République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université des Sciences et de la Technologie Houari Boumediene

Faculté de Mathématiques

Département de Probabilité/Statistiques

TP

Estimation Non Parametrique

Etudiante:
BELHADDAD Maguedouda Samia
Date : Mars 2023

1 . Estimation Non Parametrique des la distribution Jointe de deux variables aleatoires continues :

Dans le cas de l'estimation de la densité conjointe de deux variables, X et Y, un estimateur de densité à noyau bivarié peut être utilisé.

Cela implique l'utilisation d'une fonction noyau bidimensionnelle pour lisser les données observées. La largeur de bande de la fonction noyau détermine le degré de lissage .

L'estimateur de densité à noyau de la densité conjointe de x et y peut être approximé par une somme de fonctions à noyau centrées sur chaque point de données (x_i, y_i).

L'estimateur de densité de noyau de la PDF conjointe peut être écrit comme suit :

$$f(x,y) = \frac{1}{nh_x h_y} \sum_{i=1}^{n} K[\frac{x-x_i}{h_x}, \frac{y-y_i}{h_y}]$$

h_x et h_y est un paramètre de largeur de bande qui contrôle le lissage de la PDF estimée,

et K(u,v) est une fonction noyau.

K(u,v) peut etre un noyau gaussien, le noyau d'Epanechnikov …

Une petite valeur de h donnera lieu à une estimation à haute variance qui s'ajuste étroitement aux données, tandis qu'une grande valeur de h donnera lieu à une estimation à faible variance qui est plus lisse mais peut s'ajuster insuffisamment aux données.

Le Noyau Gaussian bivariate :

K(u,v) = (1/2*pi)*exp(-(u^2 + v^2)/2)

L**e Noyau Epanechnikov bivariate:**

K(u,v) = (3/4)*(1 - (u^2 + v^2))

2 . Espérance de cet estimateur bivarié :

$$E(\hat{f}(x,y)) = E\left(\frac{1}{n h_x h_y} \sum_{i=1}^{n} K\left(\frac{x-X_i}{h_x}, \frac{y-Y_i}{h_y}\right)\right)$$

$$= \frac{1}{h_x h_y} E\left(K\left(\frac{x-X_1}{h_x}, \frac{y-Y_1}{h_y}\right)\right)$$

$$= \frac{1}{h_x h_y} \iint K\left(\frac{x-\mu_x}{h_x}, \frac{y-\mu_y}{h_y}\right) f_{X_1,Y_1}(\mu_x,\mu_y)\, d\mu_x\, d\mu_y$$

on pose : $t_x = \frac{x-\mu_x}{h_x}$ et $t_y = \frac{y-\mu_y}{h_y}$ et on obtient :

$$E(\hat{f}(x,y)) = \frac{1}{h_x h_y} \iint K(t_x,t_y) f_{X,Y}(x-h_x t_x, y-h_y t_y)\, h_x h_y\, dt_x\, dt_y$$

$$\boxed{= \iint K(t_x,t_y) f_{X_1,Y_1}(x - t_x V, y - t_y V)\, dt_x\, dt_y}$$

3 . Variance de l’estimateur

$$Var(\hat{f}(x,y)) = var\left(\frac{1}{n h_x h_y} \sum_{i=1}^{n} K\left(\frac{x-X_i}{h_x}, \frac{y-Y_i}{h_y}\right)\right)$$

$$= \frac{1}{n h_x^2 h_y^2} var\left(K\left(\frac{x-X_1}{h_x}, \frac{y-Y_1}{h_y}\right)\right)$$

$$= \frac{1}{n h_x^2 h_y^2}\left[E\left(K^2\left(\frac{x-X_1}{h_x}, \frac{y-Y_1}{h_y}\right)\right) - E^2\left(K\left(\frac{x-X_1}{h_x}, \frac{y-Y_1}{h_y}\right)\right)\right]$$

4 . Majoration de la variance :

