^{\alpha-1} (1-x_i)^{\beta-1} 1_{\min x_i > 0} 1_{\max x_i < 1}$

$$\forall x \in]0, 1[; l(X, \theta) = n [\ln(\Gamma(\alpha + \beta)) - \ln(\Gamma(\alpha)) - \ln(\Gamma(\beta))] + (\alpha-1) \sum_{i=1}^n \ln(x_i) + (\beta-1) \sum_{i=1}^n \ln(1-x_i) \quad (4)$$

Différencions la log-vraisemblance :

$$\begin{aligned}
\frac{\partial l}{\partial \alpha}(X, \theta) &= n \left[\frac{\Gamma'(\alpha + \beta)}{\Gamma(\alpha + \beta)} - \frac{\Gamma'(\alpha)}{\Gamma(\alpha)} \right] + \sum_{i=1}^n \ln(x_i) \\
\frac{\partial l}{\partial \beta}(X, \theta) &= n \left[\frac{\Gamma'(\alpha + \beta)}{\Gamma(\alpha + \beta)} - \frac{\Gamma'(\beta)}{\Gamma(\beta)} \right] + \sum_{i=1}^n \ln(1-x_i)
\end{aligned}$$

Appliquer la méthode du maximum de vraisemblance consiste à maximiser la log-vraisemblance. Toutefois, égaliser les deux équations ci-dessus à 0 nous ramène à un système non linéaire

qui ne possède pas de solution explicite. **L'estimateur du maximum de vraisemblance ne peut donc se calculer explicitement.**

Nous allons utiliser l'algorithme de Newton-Raphson :

Choisissons un θ_0 pour initialiser. On prend celui calculé lors de la question 1. Il est fortement convergent et asymptotiquement normal.

Les itérations s'écrivent ensuite :

$$\theta^{(t+1)} = \theta^t - \left(H^{(t)}\right)^{-1} \frac{\partial l}{\partial \theta}(\theta^{(t)})$$

avec $H^{(t)} = -n\hat{I}_1(t)$

Précisons la forme de H, et pour cela calculons :

$$\begin{aligned} \frac{\partial^2 l}{\partial \alpha^2}(X, \theta) \big|_{\alpha=\alpha(t), \beta=\beta(t)} &= n \left[\frac{\partial^2 \ln \Gamma(\alpha + \beta)}{\partial \alpha^2} - \frac{\partial^2 \ln \Gamma(\alpha)}{\partial \alpha^2} \right] = nm_{1,1} \\ \frac{\partial^2 l}{\partial \beta^2}(X, \theta) \big|_{\alpha=\alpha(t), \beta=\beta(t)} &= n \left[\frac{\partial^2 \ln \Gamma(\alpha + \beta)}{\partial \beta^2} - \frac{\partial^2 \ln \Gamma(\beta)}{\partial \beta^2} \right] = nm_{2,2} \\ \frac{\partial^2 l}{\partial \alpha \partial \beta}(X, \theta) \big|_{\alpha=\alpha(t), \beta=\beta(t)} &= n \left[\frac{\partial^2 \ln \Gamma(\alpha + \beta)}{\partial \alpha \partial \beta} \right] = nm_{1,2} = nm_{2,2} \end{aligned}$$

D'où

$$H = n \begin{pmatrix} m_{1,1} & m_{1,2} \\ m_{2,1} & m_{2,2} \end{pmatrix}$$

L'algorithme se termine lorsque l'écart de la valeur entre deux itérations sera inférieur à un seuil arbitraire.

