

# TP3-MPA-2022

KERZAZI Achraf

CHERKAOUI Ayoub

OUAZZANI CHAHDI Mohammed

## Première partie

### Question 1

On a :

$$y = (y_1, y_2, \dots, y_n)$$

$$\begin{aligned} p(y|\theta) &= \prod_{i=1}^n p(y_i|\theta) \\ &= \prod_{i=1}^n \frac{\theta^{y_i} \exp(-\theta)}{y_i!} \\ &= \frac{\theta^{\sum_{i=1}^n y_i} \exp(-n\theta)}{\prod_{i=1}^n y_i!} \end{aligned}$$

Posons  $\bar{y} = \sum_{i=1}^n y_i$

Alors :

$$p(y|\theta) = \frac{\theta^{\bar{y}} \exp(-n\theta)}{\prod_{i=1}^n y_i!}$$

### Question 2:

On a :

$$\begin{aligned} p(\theta|y) &\propto p(y|\theta)p(\theta) \\ &\propto \exp(-n\theta) \theta^{\bar{y}} \frac{1}{\theta} \\ &\propto \theta^{\bar{y}-1} \exp(-n\theta) \end{aligned}$$

Donc:

$$\theta|y \sim \Gamma(\bar{y}, 1/n)$$

### Question 3:

Principe de l'algorithme de Metropolis-Hasting:

partant de  $\theta_0$  fixé, l'algorithme est défini à l'itération  $t$  par :

On tire  $\theta^* \sim q(\theta^{t-1}, \theta^*)$  avec  $q$  la loi instrumentale

Calculer  $r = \frac{\pi(\theta^*)q(\theta^{t-1}, \theta^*)}{\pi(\theta^{t-1})q(\theta^*, \theta^{t-1})}$

Tirer  $u \sim \mathcal{U}(0, 1)$  : si  $u < r$  alors  $\theta^t = \theta^*$ , sinon  $\theta^t = \theta^{t-1}$

Prenons comme loi instrumentale  $q(\theta^t, \theta^*)$  la loi exponentielle de paramètre  $\frac{1}{\theta^t}$

Alors l'algorithme s'écrit de la manière suivante en R:

```
data <- read.table("/Users/simo/Desktop/data.txt")
y <- c()
for ( i in 1:length(data)){
  for (j in 1:length(data[[i]])){
    y <- c(y, data[[i]][j])
  }
}
n <- length(y)

theta_MH <- c(10)
nb_sim <- 10000
for ( i in 1 : nb_sim){
  theta_t <- theta_MH[length(theta_MH)][[1]]
  theta_e <- rexp(1,1/theta_t)

  r1 <- dgamma(theta_e, shape=sum(y), scale = 1/n) / dgamma(theta_t, shape=sum(y), scale = 1/n)
  r2 <- dexp(theta_t,rate=1/theta_e) / dexp(theta_e,rate=1/theta_t)
  r <- r1 * r2

  u <- runif(1)
  if (u < r){
    theta_MH <- c(theta_MH,theta_e)
  }else{
    theta_MH <- c(theta_MH,theta_t)
  }
}
```

#### Question 4

```
data <- read.table("/Users/simo/Desktop/data.txt")
y <- c()
for ( i in 1:length(data)){
  for (j in 1:length(data[[i]])){
    y <- c(y, data[[i]][j])
  }
}
n <- length(y)

theta.post <- rgamma(n = 10000, shape=sum(y), scale = 1/n )
moyenne.post <- mean(theta.post)
median.post <- median(theta.post)

int_conf <- function(loi){
  lon <- lengt(loi)
  triee <- sort(loi)
}
```

```

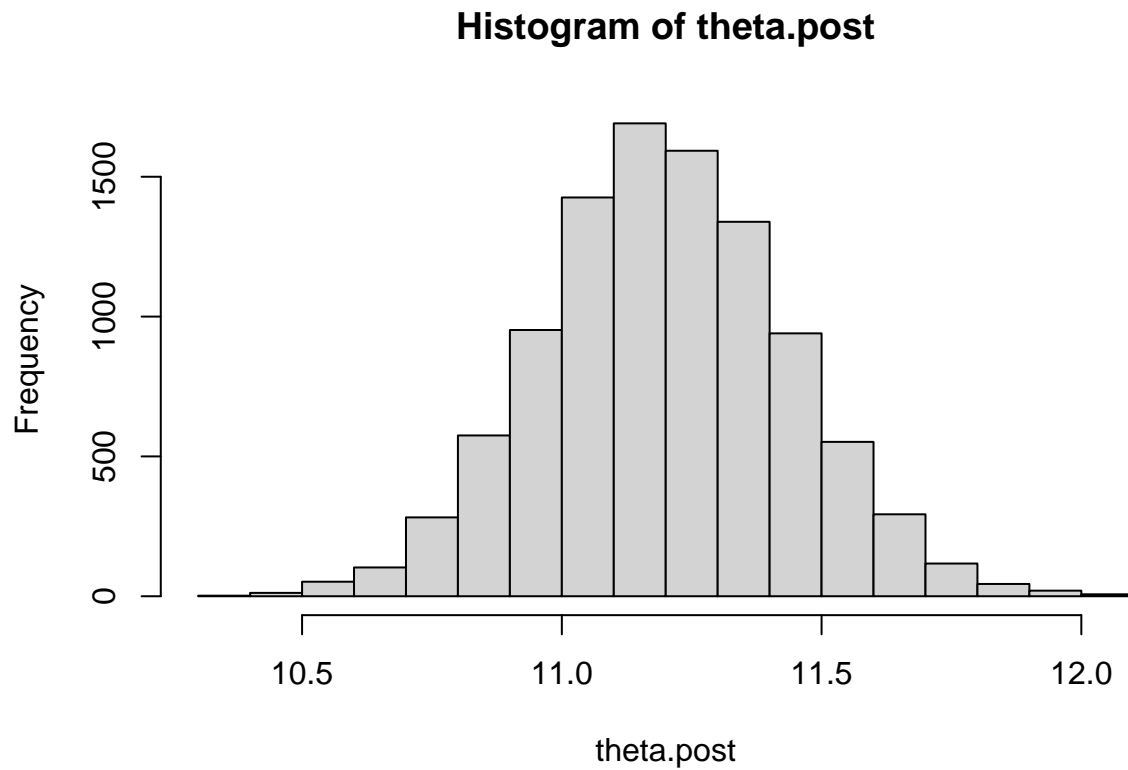
reorganiser_theta_selon_proba <- function(thetaf,yf ){
  nf <- length(thetaf)
  x <- matrix(0,2,nf)
  x[1,] <- thetaf
  x[2,] <- dgamma(thetaf,shape=sum(yf), scale = 1/nf )
  for (i in 1:nf){
    i_max <- 1
    v_max <- x[2,i_max]
    ppp <- nf-i+1
    for(j in 1:ppp){
      if (x[2,j] > v_max){
        i_max <- j
        v_max <- x[2,i_max]
      }
    }
    copy <- x[,nf-i+1]
    x[,nf-i+1] <- x[,i_max]
    x[,i_max] <- copy
  }
  return(x)
}

construire_IC <- function(matrice,nf){
  val <- round(nf*0.25)
  points_in_IC <- matrice[1,][val:nf]
  a <- min(points_in_IC )
  b <- max(points_in_IC )
  return(c(a,b))
}

mat <- reorganiser_theta_selon_proba(theta.post,y )
IC <- construire_IC(mat,length(theta.post))

# un intervalle de confiance à 95% au tour de la valeur la plus probable est IC
hist(theta.post)

```



Remarque : un intervalle de confiance à 95% au tour de la valeur la plus probable est IC