- $\mathrm{Var}\left(\hat{f}(x,y)\right) = O\left(\frac{1}{n h_x h_y}\right)$

et ceci si f : * bornée, deux fois continuement différentiable

* à dérivées partielles bornées

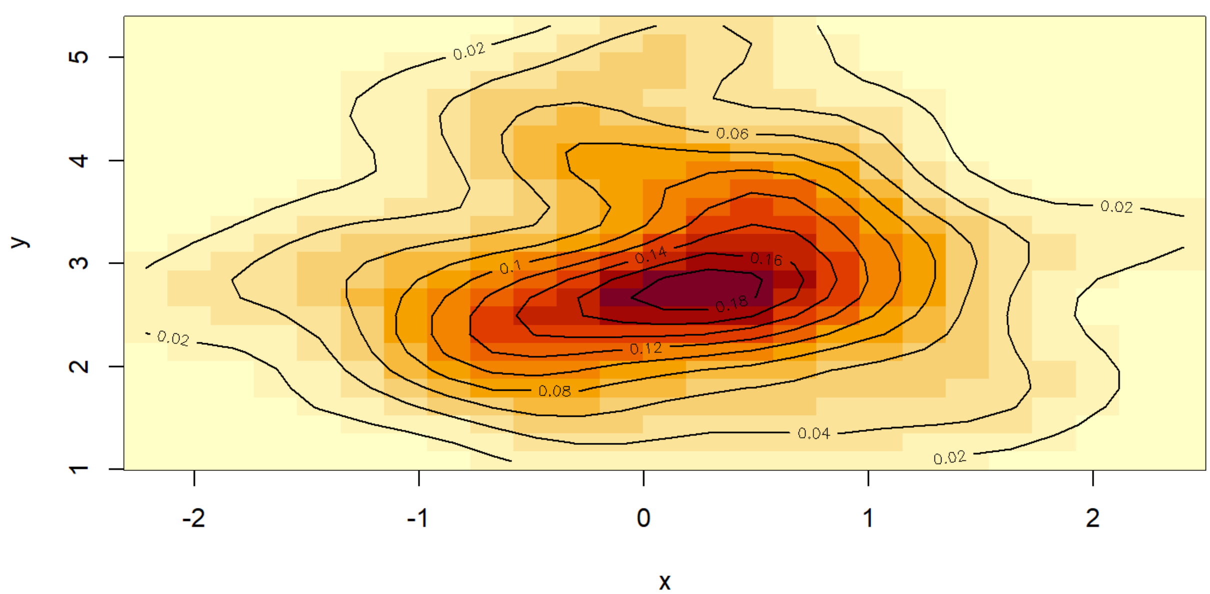
* si $\iint K^2(t_x, t_y)\, dt_x\, dt_y$ fini

* si $\iint |t_k| K^2(t_x, t_y)\, dt_x\, dt_y < +\infty$: $k = 1, 2$

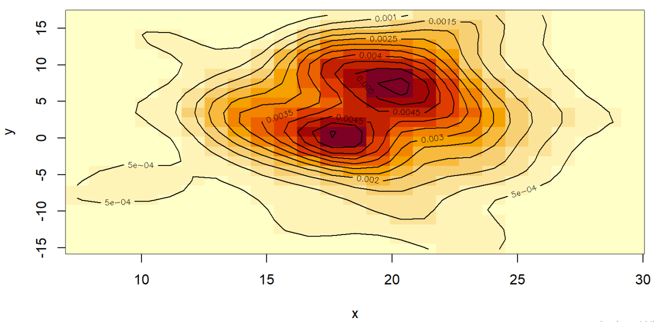
5 . Application : Pour un Kernel Gaussien

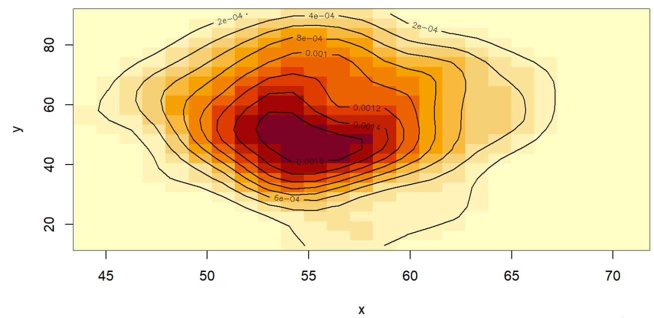
Cas X et Y indépendantes de loi respective N(0 , 1) et N(m, 1) :

```r
set.seed(1)
# Generate some x and y data
size =100
m=3
x <- rnorm(size,mean =0,sd=1)
y <- rnorm(size,mean =m,sd=1)
# Create dataframe
data <- data.frame(x, y )

# Estimate the joint density using kde2d
pdf <- kde2d(data$x, data$y)

# Plot the estimated density
image(pdf, xlab = "x", ylab = "y")
contour(pdf, add = TRUE)