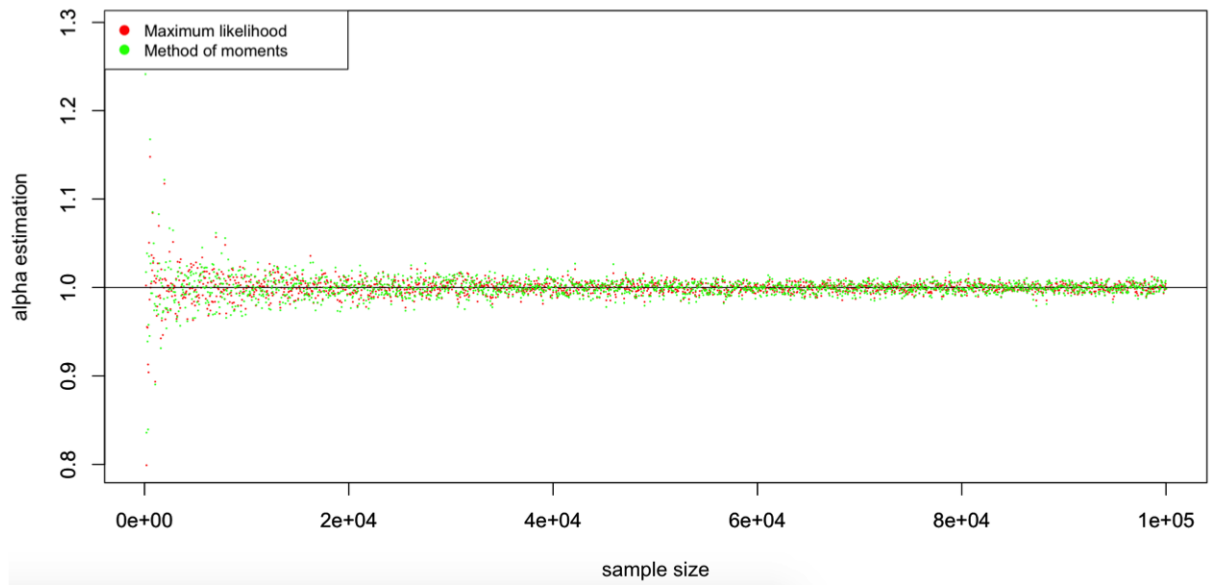
Après recherche, nous savons qu'il est possible d'exprimer les termes ci-dessus à l'aide des fonctions digamma et trigamma définies respectivement comme :

$$\begin{aligned} \psi^{(0)}(x) &= \frac{\partial \ln \Gamma(x)}{\partial x} \\ \psi^{(1)}(x) &= \frac{\partial^2 \ln \Gamma(x)}{\partial x^2} \end{aligned}$$

D'où,

$$\begin{aligned} m_{1,1} &= \psi^{(1)}(\alpha + \beta) - \psi^{(1)}(\alpha) \\ m_{2,1} &= \psi^{(1)}(\alpha + \beta) \\ m_{2,2} &= \psi^{(1)}(\alpha + \beta) - \psi^{(1)}(\beta) \end{aligned}$$

La suite de la question 2 est traitée dans notre code R .

