BNPdensity: Paquete R para estimación de densidades y clasificación

Luis E. Nieto-Barajas

(conjunto con E. Barrios e I. Prüster)

Departamento de Estadística, ITAM, Mexico

v0.5 Encuentro de usuarios de R

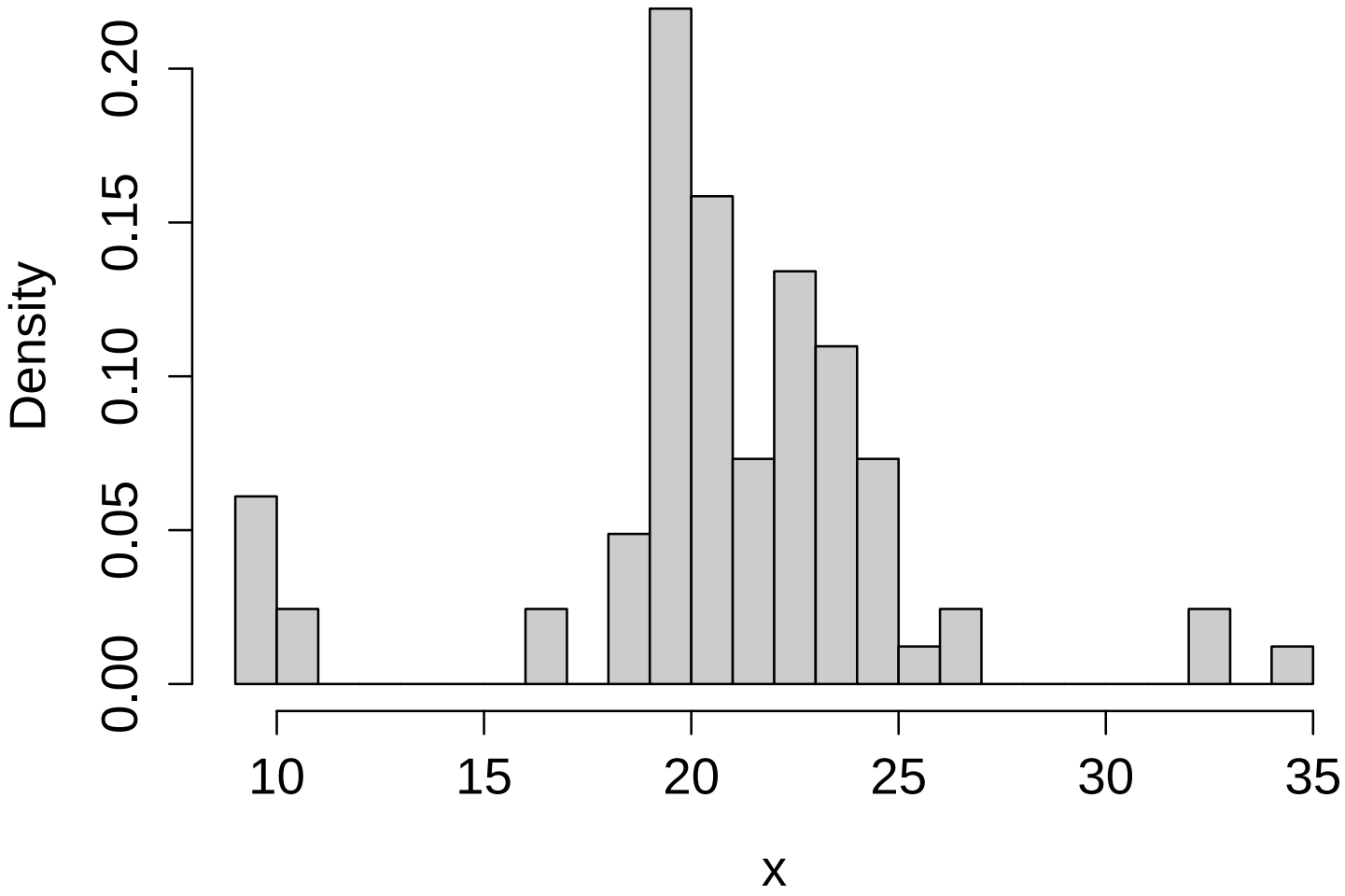
30 de marzo, 2012

Contenido

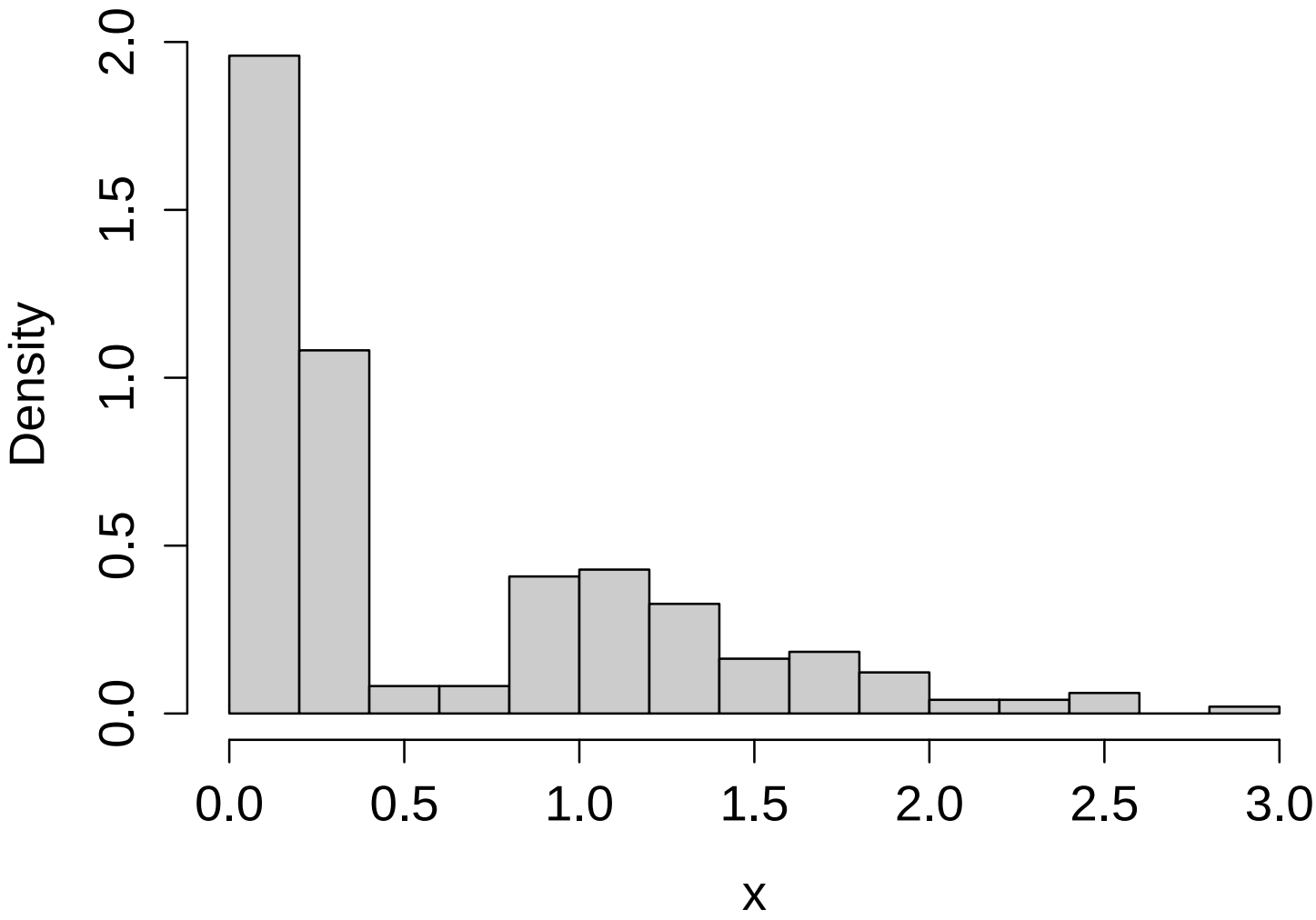
Objetivos

- Clasificación: Recuperar los componentes de una mezcla de subpoblaciones en un conjunto de datos
- Estimación de densidades: Estimar de manera adecuada la función de densidad a partir de un conjunto de datos

Datos de galaxias.



Datos de enzimas.



Introducción

- Estimación de densidades es un problema de naturaleza no paramétrica
- Comúnmente se usan modelos de mezcla

$$f(x) = \int k(x|y)\mathrm{P}(\mathrm{d}y),$$

donde $k(x|y)$ es un kernel de probabilidad paramétrico y P es la distribución de pesos de mezcla

Introducción

- Enfoques:

Introducción

- Enfoques:
  - Clásico: Si tomamos a P como la f.d.e. $\Rightarrow$ popular estimador de kernel (Silverman, 1986)

$$\hat{f}(x) = \sum_{i=1}^{n} \frac{1}{n} k(x|X_i, \sigma)$$

Introducción

- Enfoques:
  - Clásico: Si tomamos a P como la f.d.e. $\Rightarrow$ popular estimador de kernel (Silverman, 1986)

$$\hat{f}(x) = \sum_{i=1}^{n} \frac{1}{n} k(x|X_i, \sigma)$$

  - Bayesiano: Asignar a P una distribución inicial (NP) $\Rightarrow$ $f(x)$ es una función de densidad aleatoria y el estimador Bayesiano es

$$\hat{f}(x) = \mathrm{E}\left\{ \int k(x|y)\mathrm{P}(\mathrm{d}y) \middle| X_1, \ldots, X_n \right\}$$

Introducción

- Enfoques:
  - Clásico: Si tomamos a P como la f.d.e. $\Rightarrow$ popular estimador de kernel (Silverman, 1986)

$$\hat{f}(x) = \sum_{i=1}^{n} \frac{1}{n} k(x|X_i, \sigma)$$

  - Bayesiano: Asignar a P una distribución inicial (NP) $\Rightarrow$ $f(x)$ es una función de densidad aleatoria y el estimador Bayesiano es