#### Question 5

```
data <- read.table("/Users/simo/Desktop/data.txt")
y <- c()
for ( i in 1:length(data)){
  for (j in 1:length(data[[i]])){
    y <- c(y, data[[i]][j])
  }
}
n <- length(y)
moyenne_y <- mean(y)

theta_MH <- c(10)
nb_sim <- 10000
for ( b in 1 : nb_sim){
  theta_t <- theta_MH[length(theta_MH)][[1]]
  theta_e <- rexp(1,1/theta_t)

  r1 <- dgamma(theta_e, shape=sum(y), scale = 1/n) / dgamma(theta_t, shape=sum(y), scale = 1/n)
  r2 <- dexp(theta_t,rate=1/theta_e) / dexp(theta_e,rate=1/theta_t)
  r <- r1 * r2

  u <- runif(1)
  if (u < r){
    theta_MH <- c(theta_MH,theta_e)
  }
}
```

```

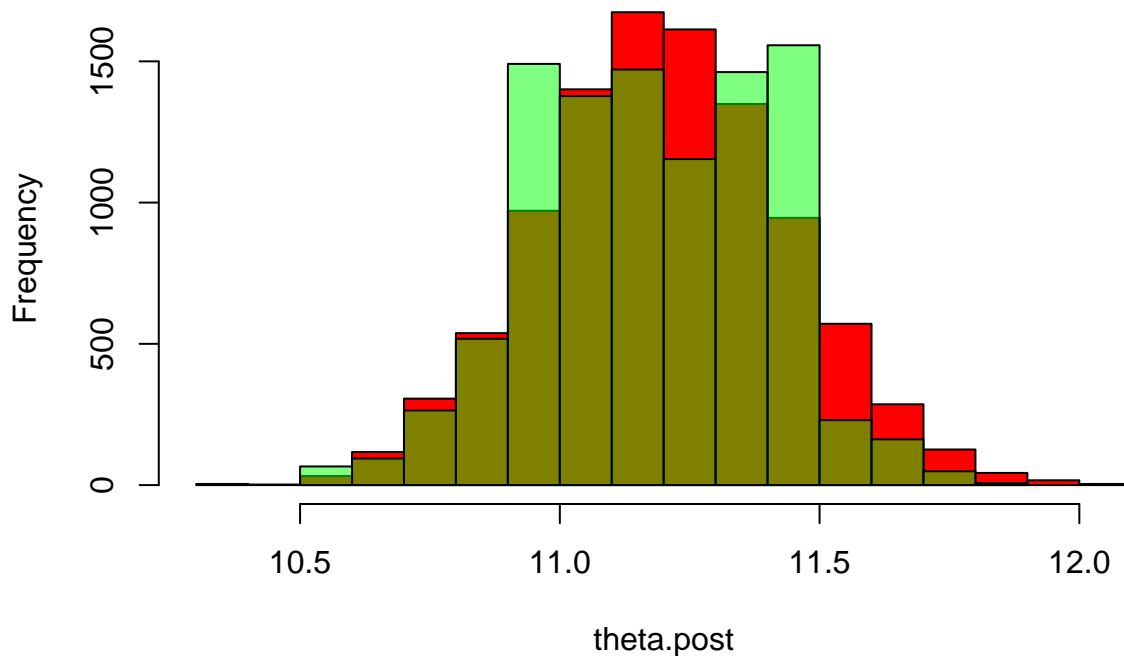
}else{
  theta_MH <- c(theta_MH,theta_t)
}
}
moyenne_MH <- mean(theta_MH)

theta.post <- rgamma(n = 10000, shape=sum(y), scale = 1/n )
moyenne.post <- mean(theta.post)
median.post <- median(theta.post)
intervalle_de_confiance <- t.test(theta.post, conf.level=0.95 )$conf.int

hist(theta.post, col="red")
hist(theta_MH[100:length(theta_MH)],add = T, col=rgb(0, 1, 0, 0.5))

```

**Histogram of theta.post**



La couleur rouge donne la partie de l'histogramme de la loi à postérieure de  $\theta$  qui n'a pas d'intersection avec l'histogramme obtenue par l'algorithme de Metropolis Hastings, et la couleur vert donne la partie de l'histogramme obtenue par l'algorithme de Metropolis Hastings de  $\theta$  qui n'a pas d'intersection avec l'histogramme de la loi à postérieure et en jaune foncé on a l'intersection de ces 2 histogrammes.

## Question 6

$$\begin{aligned}
p(y^*|y) &= \int_0^{+\infty} p(y^*|\theta)p(\theta|y)d\theta \\
&= \int_0^{+\infty} \frac{\theta^{y^*} \exp(-\theta)}{y^*!} \frac{n^{\bar{y}} \theta^{\bar{y}-1} \exp(-n\theta)}{\Gamma(\bar{y})} d\theta \\
&= \frac{n^{\bar{y}}}{y^*! \Gamma(\bar{y})} \int_0^{+\infty} \theta^{y^*!} \exp(-\theta) \theta^{\bar{y}-1} \exp(-n\theta) d\theta \\
&= \frac{n^{\bar{y}}}{y^*! \Gamma(\bar{y})} \int_0^{+\infty} \theta^{y^*!+\bar{y}-1} \exp(-\theta(n+1)) d\theta \\
&= \frac{n^{\bar{y}}}{y^*! \Gamma(\bar{y})} \frac{\Gamma(\bar{y}+y^*)}{(n+1)^{\bar{y}+y^*}} \int_0^{+\infty} \frac{(n+1)^{\bar{y}+y^*}}{\Gamma(\bar{y}+y^*)} \theta^{y^*+\bar{y}-1} \exp(-\theta(n+1)) d\theta
\end{aligned}$$

Le terme dans l'intégrale à droite est la densité de la loi

$$\Gamma(\bar{y}+y^*, \frac{1}{n+1})$$

Donc :

$$\begin{aligned}
p(y^*|y) &= \frac{n^{\bar{y}}}{y^*! \Gamma(\bar{y})} \frac{\Gamma(\bar{y}+y^*)}{(n+1)^{\bar{y}+y^*}} \\
&= \frac{\Gamma(\bar{y}+y^*)}{y^*! \Gamma(\bar{y})} \left(\frac{n}{n+1}\right)^{\bar{y}} \left(\frac{1}{n+1}\right)^{y^*} = \frac{\Gamma(\bar{y}+y^*)}{y^*! \Gamma(\bar{y})} \left(\frac{n}{n+1}\right)^{\bar{y}} \left(1 - \frac{n}{n+1}\right)^{y^*}
\end{aligned}$$

Donc la loi prédictive suit la loi négative binomiale de paramètres  $\bar{y}, \frac{n}{n+1}$

$$y^*|y \sim NegBinom(\bar{y}, \frac{n}{n+1})$$

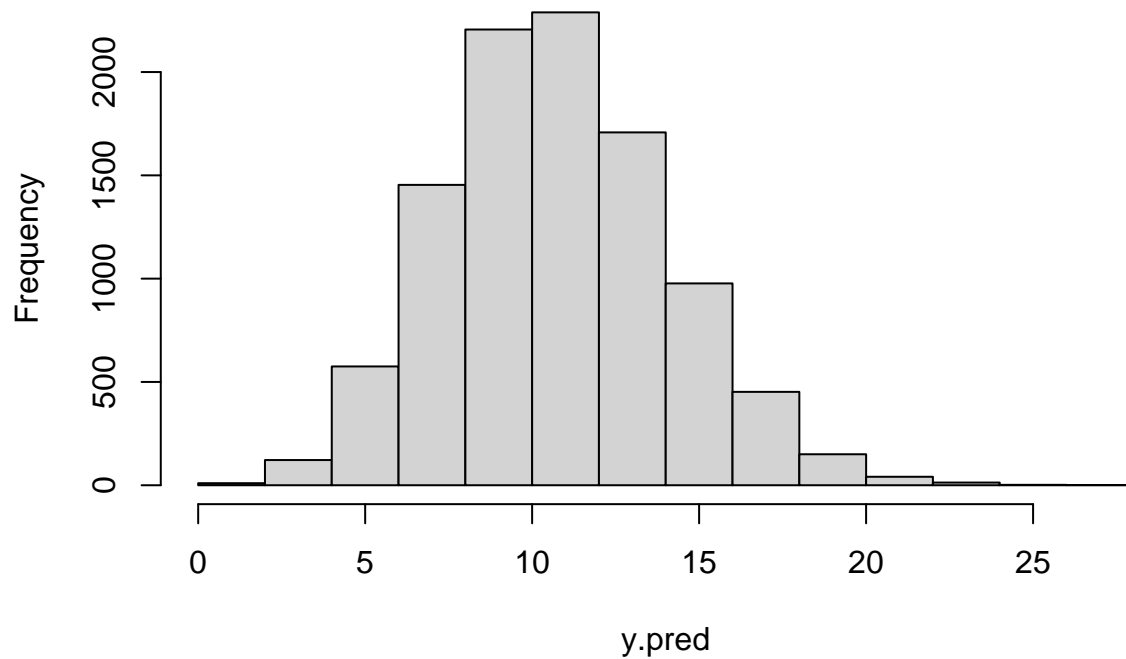
Algorithme de simulation et histogramme :

```

data <- read.table("/Users/simo/Desktop/data.txt")
y <- c()
for ( i in 1:length(data)){
  for (j in 1:length(data[[i]])){
    y <- c(y, data[[i]][j])
  }
}
n <- length(y)
y.pred <- rnbino(10000, size = sum(y), prob = n/(n+1))
hist(y.pred)