FIGURE 1 – Estimation du paramètre α selon la taille de l'échantillon

Question 3 : Comparaison graphique des estimateurs

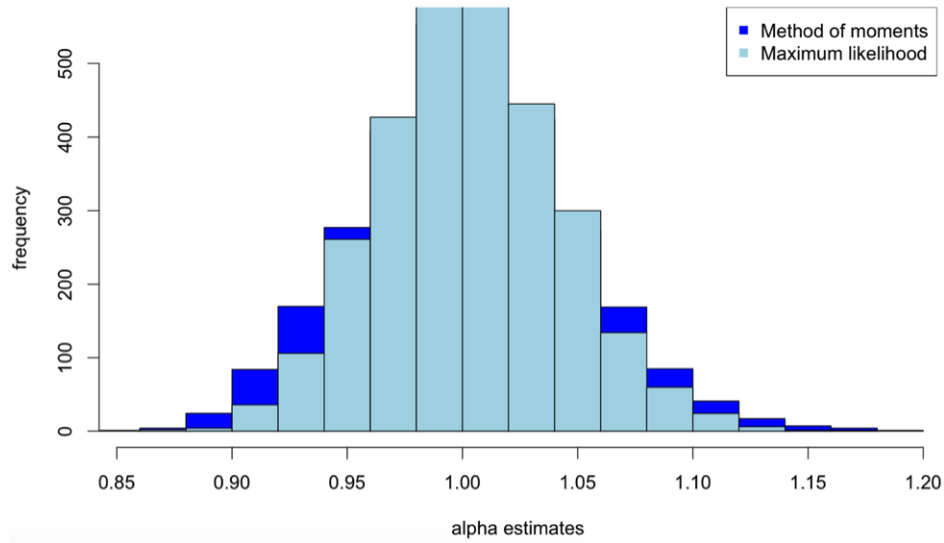
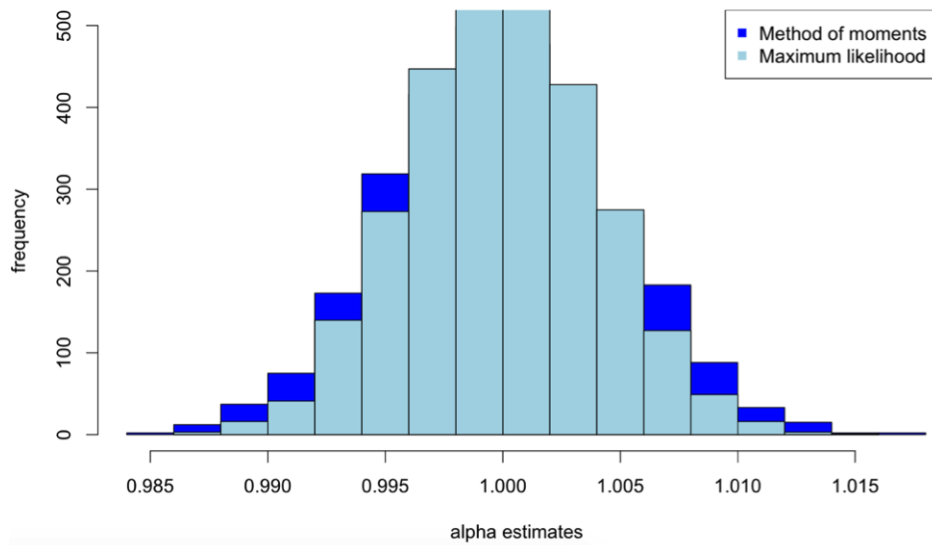
Les propriétés de α et β étant identiques, nous nous sommes limités à l'étude de α . Nous avons généré trois graphiques afin de nous aider à comparer l'estimateur du maximum de vraisemblance et l'estimateur des moments. Dans chaque étude, nous nous sommes appuyés sur une loi *Beta* de paramètre $\alpha = 1$.

Dans la Figure 1, nous nous sommes intéressés à l'importance de la taille de l'échantillon, depuis un comportement non asymptotique, jusqu'à un comportement que nous avons jugé asymptotique. Nous avons pour cela simulé 2000 (1999 exactement...) échantillons, de 100 variables générées à 100000 par pas de 50. Nous avons calculé, pour chacun de ces échantillons, des estimateurs de α et β via le maximum de vraisemblance (implémenté à l'aide de l'algorithme de Newton-Raphson), et la méthode des moments.

Nous remarquons l'importance de la taille de l'échantillon. Plus elle est importante, plus l'estimation de α converge vers la vraie valeur du paramètre : en effet, nous voyons que dans la partie droite du graphique (correspondant au comportement asymptotique), les valeurs de l'estimation par les deux méthodes sont très proches de la vraie valeur. **Cela nous permet de dire que ces deux méthodes permettent d'estimer nos paramètres sans biais.**

Nous nous sommes ensuite intéressés à la distribution suivie par ces estimateurs dans les Figures 2 et 3. Pour cela, nous avons calculé 3000 estimateurs à partir d'échantillons de taille 1000 et 100000. Nous avons représenté les résultats sous forme d'histogramme.

Dans un premier temps, ces deux histogrammes viennent confirmer ce que nous avons observé précédemment : la valeur de l'estimateur converge vers sa vraie valeur, et ce d'autant plus que la taille de l'échantillon augmente (cf l'échelle des abscisses de chacun des deux graphes pour s'en persuader). **Par ailleurs, nous remarquons que la valeur de l'estimateur converge en loi vers une distribution normale d'espérance 1, et ce quelle que soit la méthode utilisée.**

FIGURE 2 – Distribution des deux estimateurs de α (taille=1 000)FIGURE 3 – Distribution des deux estimateurs de α (taille=100 000)