$$\hat{f}(x) = \mathrm{E}\left\{ \int k(x|y)\mathrm{P}(\mathrm{d}y) \middle| X_1, \ldots, X_n \right\}$$

- Usaremos el enfoque Bayesiano con $\mathrm{P} \sim \mathscr{P}$, donde $\mathscr{P}$ denota la ley de un proceso estocástico que asigna probabilidad uno a las medidas de probabilidad discretas

Estimación Bayesiana

- Representación jerárquica del modelo Bayesiano

$$\begin{aligned} X_i \,|\, Y_i, \phi &\overset{\text{ind}}{\sim} k(\,\cdot\,|\, Y_i, \phi) \\ Y_i | \mathrm{P} &\overset{\text{iid}}{\sim} \mathrm{P}, \\ \mathrm{P} &\sim \mathscr{P} \\ \phi &\sim \pi(\phi) \end{aligned}$$

donde

- $\mathscr{P}$ denota la ley de una medida aleatoria normalizada
- $\mathrm{P}(\cdot) = A(\cdot)/A(\infty)$ con A un proceso aditivo creciente con intensidad de Lévy $\nu(\mathrm{d}s, \mathrm{d}v)$
- Tomaremos $\mathbb{X} \subseteq \mathbb{R}$ y $\mathbb{Y} \subseteq \mathbb{R}^m$

Casos particulares

- Si consideramos $k(\cdot|\mu, \sigma)$ parametrizado en términos de media μ y desviación estándar σ

Casos particulares

- Si consideramos $k(\cdot|\mu,\sigma)$ parametrizado en términos de media μ y desviación estándar σ
- Mezcla Tipo 1 (MixNRMI1)

$$
\begin{aligned}
X_i \,|\, Y_i, \phi &\overset{\text{ind}}{\sim} k(\,\cdot\,|\, Y_i, \phi) \\
Y_i|\mathrm{P} &\overset{\text{iid}}{\sim} \mathrm{P}, \\
\mathrm{P} &\sim \mathscr{P} \\
\phi &\sim \pi(\phi)
\end{aligned}
$$

Casos particulares

- Si consideramos $k(\cdot|\mu,\sigma)$ parametrizado en términos de media μ y desviación estándar σ
- Mezcla Tipo 1 (MixNRMI1)

$$\begin{aligned} X_i\,|\,Y_i,\phi &\overset{\text{ind}}{\sim} k(\,\cdot\,|\,Y_i,\phi) \\ Y_i|\mathrm{P} &\overset{\text{iid}}{\sim} \mathrm{P}, \\ \mathrm{P} &\sim \mathscr{P} \\ \phi &\sim \pi(\phi) \end{aligned}$$