```

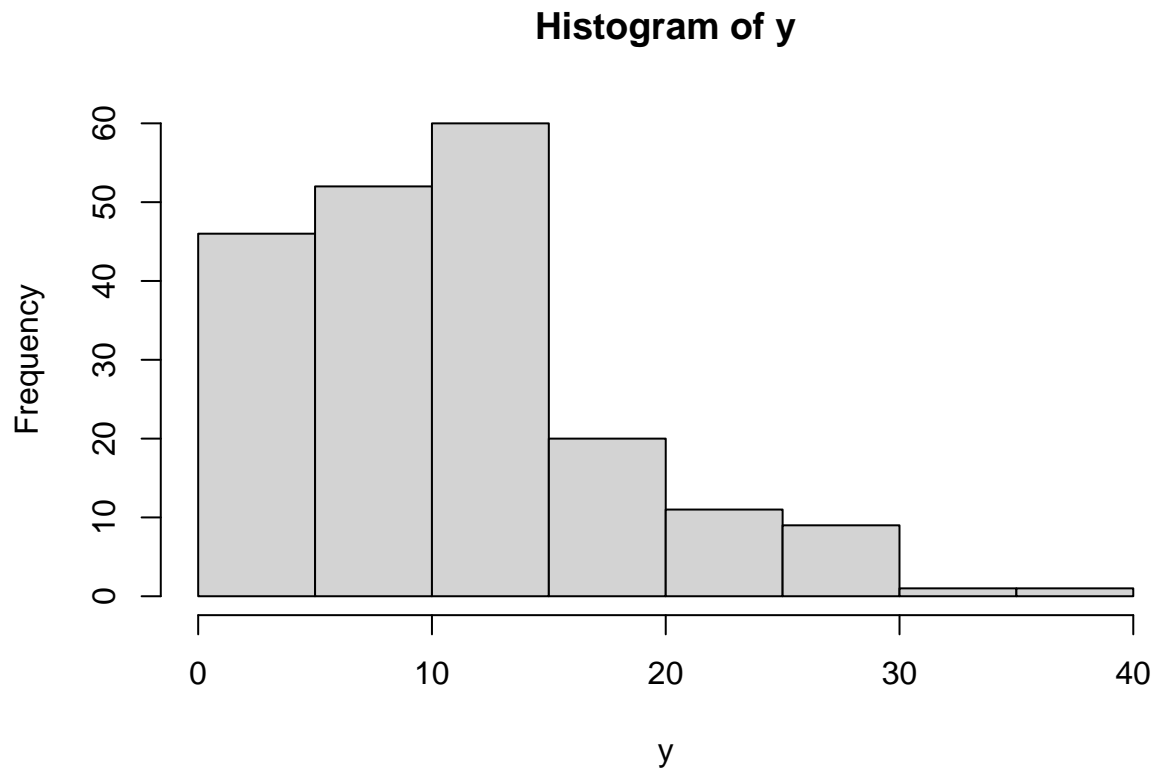
## Histogram of y.pred



```
moyenne.pred <- mean(y.pred)
moyenne.donnees <- mean(y)
```

Histogramme des données:

```
data <- read.table("/Users/simo/Desktop/data.txt")
y <- c()
for ( i in 1:length(data)){
  for (j in 1:length(data[[i]])){
    y <- c(y, data[[i]][j])
  }
}
hist(y)
```

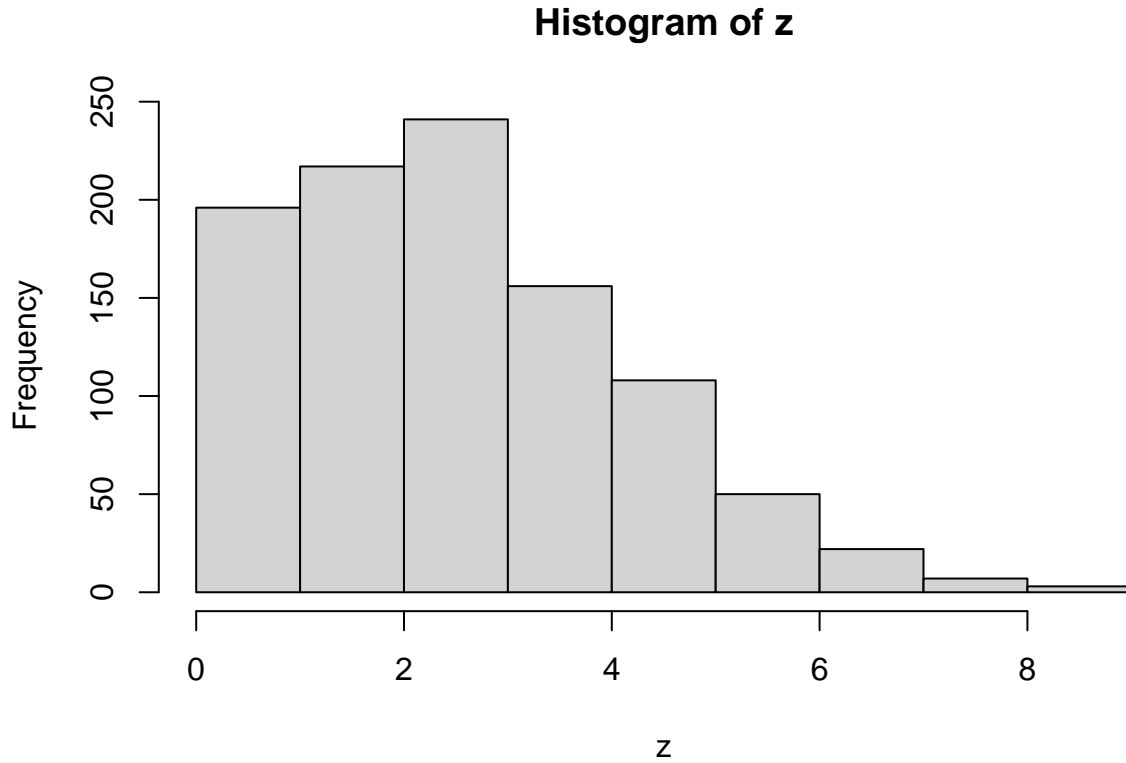


On voit que l'histogramme de résultats simulés est un peu à celui des données

Critiques: Le choix de la loi de poisson pour modéliser les données est bon choix parce que la loi de poisson pour un paramètre  $\lambda$  autour de 3 approche bien l'histogramme des données:

```
z <- rpois(1000, lambda = 3)
hist(z)
```





Par contre il se peut que pas toutes les  $y_i$  suivent cette loi pour les mêmes paramètres. Aussi le choix de la loi à priori est à revoir parce que par exemple pour des valeurs de  $\theta$  faibles la loi de Poisson de paramètre  $\theta$  n'est pas similaire à celle de données alors que la loi à priori choisie favorise ces valeurs. Un autre choix pour la loi à priori aurait pu être une gaussienne. Mais sûrement ce modèle n'est bien parce que l'histogramme de la loi prédictive n'est pas très proche de celui des données.

