Multivariate Normal and R

Bealy MECH

10/18/2021

Exercice 1 : IQ (Intelligence Quotient)

La probabilité pour que l'IQ est supérieur à 120 est donnée par:

$$P(IQ > 120) = \int_{120}^{+\infty} f(x)dx$$

La probabilité pour que l'IQ est inférieur à 100 est donnée par:

$$P(IQ < 100) = \int_{-\infty}^{100} f(x)dx$$

Ce qui peut etre calculé en R parla fonction $QI.Sup.120$ et $QI.Inf.100$ (pour cela il faut installer la $library(ggplot2)$)

The probability of having IQ:

- more than 120

```
Psup120 <- 1-pnorm(120,100,15) # mean=100, sd=15
Psup120
```

```
## [1] 0.09121122
```

- less than 100

```
Pinf100 <- pnorm(100,100,15)
Pinf100
```

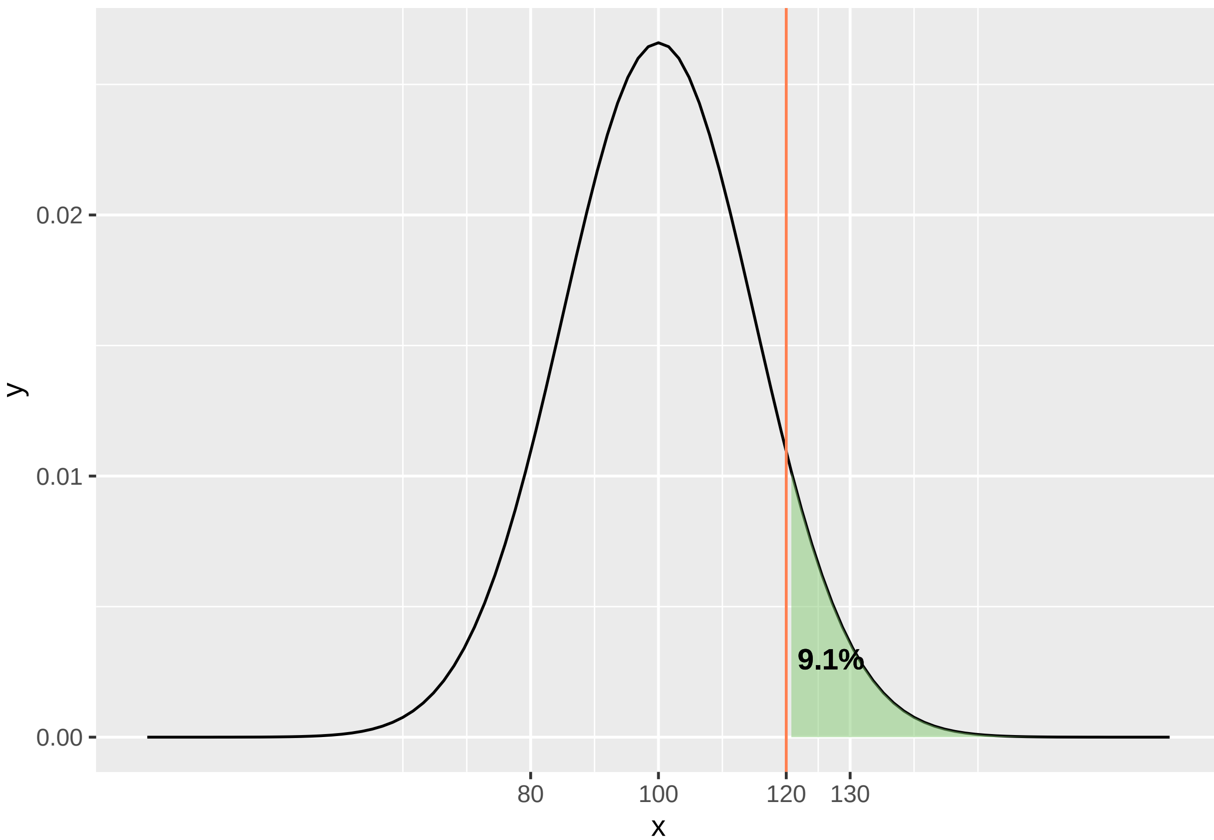
```
## [1] 0.5
```