- Mezcla Tipo 2 (MixNRMI2)

$$\begin{aligned} X_i\,|\,Y_i,Z_i &\overset{\text{ind}}{\sim} k(\,\cdot\,|\,Y_i,Z_i) \\ (Y_i,Z_i)|\mathrm{P} &\overset{\text{iid}}{\sim} \mathrm{P}, \\ \mathrm{P} &\sim \mathscr{P} \end{aligned} \tag{1}$$

Opciones de kernels

- Kernel normal:

$$k(x|\mu,\sigma) = \frac{1}{\sqrt{2\pi}b}\exp\left\{-\frac{1}{2b^2}(x-a)^2\right\}\mathbb{I}_{\mathbb{R}}(x), \quad a=\mu,\ b=\sigma$$

- Kernel doble exponencial:

$$k(x|\mu,\sigma) = \frac{1}{2b}\exp\left\{-\frac{1}{b}|x-a|\right\}\mathbb{I}_{\mathbb{R}}(x), \quad a=\mu,\ b=\sigma/\sqrt{2}$$

- kernel gamma:

$$k(x|\mu,\sigma) = \frac{b^a}{\Gamma(a)}x^{a-1}e^{-bx}\mathbb{I}_{\mathbb{R}^+}(x), \quad a=\mu^2/\sigma^2,\ b=\mu/\sigma^2$$

- kernel log-normal:

$$k(x|\mu,\sigma) = \frac{1}{x\sqrt{2\pi}b}\exp\left\{-\frac{1}{2b^2}(\log x-a)^2\right\}\mathbb{I}_{\mathbb{R}^+}(x),$$

con $a = \log\left(\frac{\mu}{\sqrt{1+\sigma^2/\mu^2}}\right)$ y $b = \sqrt{\log\left(1+\frac{\sigma^2}{\mu^2}\right)}$

Medidas de centralidad $E(P) = P_0$

- Normal:

$$p_0(x|\mu,\sigma) = \frac{1}{\sqrt{2\pi}b}\exp\left\{-\frac{1}{2b^2}(x-a)^2\right\}\mathbb{I}_{\mathbb{R}}(x),$$

con $a = \mu$ y $b = \sigma$.

- Gamma:

$$p_0(x|\mu,\sigma) = \frac{b^a}{\Gamma(a)}x^{a-1}e^{-bx}\mathbb{I}_{\mathbb{R}^+}(x),$$

con $a = \mu^2/\sigma^2$ y $b = \mu/\sigma^2$.

Algoritmo general

1. Simular la latente $U|\mathbf{Y}$: generar una propuesta $U^* \sim \text{Ga}(\delta, \delta/U^{[t]})$ con $\delta = 2$, y tomar $U^{[t+1]} = U^*$ con prob. $q_1(U^*, U^{[t]})$, e.o.c. tomar $U^{[t+1]} = U^{[t]}$.
2. Simular trayectorias de la parte del proceso sin puntos fijos de discontinuidad A^*: generar $\vartheta_i \sim \text{Ga}(1,1)$ y encontrar $J_i^{[t+1]}$ resolviendo numericamente la ecuación $\sum_{j=1}^{i} \vartheta_j = M(J_i)$; generar $\tau_i^{[t+1]}$ de P_0. Parar de simular cuando $J_{\ell+1}/\sum_{i=1}^{\ell} J_i < \epsilon$, digamos $\epsilon = 0.0001$.
3. Remuestrear los valores únicos $\{Y_j^*\}$: obtener los valores únicos $Y_j^{*[t]}$ de $\{Y_1^{[t]}, \ldots, Y_n^{[t]}\}$ y sus frecuencias $n_j^{[t]}$. Si $m = 2$ con k parametrizado en términos de media y desviación estándar, generar una propuesta $(Y_j^{*(1)})' \sim k(\bar{X}_j, (Y_j^{*(2)})^{[t]}/\sqrt{n_j^{[t]}})$ y tomar $(Y_j^{*(1)})^{[t+1]}$ igual a $(Y_j^{*(1)})'$ con probabilidad $q_2((Y_j^{*(1)})', (Y_j^{*(1)})^{[t]})$ e.o.c. tomar $(Y_j^{*(1)})^{[t+1]} = (Y_j^{*(1)})^{[t]}$. Similarmente, generar una propuesta $(Y_j^{*(2)})' \sim \text{Ga}(\delta, \delta/Y_j^{*(2)})$ con $\delta = 3$ y tomar $(Y_j^{*(2)})^{[t+1]} = (Y_j^{*(2)})'$ con prob. $q_3((Y_j^{*(2)})', Y_j^{*(2)})^{[t]})$ e.o.c. tomar $(Y_j^{*(2)})^{[t+1]} = (Y_j^{*(2)})^{[t]}$.
4. Simular los saltos fijos del proceso, $\{J_j^*\}$: renombrar los r distintos elementos en $\mathbf{Y}$ as $Y_j^{*[t+1]}$ y registrar se fecuancia, digamos $n_j^{[t+1]}$, $j = 1, \ldots, r$. Generar los saltos $J_j^{*[t+1]} \sim \text{Ga}(n_j^{[t+1]} - \gamma, \kappa + u^{[t+1]})$.
5. Simular el vetor latente $\mathbf{Y}$: para cada $i = 1, \ldots, n$, generar $Y_i^{[t+1]}$ de su distribución discreta evaluando el kernel $k(X_i|\cdot, \phi^{[t]})$ en las distintas localizaciones $\bar{\tau}_i$'s.
6. Simular ϕ: si $\phi \neq \emptyset$ y una dist. inicial para ϕ es asignada, generar una propuesta $\phi' \sim \text{Ga}(\delta, \delta/\phi^{[t]})$ con $\delta = 3$ y tomar $\phi^{[t+1]} = \phi'$ con prob. $q_4(\phi', \phi^{[t]})$, y e.o.c. tomar $\phi^{[t+1]} = \phi^{[t]}$.
7. Evaluar una trayectoria de la densidad aleatoria $f(x|\bar{A}^{[t+1]}, \phi^{[t+1]})$.