## Partie 2

### Question 1

On a  $I_k = \{i : z_i = k\}$  une partition de  $[1, n]$ , et puisque les données sont indépendantes on peut déduire que  $p(y|z, \theta) = p(y_i|z_i, \theta)$  Alors:

$$\begin{aligned}
 p(y|z, \theta) &= \prod_{k=1}^K \prod_{i \in I_k} p(y_i|z_i, \theta) \\
 &= \prod_{k=1}^K \prod_{i \in I_k} (\theta_k)^{y_i} \frac{\exp(-\theta_k)}{y_i!} \\
 &= \frac{\exp(-\sum_{k=1}^K \#(I_k) \theta_k)}{\prod_{i=1}^n y_i!} \prod_{k=1}^K \theta_k^{\sum_{i \in I_k} y_i}
 \end{aligned}$$

### Question 2

On peut écrire la loi à postérieure sous la forme:

$$\begin{aligned}
p(z, \theta|y) &\propto p(y|z, \theta)p(\theta)p(z) \\
&\propto \frac{\exp(-\sum_{k=1}^K \#(I_k)\theta_k)}{\prod_{i=1}^n y_i!} \prod_{k=1}^K \theta_k^{\sum_{I_k} y_i} \prod_{k=1}^K \frac{1}{\theta_k} \\
&\propto \frac{\exp(-\sum_{k=1}^K \#(I_k)\theta_k)}{\prod_{i=1}^n y_i!} \prod_{k=1}^K \theta_k^{\sum_{I_k} y_i - 1}
\end{aligned}$$

### Question 3

Calculons la loi conditionnelle de  $z_i = k|y, \theta$

On a par la formule de Bayes:

$$\begin{aligned}
p(z_i = k|y, \theta) &= p(z_i = k|y_i, \theta) \\
&= \frac{p(y_i|z_i = k, \theta)p(z_i = k|\theta)}{p(y_i|\theta)} \\
&= \frac{p(y_i|z_i = k, \theta)}{K \cdot p(y_i|\theta)}
\end{aligned}$$

Or on peut écrire le dénominateur sous la forme:

$$p(y_i|\theta) = \sum_{j=1}^K p(y_i|z_i = j, \theta)p(z_i = j) = \sum_{j=1}^K \frac{1}{K} p(y_i|z_i = j, \theta)$$

D'où le résultat:

$$\begin{aligned}
p(z_i = k|y, \theta) &= \frac{p(y_i|z_i = k, \theta)}{\sum_{j=1}^K p(y_i|z_i = j, \theta)} \\
&= \frac{\theta_k^{y_i} e^{-\theta_k}}{\sum_{j=1}^K \theta_j^{y_i} e^{-\theta_j}}
\end{aligned}$$

Dans la génération de cette loi on peut se concentrer sur le nominateur uniquement puisque le dénominateur est désormais une constante de la normalisation des probabilités.

### Question 4

$$\begin{aligned}
p(\theta_k|\theta_{-k}, y, z) &\propto p(y|\theta_k, \theta_{-k}, z)p(\theta_k|z) \\
&\propto p(y|\theta, z)p(\theta_k) \\
&\propto \prod_{i \in I_k} p(y_i|\theta, z) \prod_{i \notin I_k} p(y_i|\theta, z) \frac{1}{\theta_k} \\
&\propto \prod_{i \in I_k} p(y_i|\theta, z) \frac{1}{\theta_k} \\
&\propto \prod_{i \in I_k} \theta_k^{y_i} \exp(-\theta_k)/y_i! \frac{1}{\theta_k} \\
&\propto \prod_{i \in I_k} \theta_k^{y_i} \exp(-\theta_k) \frac{1}{\theta_k} \\
&\propto \theta_k^{n_k \bar{y}_k - 1} \exp(-n_k \theta_k) \\
&\propto \Gamma(n_k \bar{y}_k, 1/n_k)
\end{aligned}$$

## Question 5

```
#init
data <- read.table("/Users/simo/Desktop/data.txt")
y <- c()
for ( i in 1:length(data)){
  for (j in 1:length(data[[i]])){
    y <- c(y, data[[i]][j])
  }
}
n <- length(y)
niter <- 1000
```

```
#Des fonctions utiles
nk.generate = function(k, nit3, z1){
  count <- 0
  for(i in 1:n){
    if(z1[nit3, i] == k){
      count <- count + 1
    }
  }
  return(count)
}

yk.generate = function(k, nit2, z2){
  s <- 0
  l <- 0
  for(i in 1:n){
    if(z2[nit2, i] == k){
      s <- s + y[i]
      l <- l + 1
    }
  }
  if(l > 0){
    return(s/l)
  }
}

p.generator = function(i, nit1, K, theta){
  p <- c()
  for(k in 1:K){
    p <- c(p, theta[nit1, k]^y[i] * exp(-theta[nit1, k])/factorial(y[i]))
  }
  return(p);
}
```

```
#Un cycle de Gibbs
K<-3
theta <- matrix(100, 1, K)
z <- matrix(0, 1, n)
#GS1
for(i in 1:n) {
  p <- p.generator(i, 1, K, theta)
  z[1, i] <- sample(1:K, 1, prob = p)
}
```

```
#GS2
for(k in 1:K) {
  nk <- nk.generate(k, 1, z)
  yk <- yk.generate(k, 1, z)
  theta[1, k] <- rgamma(1, shape = nk*yk, scale = 1/nk)
}
```

Le but de l'algorithme est de simuler  $p(z, \theta | y)$  Pour cela on échantillonne de  $p(z_i = k | y, \theta)$  et de  $p(\theta_k | \theta_{-k}, y, z)$  itérativement

### Question 6

A l'issue d'une exécution de l'algorithme précédent, on peut estimer l'espérance de  $\theta_k$  avec  $\bar{y}_k$