```



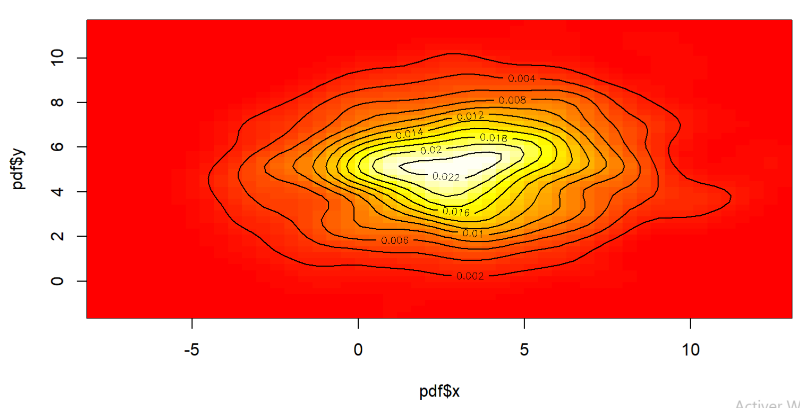
Cas X et Y indépendantes N(19, 4) et N(3, 7) VS N(56, 4) et N(56, 17) :



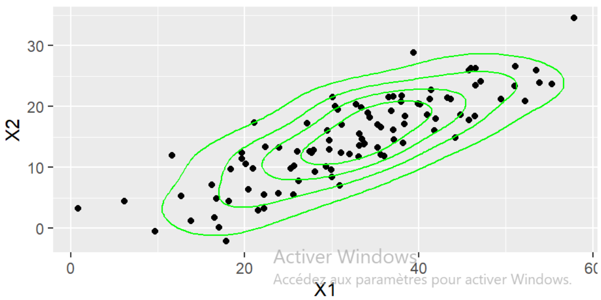


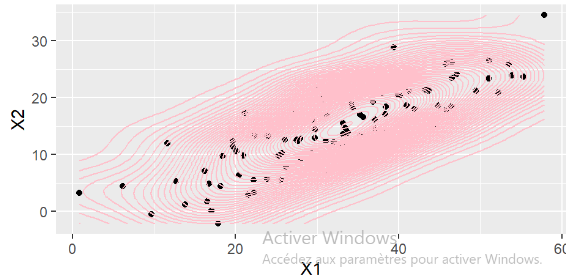
Cas (X, Y) vecteur gaussien de moyenne (m1 , m2) et de matrice de variance covariance telle que cov (X, Y)= r , var(X) = 9 et var(Y) = 4:

```
library(MASS)

# Set the mean vector and covariance matrix
mu <- c(3, 5)
sigma <- matrix(c(9, 0.5, 0.5, 4), nrow = 2)

# Generate random values of x and y
n <- 1000
data <- mvrnorm(n, mu, sigma)

# Compute the joint pdf using the density function
pdf <- kde2d(data[,1], data[,2], n = 50)

# Plot the joint pdf
image(pdf$x, pdf$y, pdf$z, col = heat.colors(50))
contour(pdf$x, pdf$y, pdf$z, add = TRUE)
```

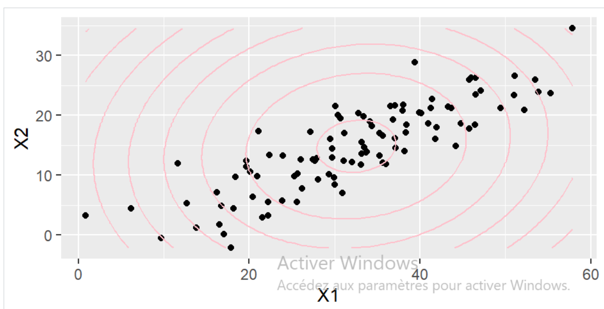


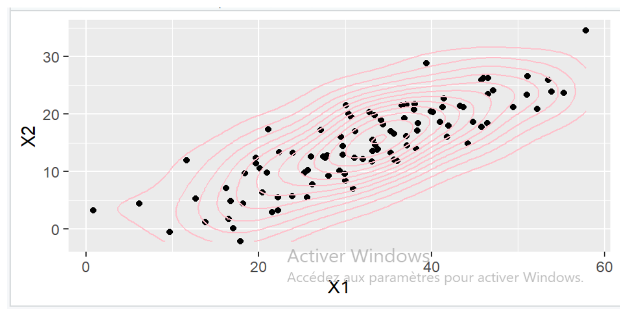
Resultats pour un choix de bins :bins=5 vs bins=50