Let's see the graphe of the probability of having IQ more than 120

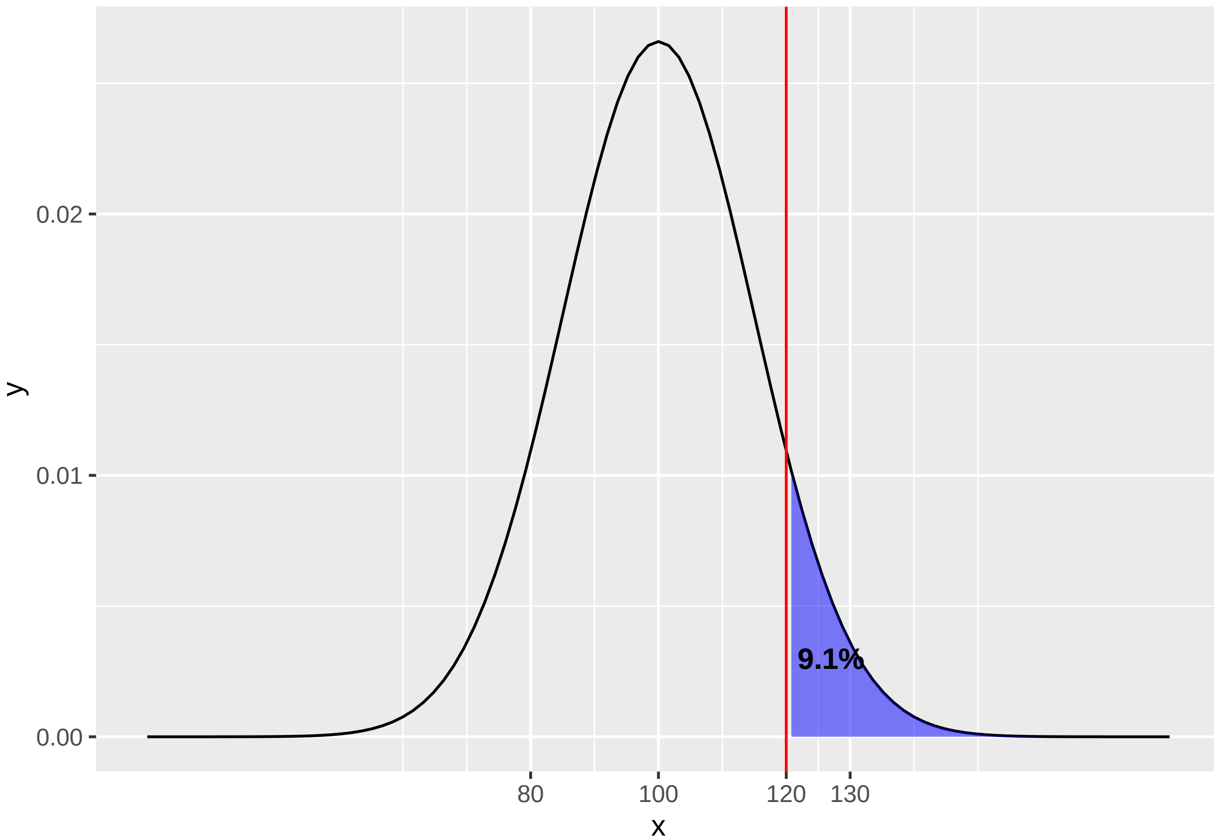
```
# 1st solution
QI.Sup.120 <- function(x){
  ifelse(x>120, dnorm(x,mean=100,sd=15), NA)
}
# test
QI.Sup.120(140)
```

```
## [1] 0.0007597324
```

```
library(ggplot2)
x <- c(20,180)
ggplot(data.frame(x), aes(x)) +
  stat_function(fun = dnorm, args = list(mean=100, sd=15)) +
  stat_function(fun = QI.Sup.120, geom = "area", fill="#84CA72", alpha=0.5, mapping = NULL) +
  geom_text(x=127, y=0.003, size=4, fontface="bold",
            label = paste0(round(pnorm(120, mean=100, sd=15, lower.tail = FALSE),3) * 100, "%")) +
  scale_x_continuous(breaks = c(80,100,120,130)) +
  geom_vline(xintercept = 120, colour = "coral")
```