On peut estimer les proportions par  $n_k/n$

### Question 7

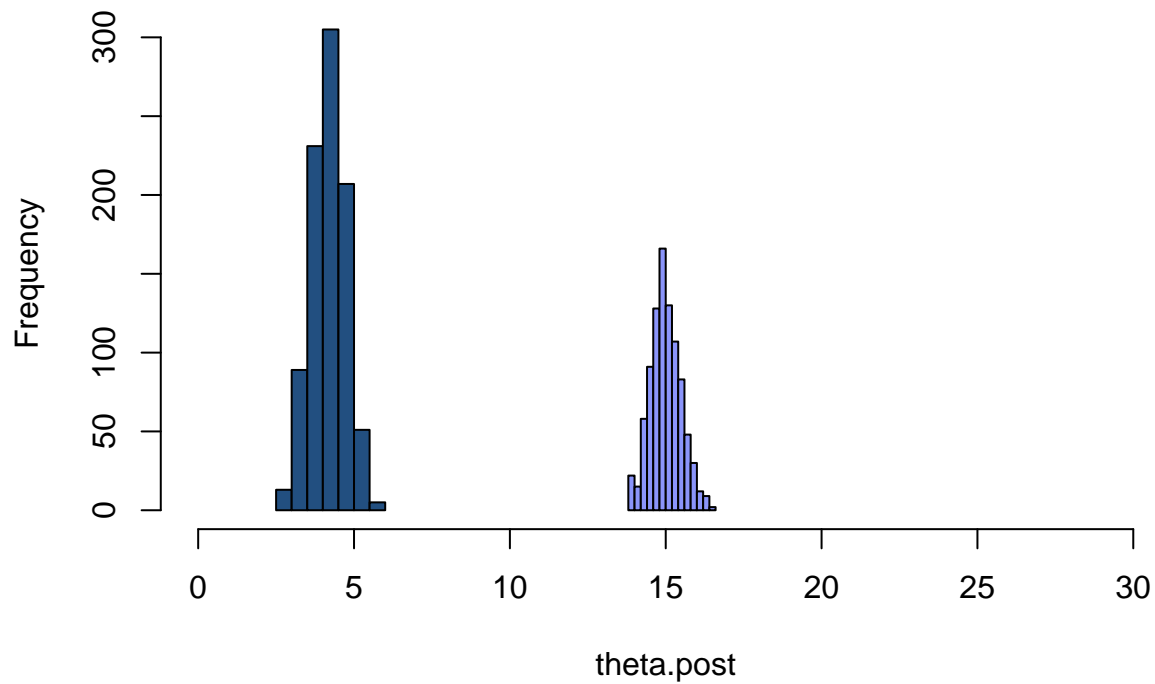
L'algorithme de Gibbs nous permet de simuler  $\theta$  et le vecteur variables cachées  $z$  sachant les observations  $y$

```
#Gibbs sampling function, return the poseterior theta and z
gibbs_sampler = function(K){
  theta <- matrix(100, niter, K)
  z <- matrix(0, niter, n)
  for(nit in 1:niter) {
    for(i in 1:n) {
      p <- p.generator(i, nit, K, theta)
      z[nit, i] <- sample(1:K, 1, prob = p)
    }

    for(k in 1:K) {
      if(nit < niter){
        nk <- nk.generate(k, nit, z)
        if(nk > 0){
          yk <- yk.generate(k, nit, z)
          theta[nit+1, k] <- rgamma(1, shape = nk*yk, scale = 1/nk)}
        if(nk == 0){
          theta[nit+1, k] <- runif(1, 0, 1)
        }
      }
    }
  }
  to_return <- list("theta"=theta, "z"=z)
  return (to_return)
}

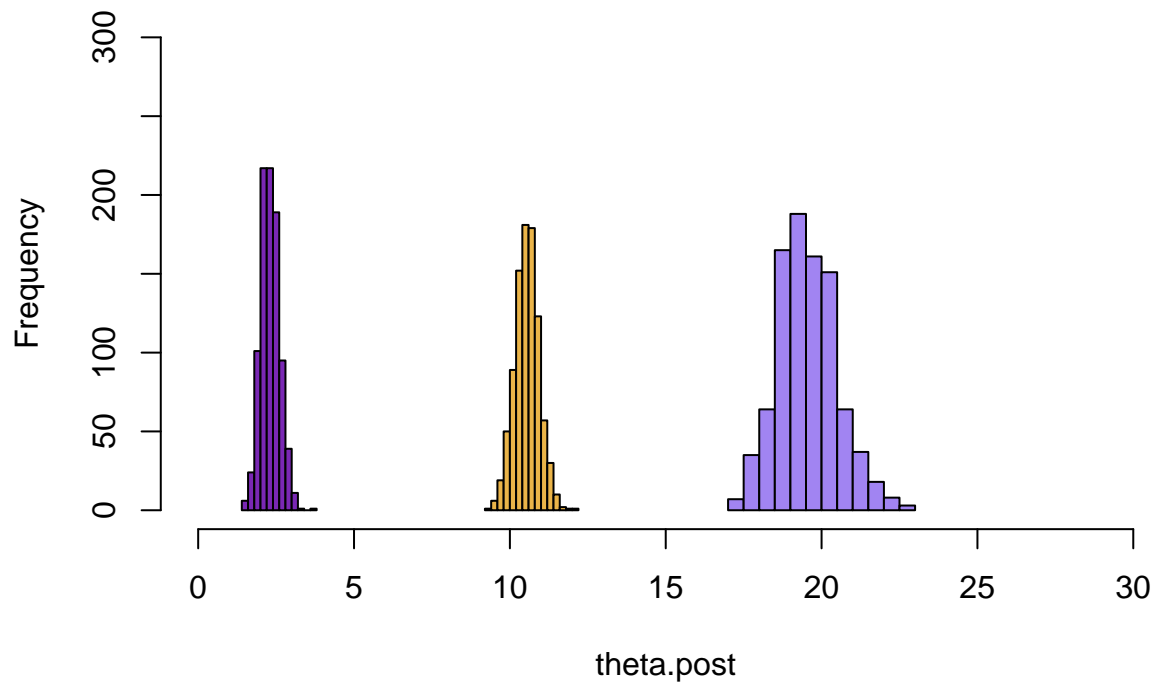
#Pour K =2 i.e 2 sources
postK2 <- gibbs_sampler(K=2)
theta.post.k2 <- postK2$theta
z.post.k2 <- postK2$z
hist(theta.post.k2[100:nrow(theta.post.k2),1],
      col=rgb(runif(1, 0, 1), runif(1, 0, 1), runif(1, 0, 1)), xlim=c(0,30), ylim=c(0,300),
      main = "histogram of theta.post means for K = 2",
      xlab = "theta.post")
hist(theta.post.k2[100:nrow(theta.post.k2),2],
      col=rgb(runif(1, 0, 1), runif(1, 0, 1), runif(1, 0, 1)), add = T)
```

## histogram of theta.post means for K = 2



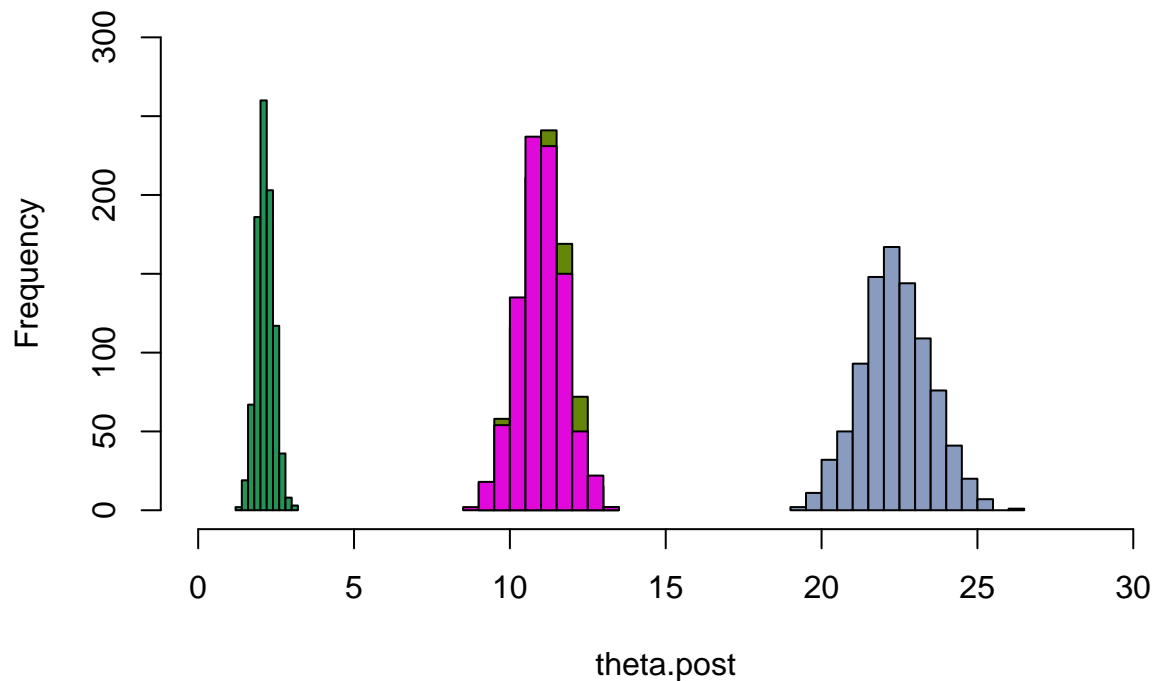
```
#Pour K = 3 i.e 3 sources
postK3 <- gibbs_sampler(K=3)
theta.post.k3 <- postK3$theta
z.post.k3 <- postK3$z
hist(theta.post.k3[100:nrow(theta.post.k3),1],
     col=rgb(runif(1, 0, 1), runif(1, 0, 1), runif(1, 0, 1)), xlim=c(0,30), ylim=c(0,300),
     main = "histogram of theta.post means for K = 3",
     xlab = "theta.post")
hist(theta.post.k3[100:nrow(theta.post.k3),2],
     col=rgb(runif(1, 0, 1), runif(1, 0, 1), runif(1, 0, 1)), add = T)
hist(theta.post.k3[100:nrow(theta.post.k3),3],
     col=rgb(runif(1, 0, 1), runif(1, 0, 1), runif(1, 0, 1)), add = T)
```