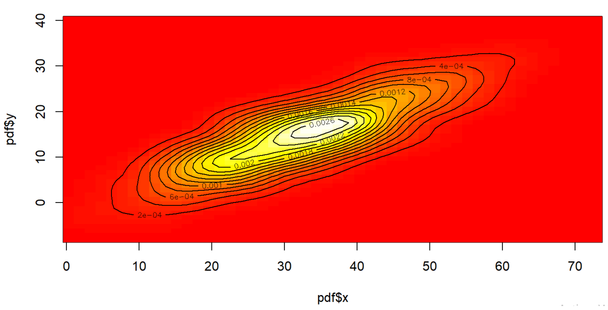


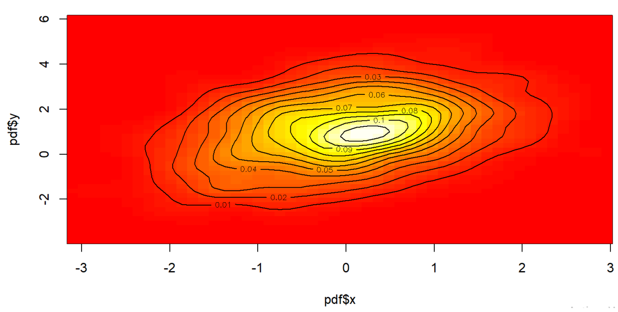
Pour h = 100 et h=15 :





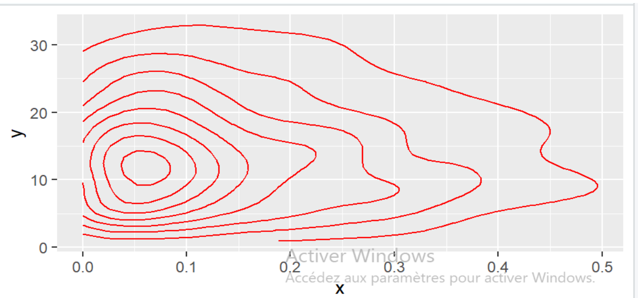
Resultat pour d'autres valeurs de la matrice variance- covariance(x,y) :
sigma <- matrix(c(145, 77, 77, 56) mu <- c(33,15) et
mu <- c(0, 1) , sigma <- matrix(c(1, 0.5, 0.5, 2), nrow = 2)

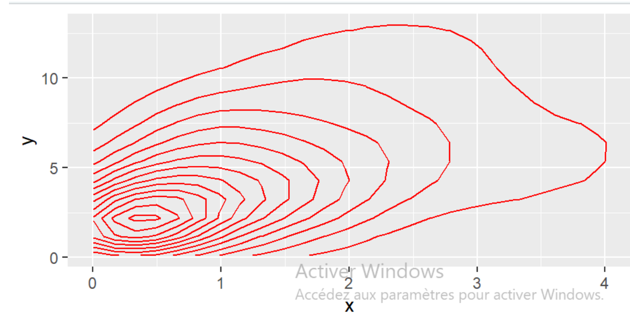




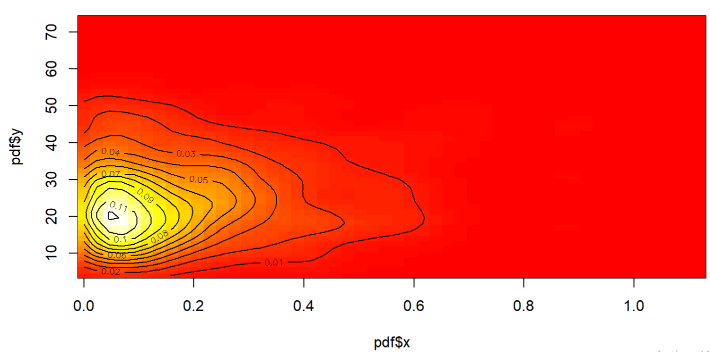
X de loi exponentielle de paramètre α et la loi conditionnelle de Y sachantX=x est la loi gamma de paramètre (3 , x+ α) :