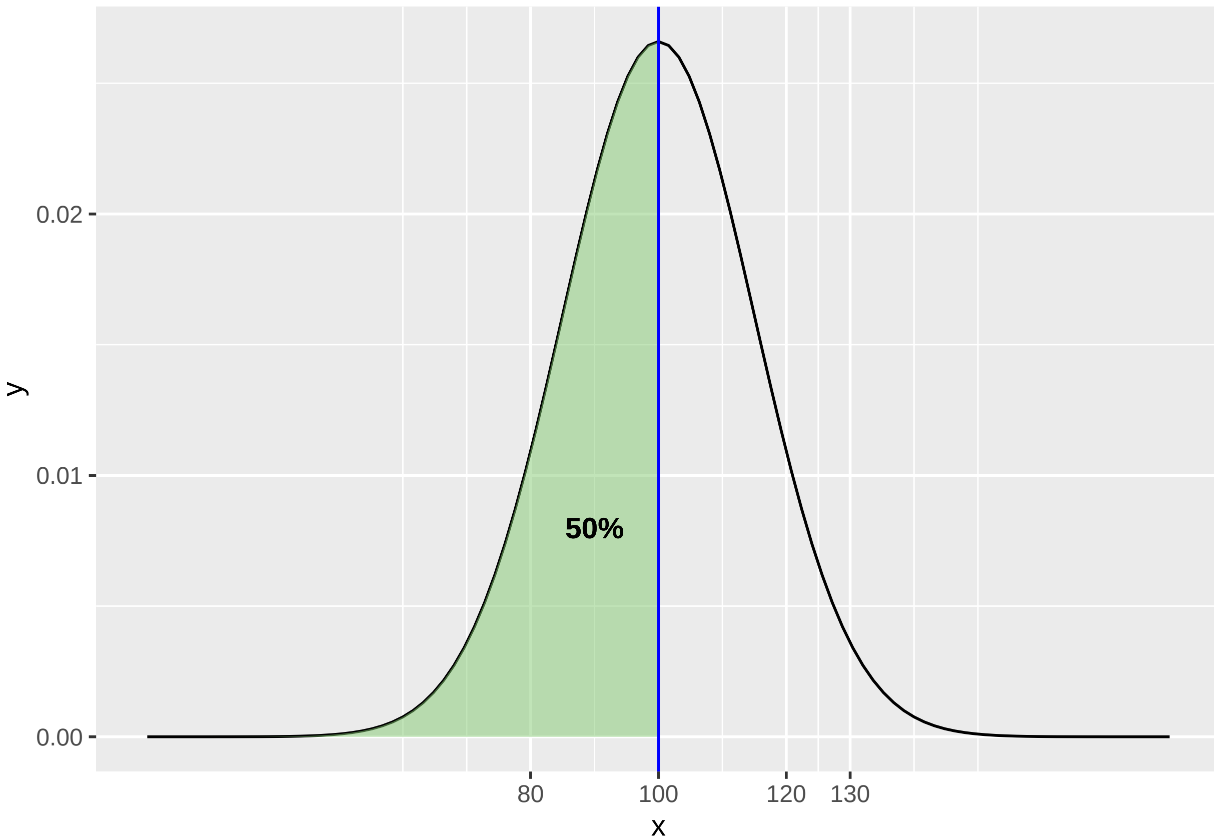

```
# 2nd solution
IQ_sup_120 <- function(x){
  ifelse(x >120,dnorm(x, mean=100, sd=15),NA)
}
library(ggplot2)
ggplot(data.frame(x=c(20,180)),aes(x)) +
  stat_function(fun=dnorm,args=list(mean=100,sd=15)) +
  stat_function(fun=IQ_sup_120,geom="area", fill="blue", alpha=0.5) +
  geom_text(x=127, y=0.003, size=4, fontface="bold",
            label = paste0(round(pnorm(120, mean=100, sd=15, lower.tail = FALSE),3) * 100, "%")) +
  scale_x_continuous(breaks = c(80,100,120,130)) +
  geom_vline(xintercept = 120, colour = "red")
```