### histogram of theta.post means for K = 3



```
#Pour K = 4
postK4 <- gibbs_sampler(K=4)
theta.post.k4 <- postK4$theta
z.post.k4 <- postK4$z
hist(theta.post.k4[100:nrow(theta.post.k4),1],
     col=rgb(runif(1, 0, 1), runif(1, 0, 1), runif(1, 0, 1)), xlim=c(0,30), ylim=c(0,300),
     main = "histogram of theta.post means for K = 4",
     xlab = "theta.post")
hist(theta.post.k4[100:nrow(theta.post.k4),2],
     col=rgb(runif(1, 0, 1), runif(1, 0, 1), runif(1, 0, 1)), add = T)
hist(theta.post.k4[100:nrow(theta.post.k4),3],
     col=rgb(runif(1, 0, 1), runif(1, 0, 1), runif(1, 0, 1)), add = T)
hist(theta.post.k4[100:nrow(theta.post.k4),4],
     col=rgb(runif(1, 0, 1), runif(1, 0, 1), runif(1, 0, 1)), add = T)
```

## histogram of theta.post means for K = 4



En

analysant ces histogrammes on peut déduire que diviser en 3 classes nous permet d'avoir une dispersion des paramètres des sources. Considérer 4 classes par contre on a deux paramètres qui se superposent. On cherche alors plus de détails pour pouvoir conclure sur K

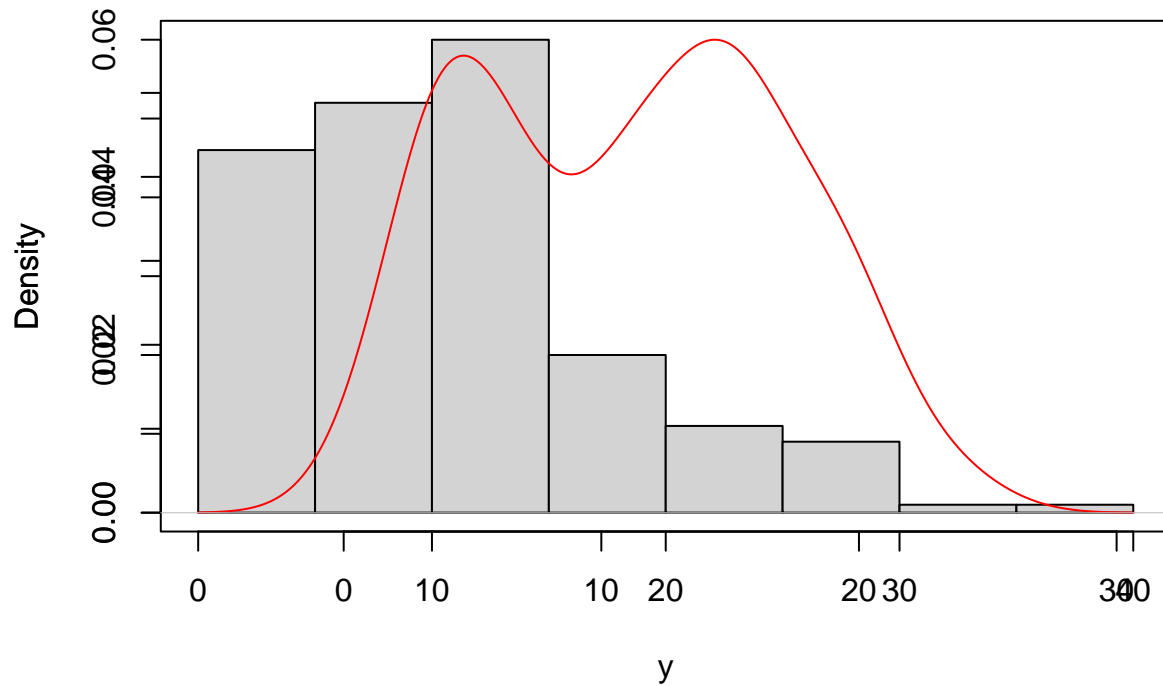
```
#Determination de la prédictive
pred.fct = function(K){
  sample.gibbs <- gibbs_sampler(K)
  theta <- sample.gibbs$theta
  z <- sample.gibbs$z
  zz <- z[niter,]
  tt <- theta[niter,]
  #for (i in 1:n){
  #  zz <- c(zz, mean(z[300:niter,i]))
  #}
  # for (k in 1:K){
  #  tt <- c(tt, mean(theta[300:niter, k]))
  # }
  y.pred <- c()
  for (k in 1:K){
    y.pred <- c(y.pred, rpois(sum(zz == k), tt[k]))
  }
  return (y.pred)
}

y.pred.k2 <- pred.fct(2)
y.pred.k3 <- pred.fct(3)
y.pred.k4 <- pred.fct(4)

hist(y, freq=FALSE, main = "")
par(new=TRUE)
```

```
plot(density(y.pred.k2), main="Histogramme de y et Loi de mélange pour K=2", xlab="",
     col = "red")
```

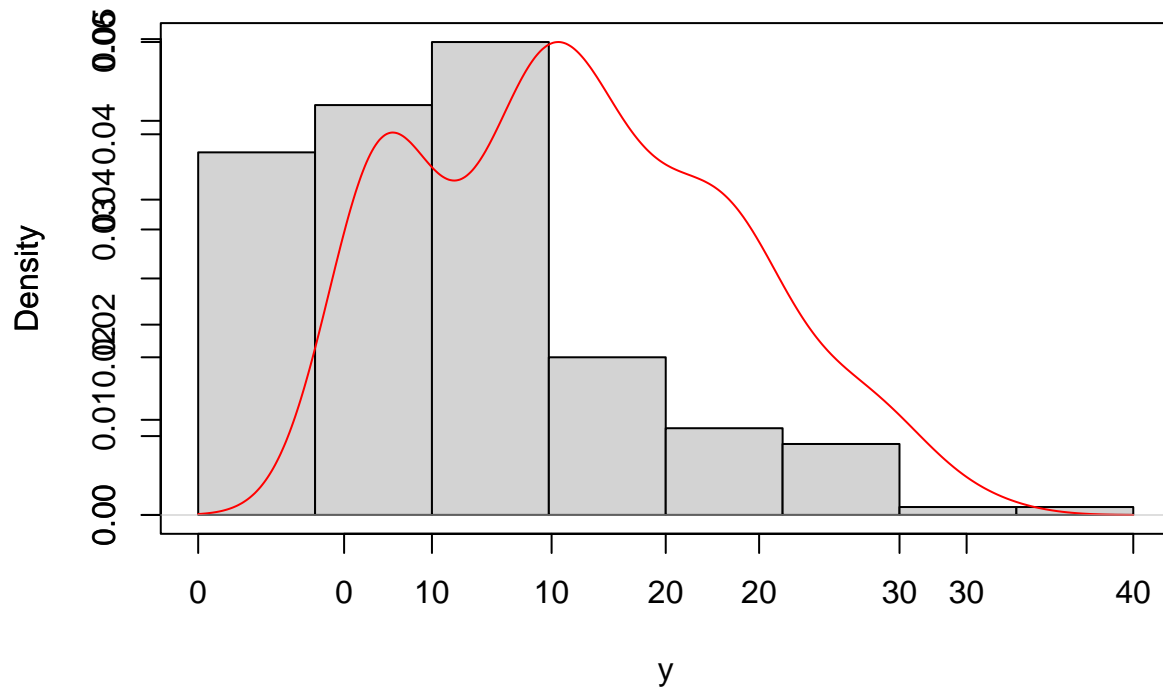
## Histogramme de y et Loi de mélange pour K=2



```
hist(y, freq=FALSE, main = "")
par(new=TRUE)
plot(density(y.pred.k3), main="Histogramme de y et Loi de mélange pour K=3", xlab="",
     col = "red")
```