Comp2

```
comp2 <- function (y, z)
{
    if (length(y) != length(z))
        stop("vectors y and z should have equal length!")
    n <- length(y)
    matY <- outer(y, y, "==")
    matZ <- outer(z, z, "==")
    mat <- matY & matZ
    jstar <- led <- rep(FALSE, n)
    for (j in seq(n)) {
        if (!led[j]) {
            jstar[j] <- TRUE
            if (j == n)
                break
            ji <- seq(j + 1, n)
            tt <- mat[ji, j] %in% TRUE
            led[ji] <- led[ji] | tt
        }
        if (all(led[-seq(j)]))
            break
    }
    ystar <- y[jstar]
    zstar <- z[jstar]
    nstar <- apply(as.matrix(mat[, jstar]), 2, sum)
    rstar <- length(nstar)
    idx <- match(y, ystar)
    return(list(ystar = ystar, zstar = zstar, nstar = nstar,
        rstar = rstar, idx = idx))
}
```

BNPdensity

- Paquete BNPdensity

BNPdensity

- Paquete BNPdensity
- Comandos

BNPdensity

- Paquete BNPdensity
- Comandos
  - MixNRMI1

```
MixNRMI1(x, probs = c(0.025, 0.5, 0.975), Alpha = 1,
Beta = 0, Gama = 0.4, distr.k = 1, distr.p0 = 1, mu.p0
= mean(x), sigma.p0 = 1.5 * sd(x), asigma = 0.1, bsigma
= 0.1, delta = 3, Delta = 2, Nm = 50, Nx = 100, Nit =
1000, Pbi = 0.1, epsilon = NULL, printtime = TRUE)
```

BNPdensity

- Paquete BNPdensity
- Comandos
  - MixNRMI1

```
MixNRMI1(x, probs = c(0.025, 0.5, 0.975), Alpha = 1,
Beta = 0, Gama = 0.4, distr.k = 1, distr.p0 = 1, mu.p0
= mean(x), sigma.p0 = 1.5 * sd(x), asigma = 0.1, bsigma
= 0.1, delta = 3, Delta = 2, Nm = 50, Nx = 100, Nit =
1000, Pbi = 0.1, epsilon = NULL, printtime = TRUE)
```

  - MixNRMI2

```
MixNRMI2(x, probs = c(0.025, 0.5, 0.975), Alpha = 1,
Beta = 0, Gama = 0.4, distr.k = 1, distr.py0 = 1,
mu.py0 = mean(x), sigma.py0 = 1.5 * sd(x), distr.pz0 =
2, mu.pz0 = 0.1, sigma.pz0 = 0.1, delta = 3, Delta = 2,
Nm = 50, Nx = 100, Nit = 1000, Pbi = 0.1, epsilon =
NULL, printtime = TRUE)
```

BNPdensity

- Salida
  - `xx`: Rejilla de evaluación de la densidad
  - `qx`: Estimadores de la densidad: media y cuantiles
  - `cpo`: medida de bondad de ajuste para cada dato
  - `R`: Número de componente de mezcla

Ejemplo 1

- Cargar paquete

```
library(BNPdensity)
```

- Fijar semilla

```
set.seed(123456)
```

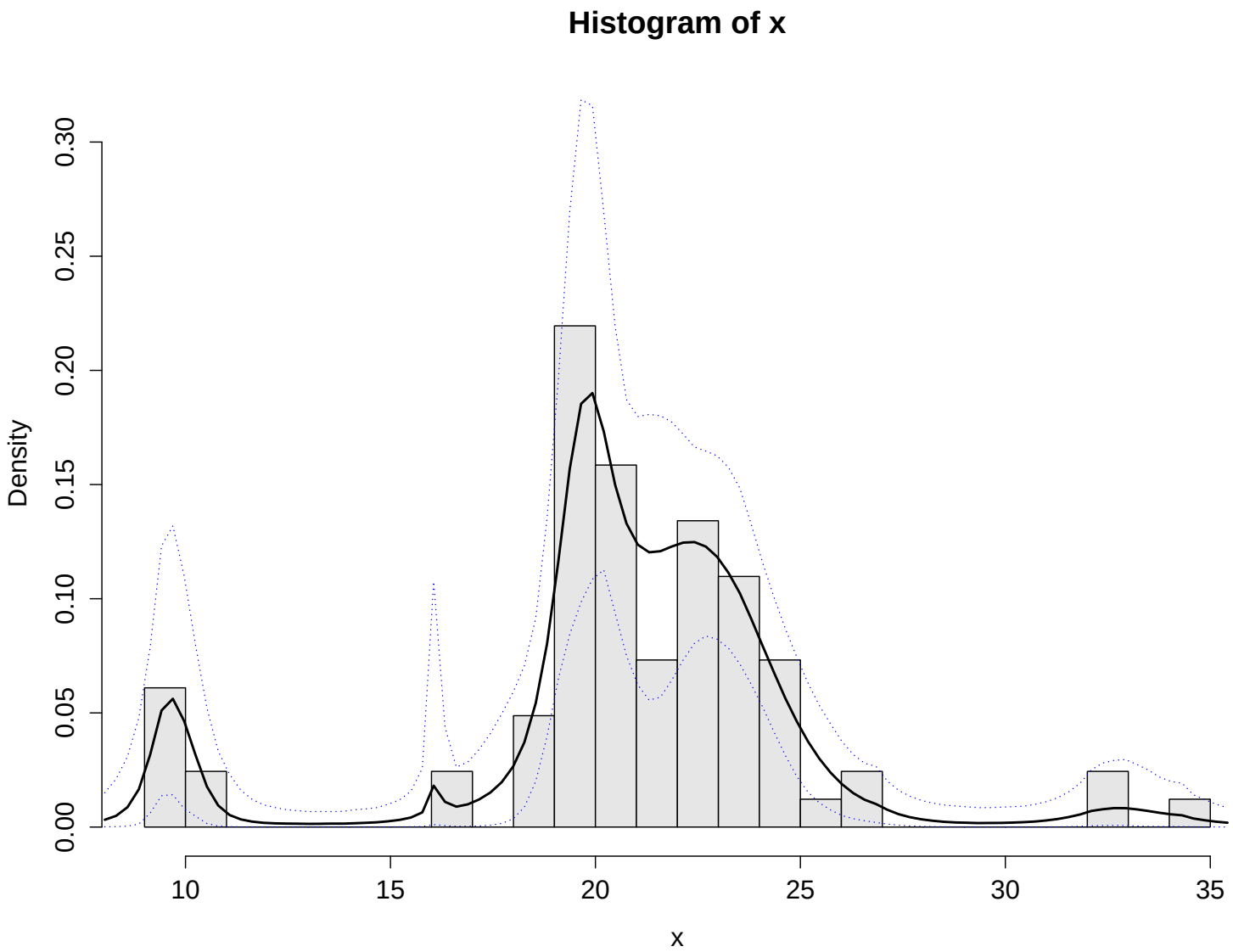
- Cargar datos

```
data(galaxy)
x<-galaxy
```