Let's see the graphe of the probability of having IQ less than 100

```
# 1st solution
QI.Inf.100 <- function(x){
  ifelse(x<=100, dnorm(x,mean=100,sd=15), NA)
}

library(ggplot2)
x <- c(20,180)
ggplot(data.frame(x), aes(x)) +
  stat_function(fun = dnorm, args = list(mean=100, sd=15)) +
  stat_function(fun = QI.Inf.100, geom = "area", fill="#84CA72", alpha=0.5, mapping = NULL) +
  geom_text(x=90, y=0.008, size=4, fontface="bold",
            label = paste0(round(pnorm(100, mean=100, sd=15, lower.tail = FALSE),3) * 100, "%")) +
  scale_x_continuous(breaks = c(80,100,120,130)) +
  geom_vline(xintercept = 100, colour = "blue")
```



Exercice 2 : Bias of the maximum likelihood estimator of the variance

L'estimateur du maximum de vraisemblance de la variance est donné par:

$$S_n^2 = \frac{1}{n}\sum_{i=1}^{n} X_i^2 - \bar{X}_n^2$$

Pour calculer son espérance on calcule d'abord $\mathbb{E}[\bar{X}_n^2]$

Par définition,

$$\bar{X}_n = \frac{1}{n}\sum_{i=1}^{n} X_i$$

$$\implies \mathbb{E}[\bar{X}_n^2] = \frac{1}{n^2}\mathbb{E}\left[\sum_{i=1}^{n} X_i^2 + \sum_{i=1}^{n}\sum_{i\neq j} X_i X_j\right]$$

Puisque les échantillons sont i.i.d. on a alors:

$$\mathbb{E}[\bar{X}_n^2] = \frac{1}{n^2}\left[nE\left[X^2\right] + n(n-1)E[X]^2\right]$$

On a donc:

$$\begin{aligned}
\mathbb{E}\left[S_n^2\right] &= \frac{1}{n}E[\sum_{i=1}^{n} X_i^2] - \frac{1}{n}E\left[X^2\right] + \frac{n-1}{n}E[X]^2 \\
&= \frac{n-1}{n}\left(\mathbb{E}[X^2] - \mathbb{E}^2[X]\right) \\
&= \frac{n-1}{n}\operatorname{Var}[X]
\end{aligned}$$

Pour obtenir un estimateur non biaisé il suffit de corriger le biais multiplicatif:

$$V_n = \frac{n}{n-1}S_n^2$$

Exercice 3 : Fisher Iris Data

```
library(tibble)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(tidyr)
data(iris)
dim(iris)
```

```
## [1] 150   5
```

```
summary(iris$Species) # show the species of data iris and its number
```

```
##     setosa versicolor  virginica
##         50         50         50
```

```
# There are 3 species: setosa, versicolor and virginica.
# The data set consists of 50 samples from each of three species
head(iris)
```