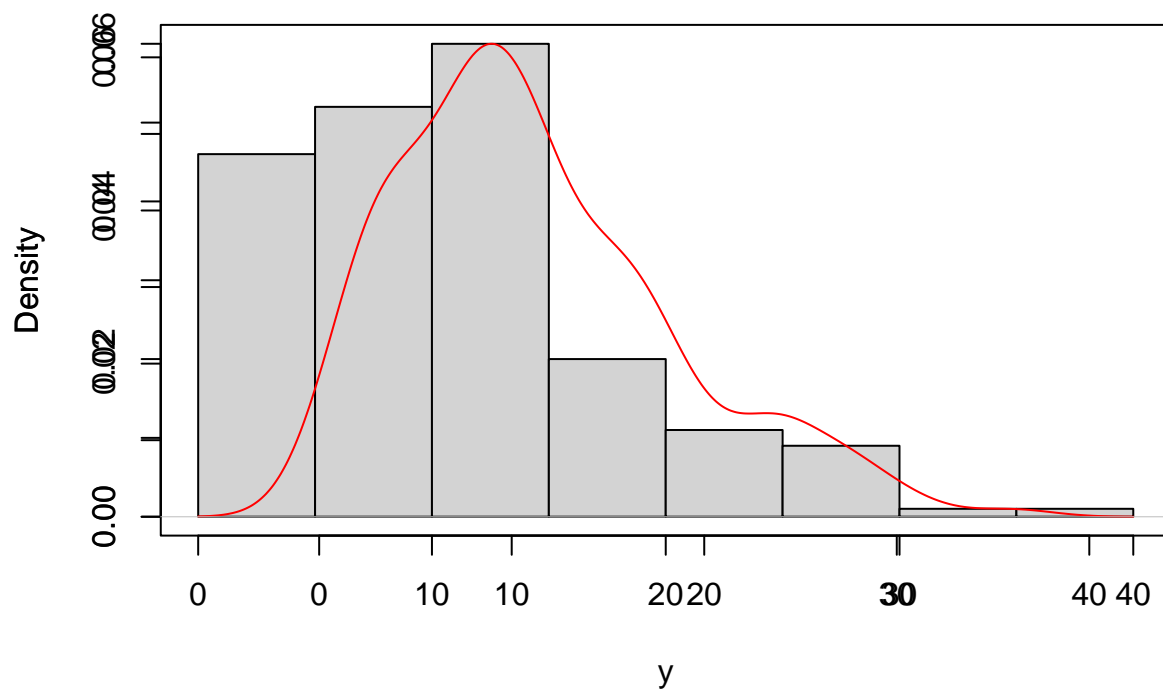


### Histogramme de y et Loi de mélange pour K=3



```
hist(y, freq=FALSE, main = "")
par(new=TRUE)
plot(density(y.pred.k4), main="Histogramme de y et Loi de mélange pour K=4", xlab="",
     col = "red")
```

### Histogramme de y et Loi de mélange pour K=4

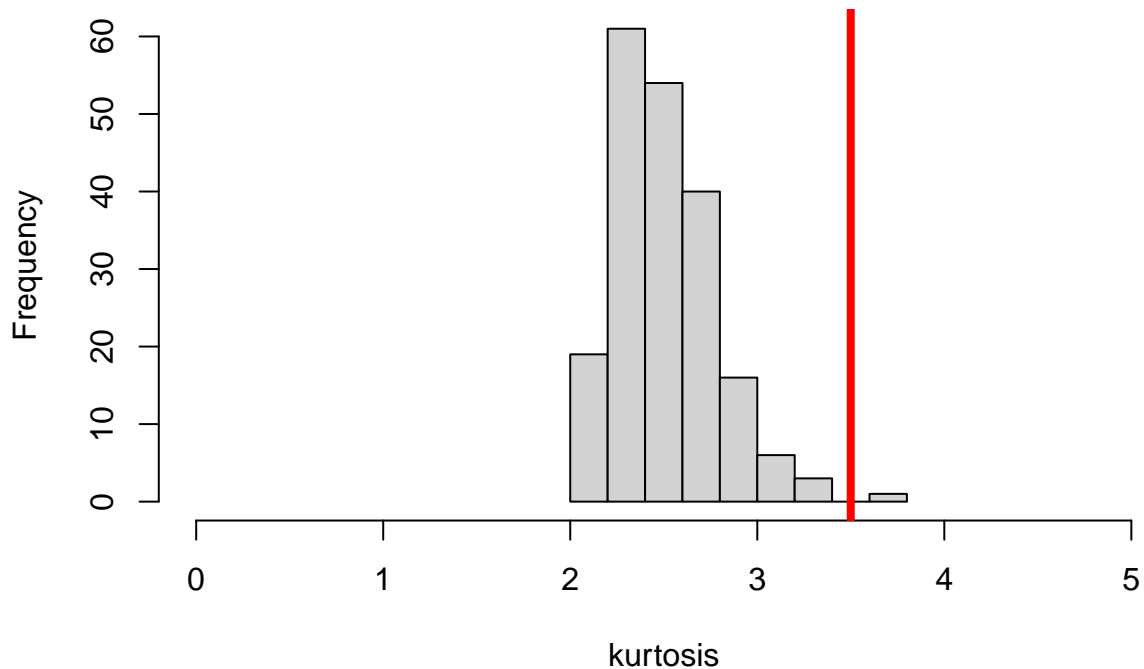


On remarque qu'on a les meilleurs résultats pour K=3 et K=4. On utilise maintenant les statistiques de skewness

et kurtosis sur les lois predictives.

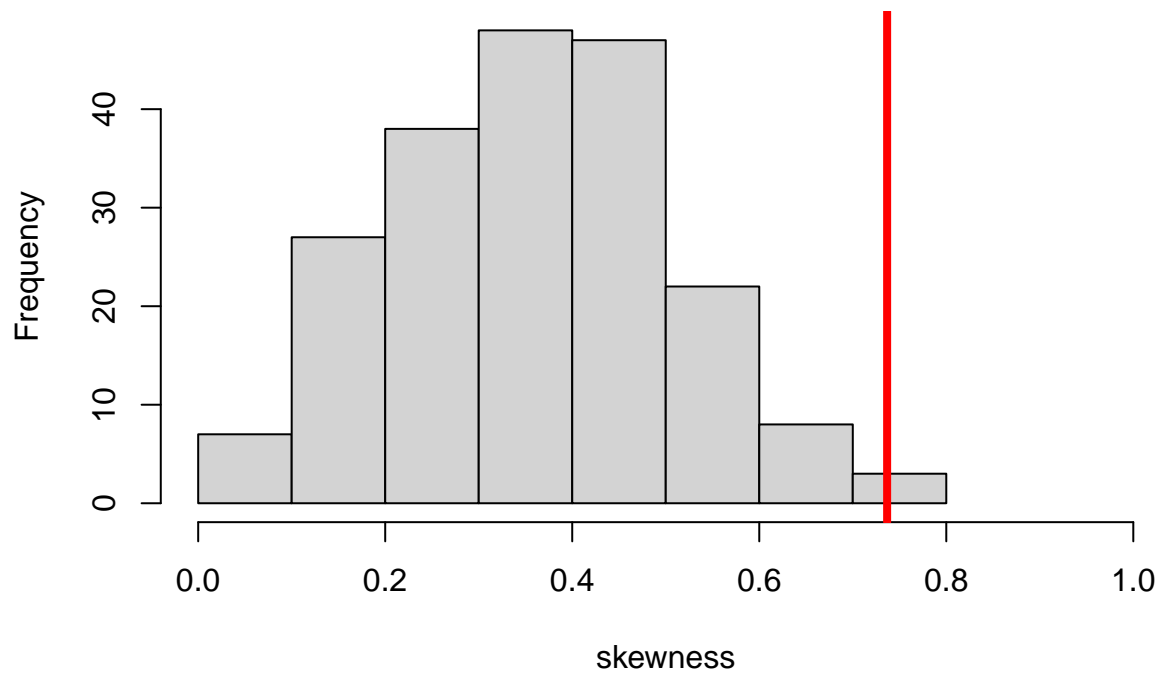
```
#Fonction pour determiner skewness et kurtosis en fonction de K
library(moments)
post.stats.fct = function(K, z, theta){
  post.stat.kurtosis = NULL
  post.stat.skewness = NULL
  for (i in 1:n){
    zz <- z[i+100,]
    tt <- theta[i+100,]
    yy <- c()
    for (k in 1:K){
      yy <- c(yy, rpois(sum(zz == k),tt[k]))
    }
    post.stat.kurtosis <- c(post.stat.kurtosis, kurtosis(yy))
    post.stat.skewness <- c(post.stat.skewness, skewness(yy))
  }
  list_to_return <- list("kurt" = post.stat.kurtosis, "skew" = post.stat.skewness)
  return (list_to_return)
}
#Pour K = 3
stats.post.k3 <- post.stats.fct(K=3, z.post.k3, theta.post.k3)
hist(stats.post.k3$kurt, main="Kurtosis test for K = 3", xlab="kurtosis", xlim=c(0,5))
abline(v=kurtosis(y), lwd=4, col = "red")
```