Alpha=5 : **Alpha=0.5**



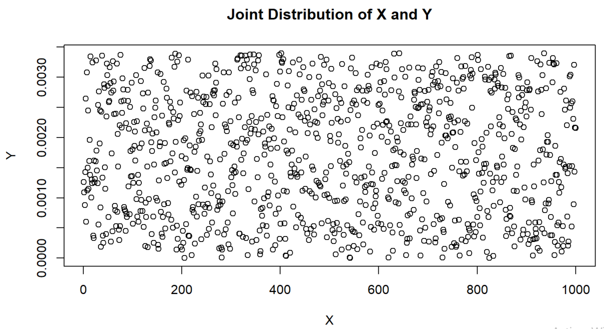



```
# Set the parameters
alpha <- 5

# Generate a random sample of X values
n <- 1000
x <- rexp(n, rate = alpha)

# Generate a random sample of Y values for each X value
y <- rep(0, n)
for (i in 1:n) {
  y[i] <- rgamma(1, shape = alpha, rate = 1 / (x[i] + alph
}
# Combine X and Y into a dataset
data <- data.frame(x, y)

# Compute the joint pdf using the kdensity2d function
pdf <- kde2d(data$x, data$y, n = 50)

# Plot the joint pdf
image(pdf$x, pdf$y, pdf$z, col = heat.colors(50))
contour(pdf$x, pdf$y, pdf$z, add = TRUE)
```



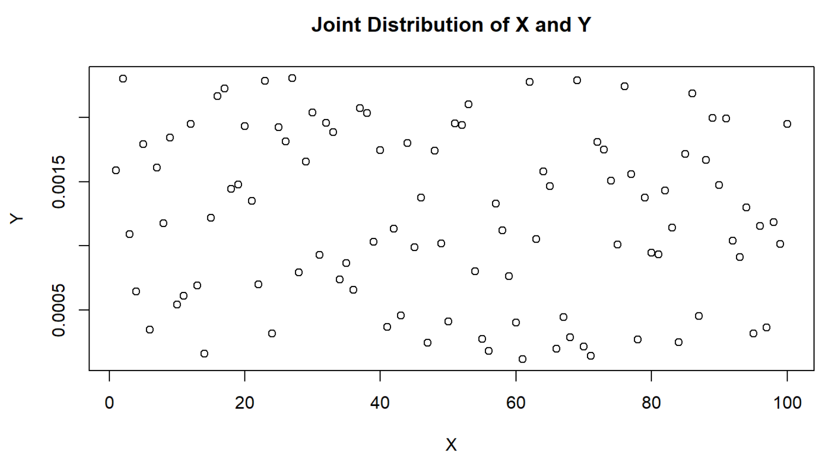
Densité theorique : Gaussian vector

```r
# Actual
# Vrai valeurs de xy
xy_actual <- dmvnorm(xy, mean =mu , sigma = sigma,log = FALSE)


plot(xy_actual, type = "p", main = "Joint Distribution of X and Y", xlab = "X", ylab = "Y")
```



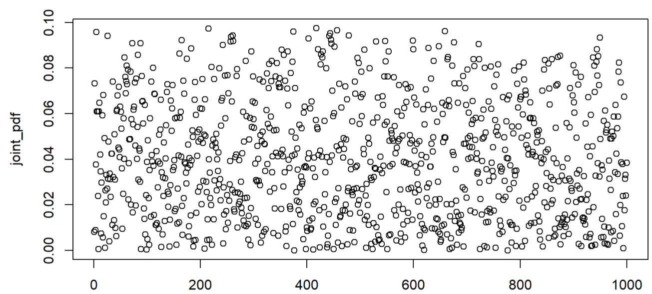
Densité theorique : Two Independent Normal RV

```r
# Actual Data :
x_actual <- dnorm(x,mean =56,sd=4)
y_actual <- dnorm(y,mean =56,sd=17)

# Joint PDF
joint_pdf <- x_actual * y_actual

# Reshape the PDF to a matrix and plot it as a heatmap


plot(joint_pdf, type = "p", main = "Joint Distribution of X and Y", xlab = "X", ylab = "Y")
```



Densité theorique : X : Normale :Y: Conditionelle Gamma

```
# Actual
x_actual <- dexp(x, rate = alpha)


y_actual <- rep(0, n)
for (i in 1:n) {
  y_actual[i] <- dgamma(y[i],shape = alpha, rate = 1 / (x_actual[i] + alpha))
}

joint_pdf <- x_actual* y_actual

plot(joint_pdf)
```



6. Calcul de la MSE :

```
> mean((xy_actual- xy)^2)
[1] 777.2714
```