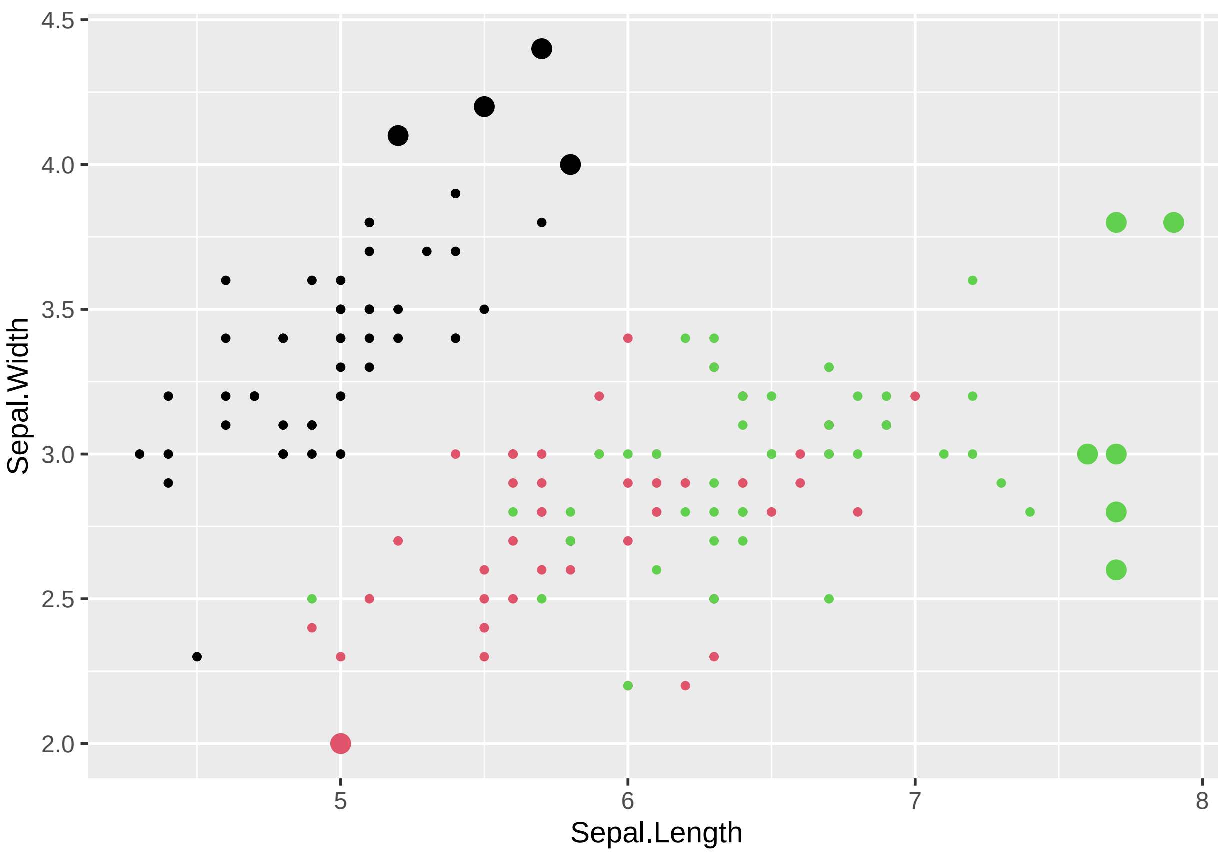
```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
## 6          5.4         3.9          1.7         0.4  setosa
```

Find the flowers whose measured widths and lengths are exceptionally large or small:

```
params <- as_tibble(iris) %>%
          select(-"Species") %>% # delete Species from the dataset
          gather(factor_key = TRUE) %>%
          group_by(key) %>%
          summarise(mean= mean(value), sd = sd(value)) %>%
          mutate(min = mean - 2*sd, max = mean + 2*sd)
params
```

```
## # A tibble: 4 x 5
##   key           mean    sd    min   max
##   <fct>        <dbl> <dbl>  <dbl> <dbl>
## 1 Sepal.Length  5.84 0.828  4.19   7.50
## 2 Sepal.Width   3.06 0.436  2.19   3.93
## 3 Petal.Length  3.76 1.77   0.227  7.29
## 4 Petal.Width   1.20 0.762 -0.325  2.72
```

```
# the flowers whose measured widths and lengths are exceptionally large or small
flower.outliers <- (apply(t((t(iris[,1:4]) < params$min) + (t(iris[,1:4]) > params$max)),1,sum)>0)
ggplot(iris, aes(x=Sepal.Length, y=Sepal.Width)) +
      geom_point(colour=as.numeric(iris$Species), size= flower.outliers*2 + 1)
```



Exercice 4 : Equiprobability Ellipses

Let $(x^1, \ldots, x^p)$ are i.i.d. variables following $\mathcal{N}(0,1)$, then $\left(x^1, \ldots, x^p\right) \sim \mathcal{N}_p\left(0, I_p\right)$.

Find a matrix A of size (p,p) such that Ax has variance Σ, i.e. $AA^t = \Sigma$.