Intéressons-nous enfin à la variance des estimateurs de ces deux méthodes. Nous sommes renseignés dès la Figure 1 : les points rouges associés à l'estimateur du maximum de vraisemblance semblent davantage concentrés autour de la vraie valeur du paramètre que ne le sont les points verts associés à la méthode des moments. **Cela nous permet de penser que la variance associée à la méthode des moments est plus élevée.** Cette observation se confirme dans les Figures 2 et 3. En effet, les estimations associées à la méthode des moments sont davantage présentes aux extrémités de la distribution, alors que celles associées au maximum de vraisemblance sont concentrées autour de la vraie valeur du paramètre. **Nous pouvons donc dire que l'estimateur du maximum de vraisemblance est plus efficace que la méthode des moments.**

Question 4 : test de Wald de l'hypothèse $\alpha = \beta$

Nous allons adapter l'algorithme de Newton-Raphson pour effectuer un test de Wald de l'hypothèse $H_0 : \alpha = \beta$

Construction du test de Wald

- On pose $g(\theta) = g(\alpha, \beta) = \alpha - \beta$ afin de tester l'hypothèse $H_0 : g(\theta) = 0$
- La région critique du test de Wald est donnée par :

$$W_n = \{\xi_n > \chi_r^2(1 - R)\}$$

avec ξ_n la statistique de test, r la dimension de $g(\theta)$ (1 ici), et R le niveau asymptotique du test (nous prendrons 5% dans l'application).

- La statistique de test est donnée par :

$$\xi_n = ng(\hat{\theta}_n)' \left[\frac{\partial g}{\partial \theta'}(\hat{\theta}_n) I_1(\hat{\theta}_n) \frac{\partial g'}{\partial \theta}(\hat{\theta}_n) \right]^{-1} g(\hat{\theta}_n)$$

Précisons les termes de la matrice de variance covariance du couple $(\hat{\alpha}, \hat{\beta})$:

$$(nI_1(\hat{\theta}))^{-1} = \begin{pmatrix} V(\hat{\alpha}) & cov(\hat{\alpha}, \hat{\beta}) \\ cov(\hat{\alpha}, \hat{\beta}) & V(\hat{\beta}) \end{pmatrix}$$

D'où :

$$\xi_n = \frac{g(\hat{\theta})^2}{V(\hat{\alpha}) + V(\hat{\beta}) - 2cov(\hat{\alpha}, \hat{\beta})}$$

- On sait que sous H_0 , $\xi_n \hookrightarrow \chi_1^2$

Adaptation de l'algorithme de Newton-Raphson

- On calcule des estimations de α et β à partir de l'implémentation précédente.
- On calcule ξ_n à l'aide de $\hat{\alpha}, \hat{\beta}, I_1(\hat{\theta}_n)$. Le calcul de la matrice de variance-covariance s'obtient grâce à l'inverse de l'information de Fisher (à un facteur $1/n$ près).
- *Application* : pour $\alpha = 3$ et $\beta = 3$, on trouve par exemple 0,2238307. Le quantile étant 3,841459, on échoue donc à rejeter H_0

Seuil non asymptotique

Comme dit plus haut, sous H_0 , $\xi_n \hookrightarrow \chi_1^2$

Cherchons pour quelle valeur de n ce n'est plus le cas.