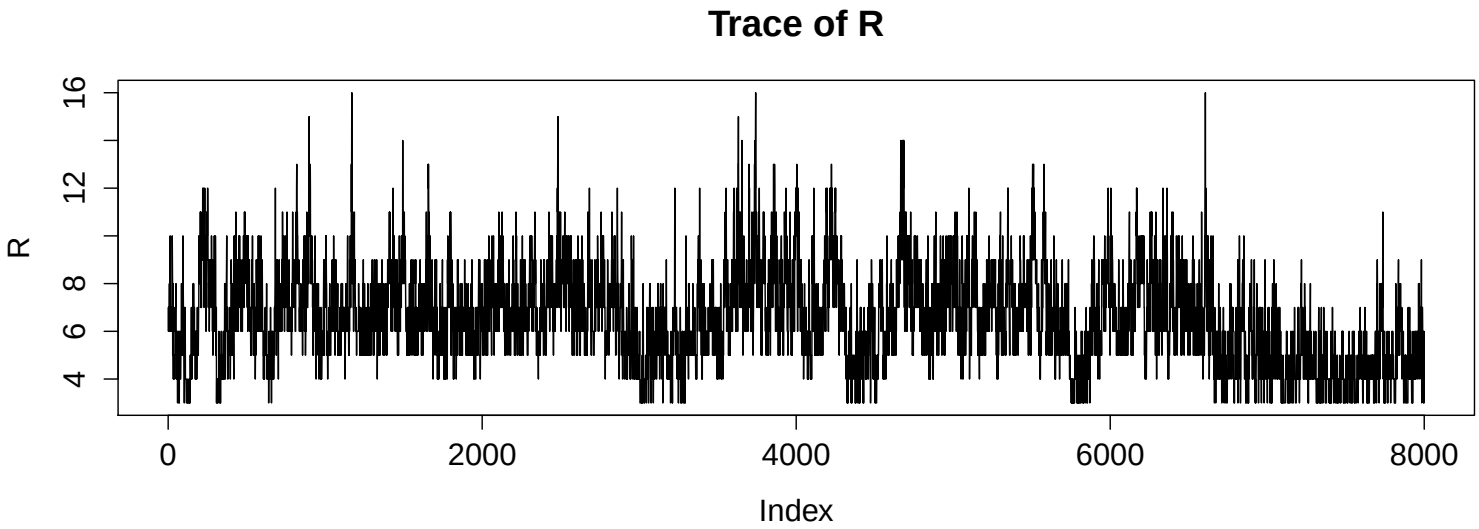
- Ajuste del modelo (bajo especificaciones modificadas)

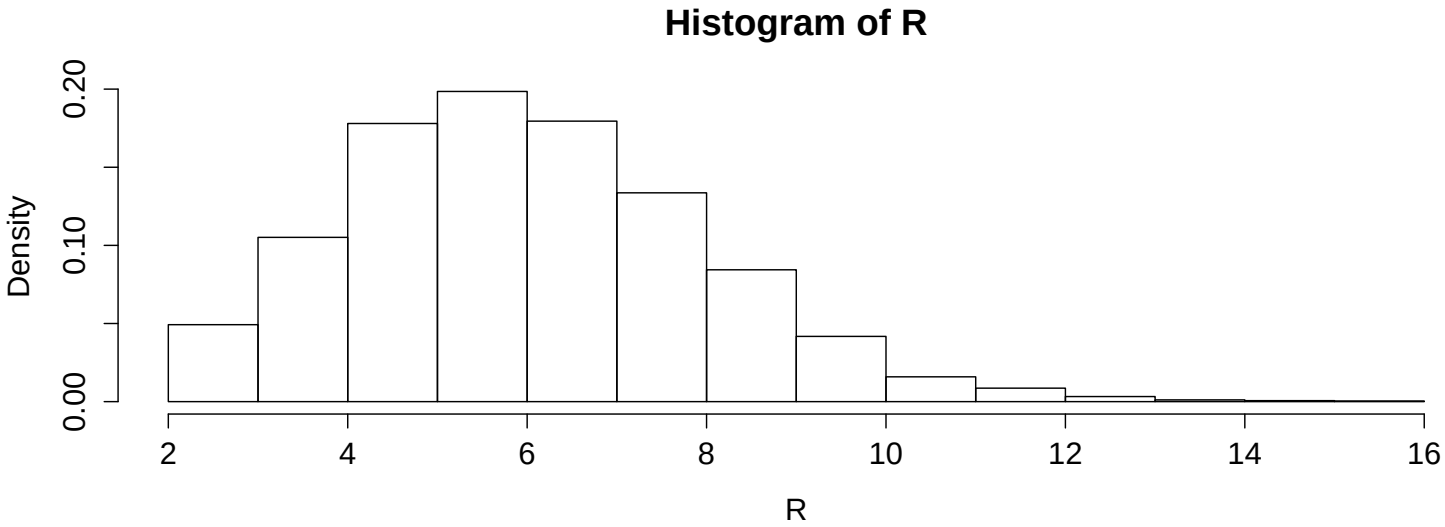
```
Galaxy2.out <- MixNRMI2(x, Alpha = 1, Beta = 0.015, Gama =
0.5, distr.k = 1, distr.py0 = 2, mu.py0 = 20, sigma.py0 =
20, distr.pz0 = 2, mu.pz0 = 1, sigma.pz0 = 1, Nit = 5000,
Pbi = 0.2)
```

Datos de galaxias (Estimador de densidad)