### Kurtosis test for K = 3



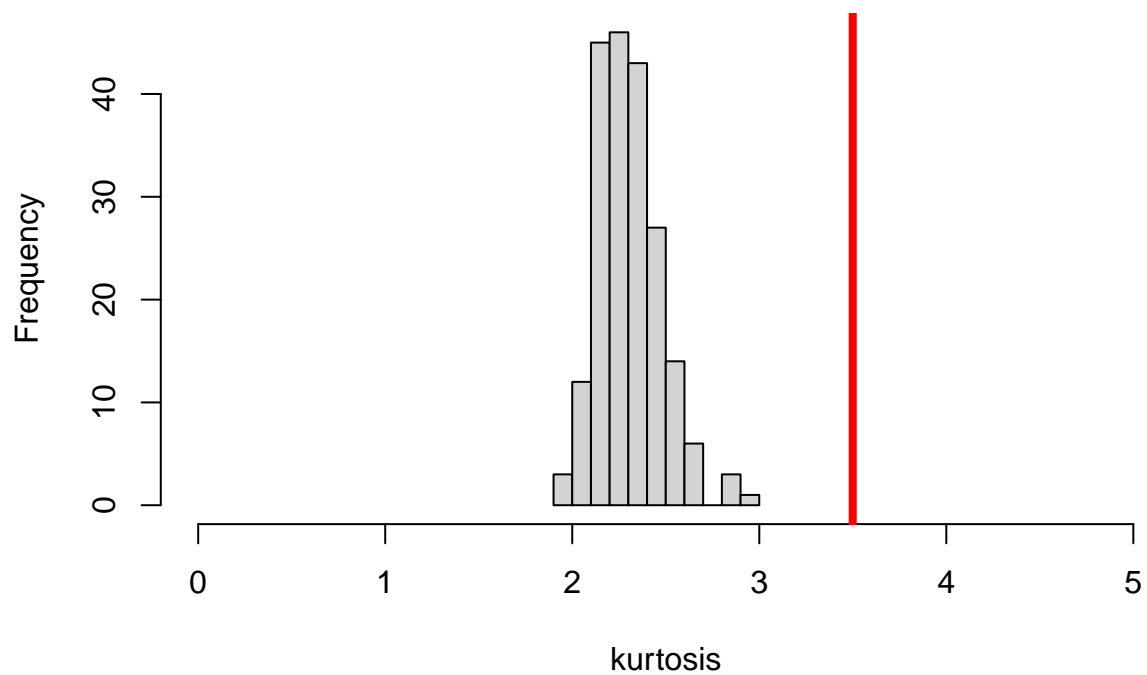
```
hist(stats.post.k3$skew, main="Skewness test for K = 3", xlab="skewness", xlim=c(0,1))
abline(v=skewness(y), lwd=4, col = "red")
```

### Skewness test for K = 3

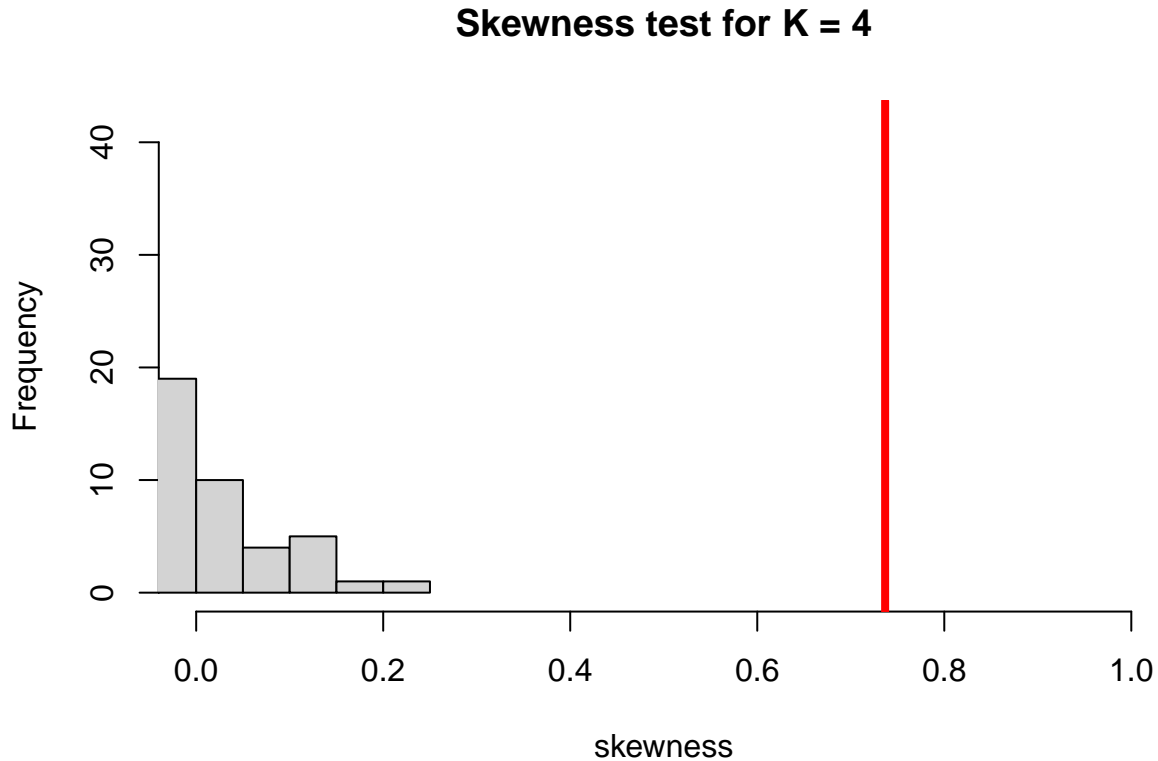


```
#Pour K = 4
stats.post.k4 <- post.stats.fct(K=4, z.post.k4, theta.post.k4)
hist(stats.post.k4$kurt, main="Kurtosis test for K = 4", xlab="kurtosis", xlim=c(0,5))
abline(v=kurtosis(y), lwd=4, col = "red")
```

### Kurtosis test for K = 4



```
hist(stats.post.k4$skew, main="Skewness test for K = 4", xlab="skewness", xlim=c(0,1))
abline(v=skewness(y), lwd=4, col = "red")
```



On trouve alors de bonnes résultats pour  $K = 3$ , mais nous ne pouvons pas non plus écarter complètement le cas de  $K = 4$

Critiques: Le choix de la loi uniforme pour  $z$  est exigent. En effet à priori on ne sait pas si une source émet plus que les autres, on peut résoudre partiellement ce problème en utilisant la loi de Dirichlet. Aussi on a introduit plusieurs paramètres en fonction des données alors que le nombre de données n'est pas très grand, ce qu'on ne veut pas en général.