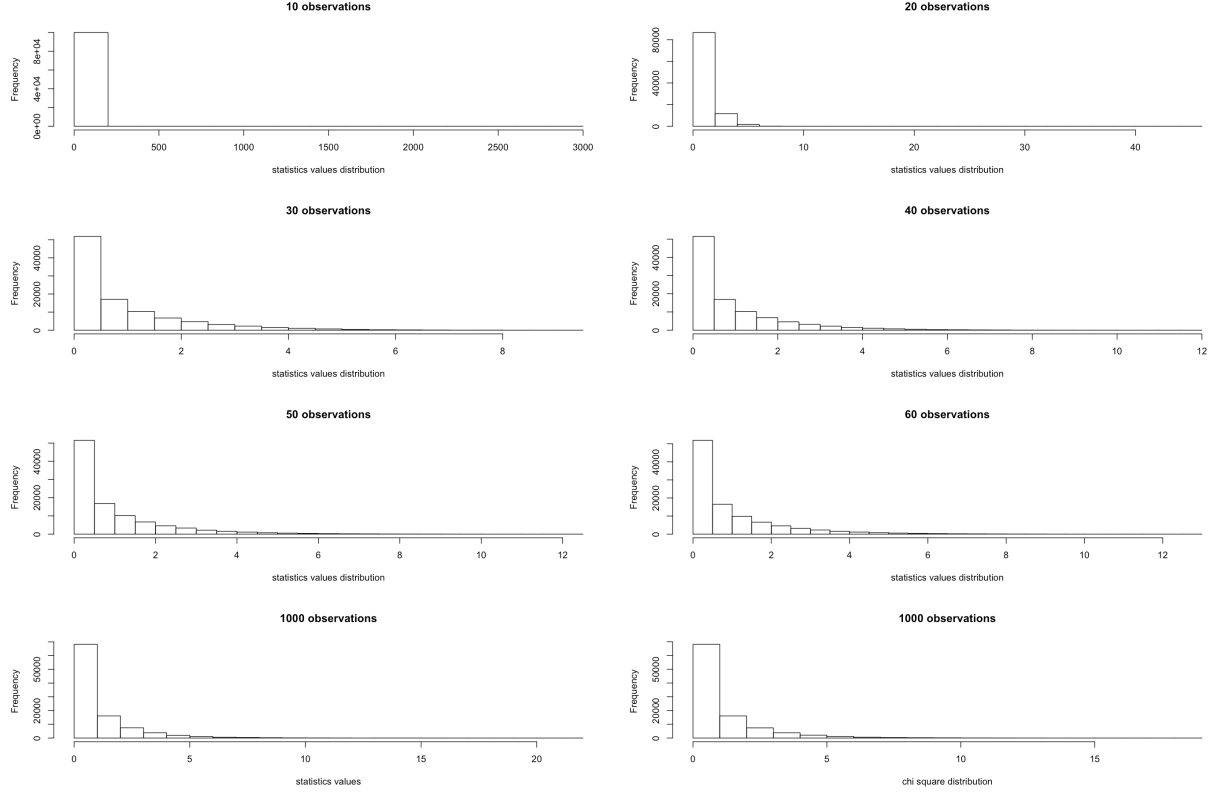
Pour cela, recherchons la distribution de ξ_n pour diverses valeurs de n . Nous générons $(X_1, \dots, X_n) \hookrightarrow \text{Beta}(\alpha, \alpha)$ (sous H_0) pour $n=10, 20, 30, 40, 50, 1000$ et observons la distribution de ξ_n après avoir calculé 100 000 fois cette statistique.

La figure 4, complétée du tableau 1 recensant les différents quantiles correspondants, prouvent deux choses.

Premièrement, au regard des deux histogrammes du bas, pour n large ($=1000$), la distribution $\xi_n \hookrightarrow \chi_1^2$

Par ailleurs, on se rend bien compte que plus n est faible, moins notre distribution empirique est proche de la distribution du χ_1^2 .

L'analyse des quantiles nous permet de dire que nous ne suivons plus une χ_1^2 pour $n=30$ environ.

FIGURE 4 – Comparaison des distributions de ξ_n avec la χ^2_2 en fonction de n

	q_0	$q_{0,25}$	$q_{0,5}$	$q_{0,75}$	q_1
Quantiles simulés pour 10 observations	9.405e-12	1,130e-01	4,789e-01	1,192e00	2,813e03
Quantiles simulés pour 20 observations	7.288e-10	1,079e-01	4,636e-01	1,263e00	4,554e01
Quantiles simulés pour 30 observations	3.649e-11	1,062e-01	4,608e-01	1,273e00	9,209e01
Quantiles simulés pour 40 observations	1,639e-11	1,063e-01	4,674e-01	1,303e00	1,151e00
Quantiles simulés pour 50 observations	3,003e-12	1,035e-01	4,665e-01	1,308e00	1,239e01
Quantiles simulés pour 60 observations	4,448e-11	1,039e-01	4,578e-01	1,307e00	1,259e01
Quantiles de la loi de Chi-deux	1,574e-10	1,026e-01	4,574e-01	1,325e00	1,858