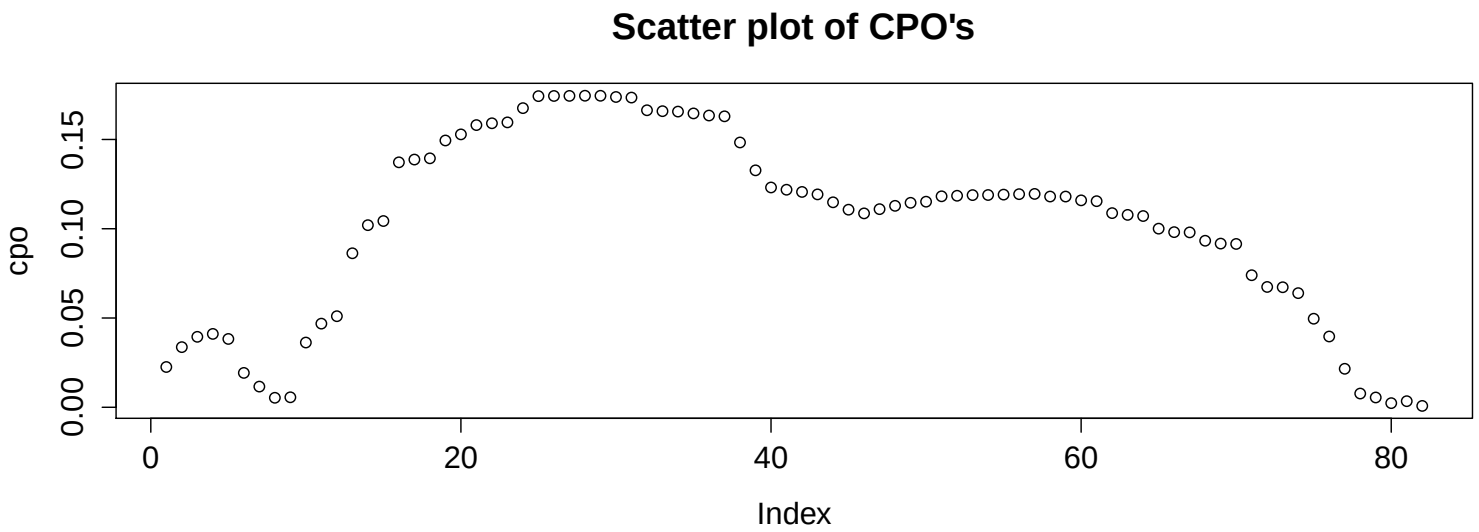


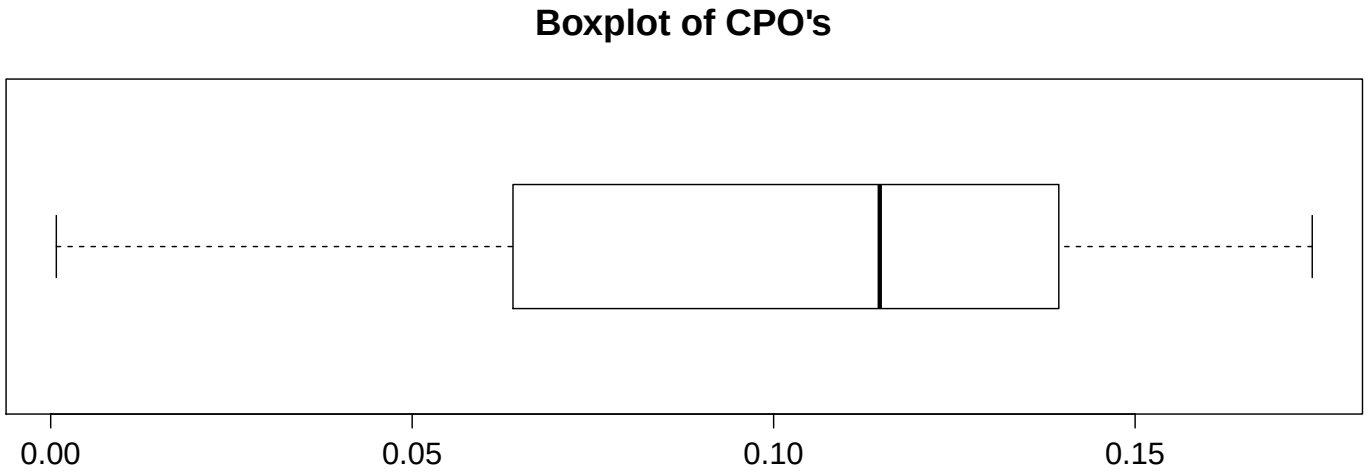
Datos de galaxias (Número de componentes)





Datos de galaxias (Bondad de ajuste)





Ejemplo 2

- Fijar semilla

```
set.seed(123456)
```

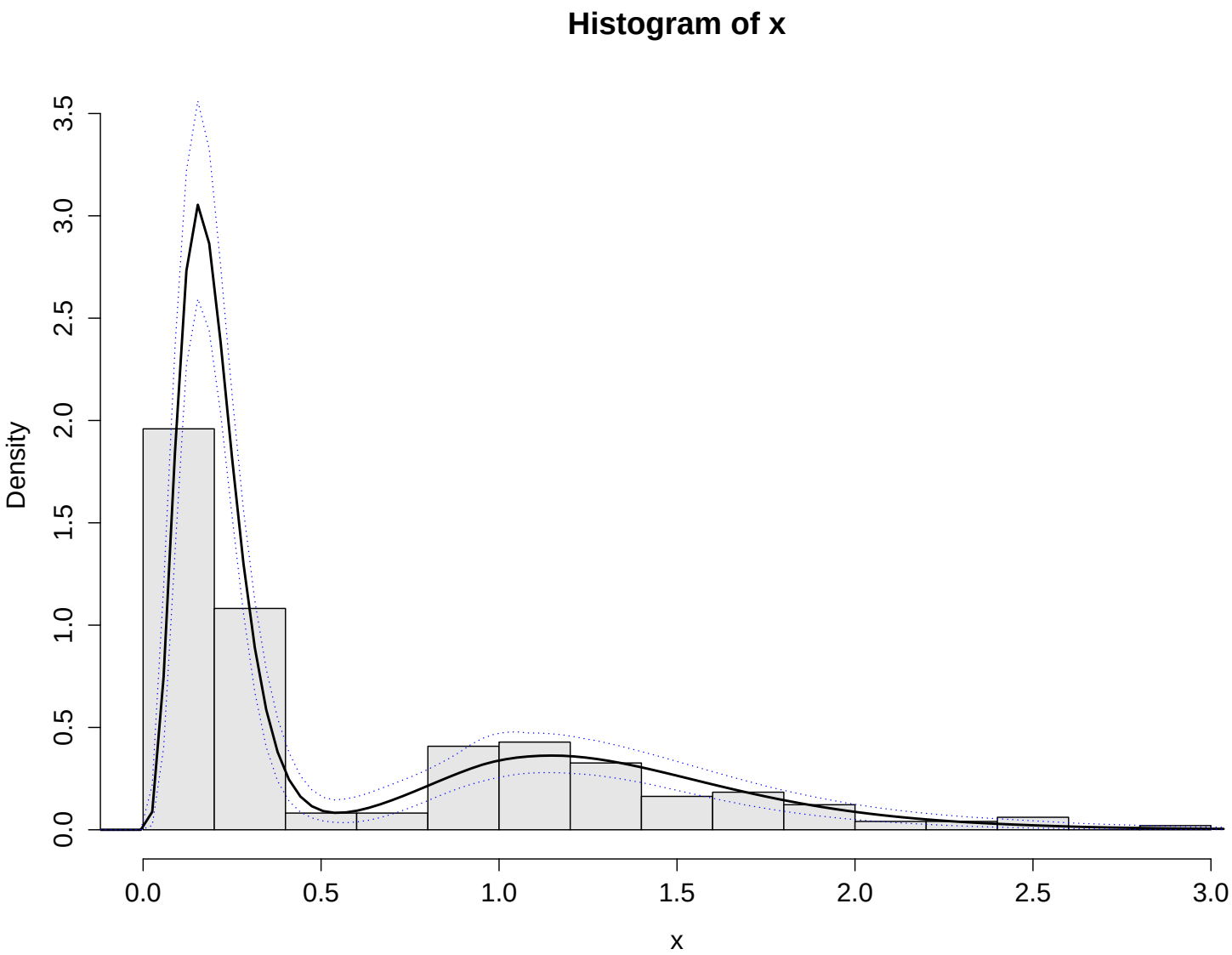
- Cargar datos

```
data(enzyme)
x<-enzyme
```