TABLE 1 – Comparaison des quantiles avec celui de la Chi-deux en fonction de n

Annexe : Code *R*

```

# Data generation

## Generation of  $X_1, \dots, X_n$  following a  $Beta(\alpha, \beta)$ 
f.generate <- function(nobs, alpha = 1, beta = 3) {
  data <- rbeta(nobs, alpha, beta)
  return(data)}

# Method of moments

f.methodOfMoments <- function(data) {
  ## Empirical mean and variance
  xBar <- mean(data)
  v <- var(data)
  ## alpha_hat and beta_hat estimates
  alpha_hat <- xBar*(xBar*(1-xBar)/v-1)
  beta_hat <- (1-xBar)*(xBar*(1-xBar)/v-1)
  ## Resultat
  return(c(alpha_hat, beta_hat)) }

#Newton-Raphson algorithm

f.newtonRaphson <- function(data, precisionThreshold = 0.0001) {
  ## Initialization: theta_hat from method of moments
  ## Vector of precision to compare with, at each iteration
  ##termination if the difference between 2 iterations is under precisionThreshold
  theta_hat = f.methodOfMoments(data)
  precision <- c(10000, 10000)
  # Initialization of alpha and beta precision to huge values
  ## Iterations
  while (abs(precision[1] > precisionThreshold) | precision[2] > precisionThreshold) {
    alpha_hat <- theta_hat[1]
    beta_hat <- theta_hat[2]
    ### Let's compute the gradient of the likelihood function and its hessian matrix
    ### using digamma and trigamma functions for simpler calculus
    ### We get rid of n (number of observations) as they go
    ###when we take the product of gradient-1 with hessian
    gradient <- matrix(c(digamma(alpha_hat+beta_hat)
      - digamma(alpha_hat) + mean(log(data)), digamma(alpha_hat+beta_hat)
      - digamma(beta_hat) + mean(log(1-data))), 2,1)
    hessian <- solve(matrix(c(trigamma(alpha_hat+beta_hat) - trigamma(alpha_hat),
      trigamma(alpha_hat+beta_hat),
      trigamma(alpha_hat+beta_hat), trigamma(alpha_hat+beta_hat) - trigamma(beta_hat)),
    ### solve = inverse matrix

    ### Iteration updates
    precision <- hessian %*% gradient
    theta_hat <- theta_hat - precision  }
  return(theta_hat)
}

```

```

# Comparison of the maximum likelihood and the method of moments

# Alpha and Beta properties are the same. Let's focus on Alpha
## How does the sample size play a role?
## Chart 1
### Retrieve estimates for different sample sizes
nobs <- seq(100, 100000, by=50)
alpha_methodOfMoments <- vector("numeric", length(nobs))
alpha_newtonRaphson <- vector("numeric", length(nobs))
for (i in seq(1:length(nobs))) {
  data <- f.generate(nobs = nobs[i])
  alpha_methodOfMoments[i] <- f.methodOfMoments(data)[1]
  alpha_newtonRaphson[i] <- f.newtonRaphson(data)[1]
}

### Plot

plot.new()
par(mfrow=c(1,1))
plot(nobs,alpha_newtonRaphson,
type="p",xlim=range(nobs),
ylim=range(c(alpha_newtonRaphson,alpha_methodOfMoments)),
pch=19,cex = 0.1, axes = TRUE, col="red ", xlab = "sample size",
ylab = "alpha estimation")
par(new = TRUE)
abline(h = 1, col = "black")
par(new = TRUE)
plot(nobs,alpha_methodOfMoments, type="p",xlim=range(nobs),
ylim=range(c(alpha_newtonRaphson,alpha_methodOfMoments)),
pch=19,cex = 0.1, axes = TRUE, col="green ", xlab = "", ylab = "")
legend("topleft", legend = c("Maximum likelihood",
"Method of moments"), col=c("red","green"),
pch=c(19,19) , bty="p", pt.cex = 1, cex = 0.8)
dev.off()

## What does the alpha distribution look like?

## Charts 2&3
f.drawHistogram <- function(nobs, nSamples = 3000) {
  ### Retrieve estimates
  alpha_methodOfMoments <- vector("numeric", length(nSamples))
  alpha_newtonRaphson <- vector("numeric", length(nSamples))
  for (i in seq(1:nSamples)) {
    data <- f.generate(nobs)
    alpha_methodOfMoments[i] <- f.methodOfMoments(data)[1]
    alpha_newtonRaphson[i] <- f.newtonRaphson(data)[1] }

  ### Plot
  histogramMOM <- hist(alpha_methodOfMoments)
  histogramNR <- hist(alpha_newtonRaphson)
  histogramRange <- range(c(alpha_methodOfMoments, alpha_newtonRaphson))

```

```

plot.new()
par(mfrow = c(1,1))
plot(histogramMOM,col= "blue",xlim= histogramRange,
xlab = "alpha estimates",ylab = "frequency")
plot(histogramNR,col="lightblue",xlim=histogramRange,add=T)
legend("topright", c("Method of moments","Maximum likelihood"),
col=c("blue","lightblue") , pch=15:15) }
f.drawHistogram(nobs = 1000)
f.drawHistogram(nobs = 100000)

# Adapated Newton Raphson and Wald test statistics
f.newtonRaphsonAdapted <- function(data, precisionThreshold = 0.0001) {
  ## Newton Raphson subsection
  theta_hat <- f.newtonRaphson(data = data, precisionThreshold = precisionThreshold)
  alpha_hat <- theta_hat[1]   beta_hat <- theta_hat[2]
  ## Test subsection
  ### Test Statistics
  fisherInformation <- matrix(c(trigamma(alpha_hat + beta_hat) - trigamma(alpha_hat),
  trigamma(alpha_hat + beta_hat), trigamma(alpha_hat + beta_hat),
  trigamma(alpha_hat + beta_hat) - trigamma(beta_hat)), 2,2)
  varCov <- 1/length(data) * solve(fisherInformation)
  statChiSquare <- (alpha_hat - beta_hat)^2 /
  (varCov[1,1] + varCov[2,2] - 2*varCov[1,2])
  return(abs(statChiSquare)) }

# Wald Test
f.waldTest <- function(alpha, nobs) {
  data <- f.generate(nobs = nobs, alpha = alpha, beta = alpha)
  return(f.newtonRaphsonAdapted(data = data)) }
qchisq(0.95, df=1, ncp=0, lower.tail=TRUE, log.p=FALSE)
f.waldTest(alpha = 3, 1000000)

# Asymptotic threshold value

## Generate statistics distribution under H0: alpha = beta
### genNumber: how many times do we compute the statistics for each num of obs?
f.statisticsDistribution <- function(nobs, genNumber = 100000) {
  statisticsValues <- vector("numeric", genNumber)   for (i in seq(1:genNumber)) {
  data <- f.generate(nobs = nobs, alpha = 2, beta = 2)
  statisticsValues[i] <- f.newtonRaphsonAdapted(data = data)   }
  result <- list("sampleSize" = nobs, "statisticsValues" = statisticsValues)
  return(result) }

distribution1000 <- f.statisticsDistribution(1000)
# Asymptotic distribution of the statistics for 100000 obs
chiSquareDistrib <- rchisq(1000, 1)
# True chi square distribution under one df and for 100000 obs

## Plot results for small sizes
## Plot results for a large sample size (1000)

```

```
## Plot a true chi square distribution with 1 df
nobs = seq(10, 60, by = 10) plot.new()
par(mfrow = c(4,2))
for (i in seq(1, length(nobs))) {
  statistics <- f.statisticsDistribution(nobs[i])
  print(paste(c("Quantiles for", nobs[i], "observations"), collapse = " "))
  print(quantile(statistics$statisticsValues))
  hist(statistics$statisticsValues,
  main = paste(c(nobs[i], "observations"), collapse = " "),
  xlab = "statistics values distribution") }
  hist(distribution1000$statisticsValues,
  main = paste(c(1000, "observations"), collapse = " "),
  xlab = "statistics values")
  hist(chiSquareDistrib, main = paste(c(1000, "observations"), collapse = " "),
  xlab = "chi square distribution")
  dev.off()
  quantile(distribution1000$statisticsValues)
  quantile(chiSquareDistrib)

finalTest <- f.generate(nobs = 20, alpha = 2, beta = 3)
f.newtonRaphsonAdapted(data = finalTest)
qchisq(0.95, df=1, ncp=0, lower.tail=TRUE, log.p=FALSE)
```