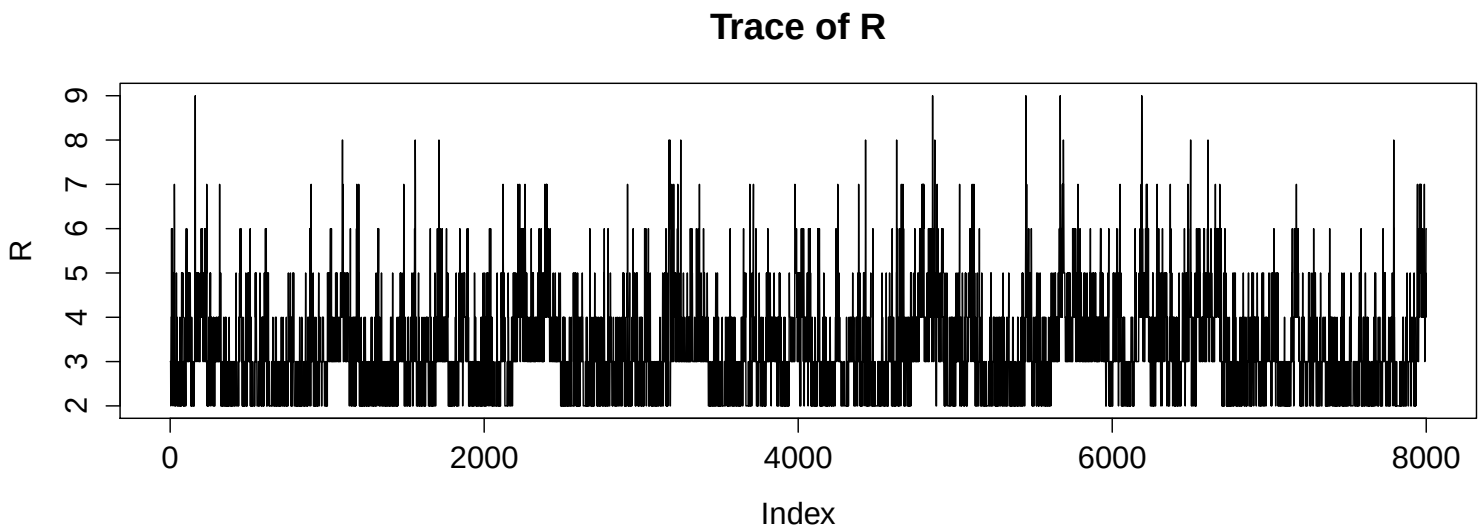
- Ajuste del modelo (bajo especificaciones modificadas)

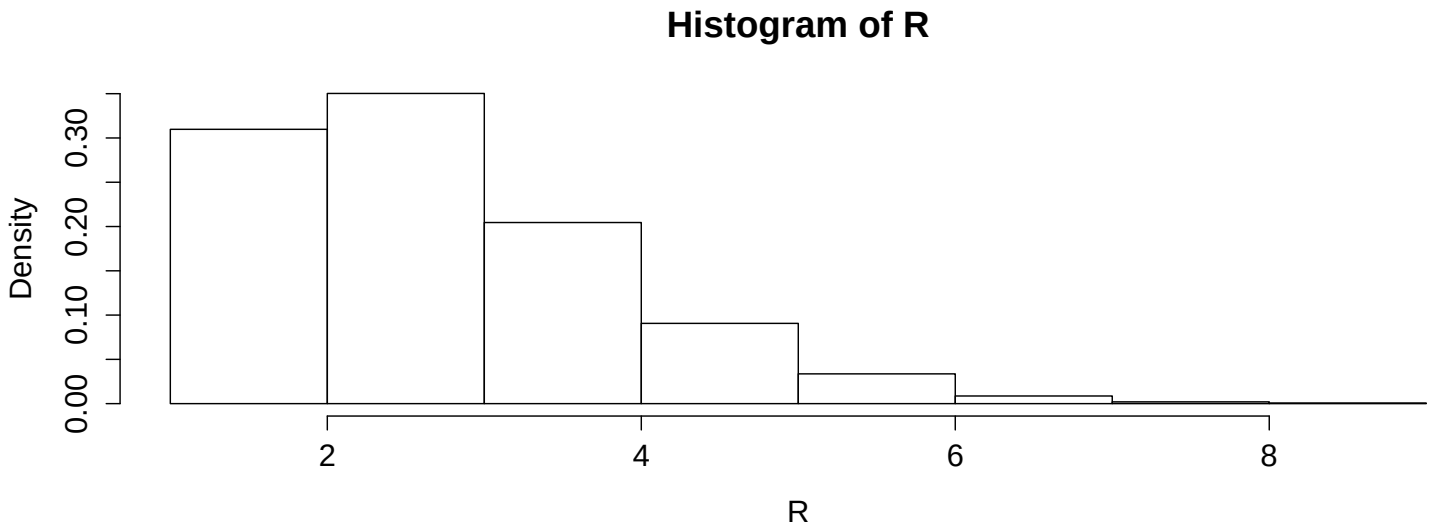
```
Enzyme2.out <- MixNRMI2(x, Alpha = 1, Beta = 0.007, Gama =
0.5, distr.k = 2, distr.py0 = 2, mu.py0 = 10, sigma.py0 =
10, distr.pz0 = 2, mu.pz0 = 1, sigma.pz0 = 1, Nit = 5000,
Pbi = 0.2)
```

Datos de enzimas (Estimador de densidad)

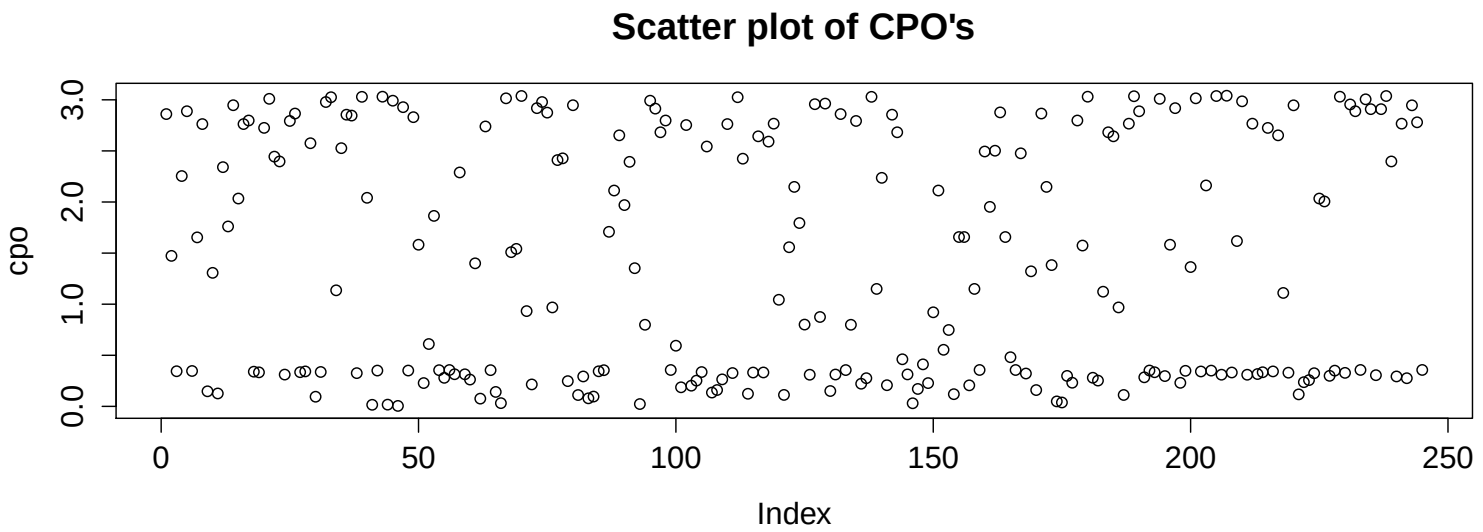


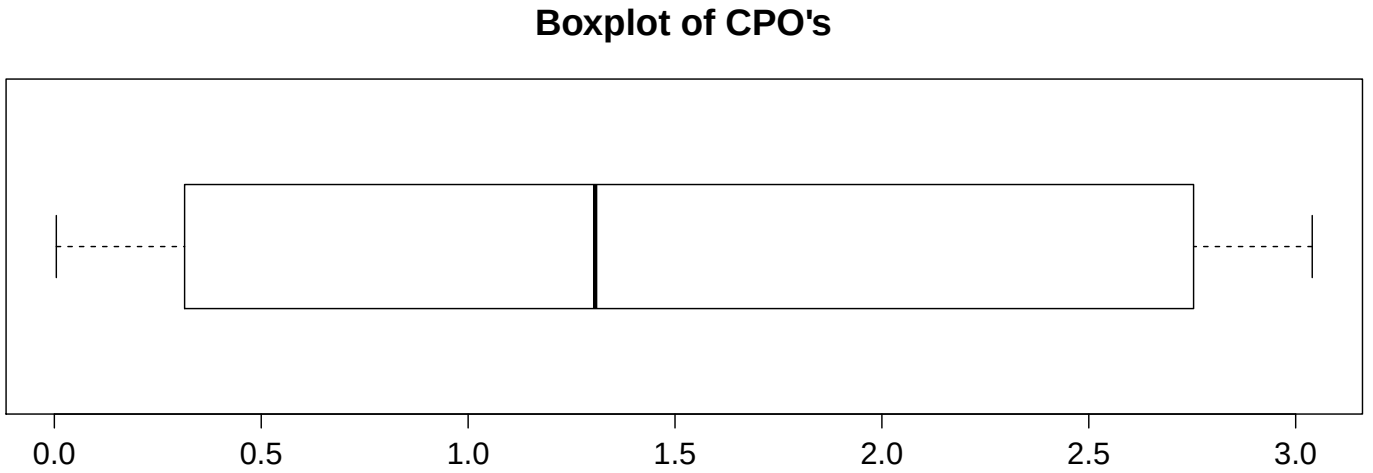
Datos de enzimas (Número de componentes)





Datos de enzimas (Bondad de ajuste)





Ejemplo 3

- Cargar datos

```
data(acidity)

x<-acidity
```

- Ajuste del modelo (bajo especificaciones por defecto)

```
out<-MixNRMI1(x)
```

- Graficación de los estimadores

```
attach(out)
m<-ncol(qx)
ymax <- max(qx[,m])
par(mfrow=c(1,1))
hist(x,probability=TRUE,breaks=20,col=grey(.9),ylim=c(0,ymax))
lines(xx,qx[,1],lwd=2)
lines(xx,qx[,2],lty=3,col=4)
lines(xx,qx[,m],lty=3,col=4)

detach()
```

Datos de grado de acidez en lagos