Sevral solutions are possible to find the matrix A:

- **Cholesky** :

$$\Sigma = TT^t$$

  where T is a triangular inferior matrix ($A = T^t$)

- **SVD** (Singular Value Decomposition)

$$\Sigma = UDU^t$$

  where D is a diagonal matrix of eigenvalues and U is an orthogonal matrix of eigenvectors.

Then we obtain:

$$\boldsymbol{y} = A\boldsymbol{x} + \boldsymbol{\mu} \sim \mathcal{N}_p\left(0, \Sigma\right)$$

If $\boldsymbol{x} \sim \mathcal{N}_p(\boldsymbol{\mu}, \Sigma)$ so that $\boldsymbol{y} = \Sigma^{-1/2}(\boldsymbol{x} - \mu) \sim \mathcal{N}_p\left(0, I_p\right)$ and we obtain:

$$Q = \boldsymbol{y}^t\boldsymbol{y} \sim \chi^2_p$$

The equation below show the probability of Q:

$$P(Q \leq q) = \alpha$$

with $q = \chi^2_{p,\alpha}$ defines an $\alpha - \mathit{level}$ of equiprobability ellipsoid.

- Generate 1000 observations of a two-dimensional normal distribution $\mathcal{N}_p\left(\mu, \Sigma\right)$ with:

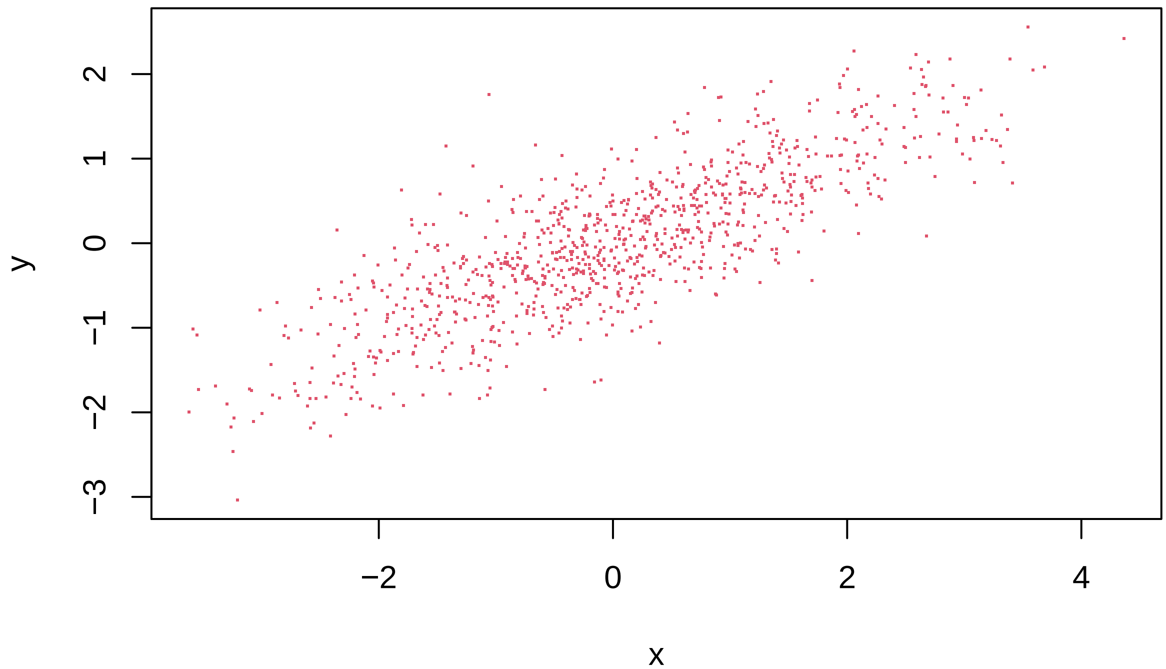
$$\mu = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

$$\Sigma = \begin{pmatrix} 2 & 1 \\ 1 & 0.75 \end{pmatrix}$$

```
# par(mfrow=c(1,3)) # pour partager l'affichage en 2
sigma <- matrix(c(2,1,1,0.75),2,2) # la matrice de grand sigma (matrice de variance)
A <- chol(sigma)
# check the sigma matrix
t(A)%*%A
```

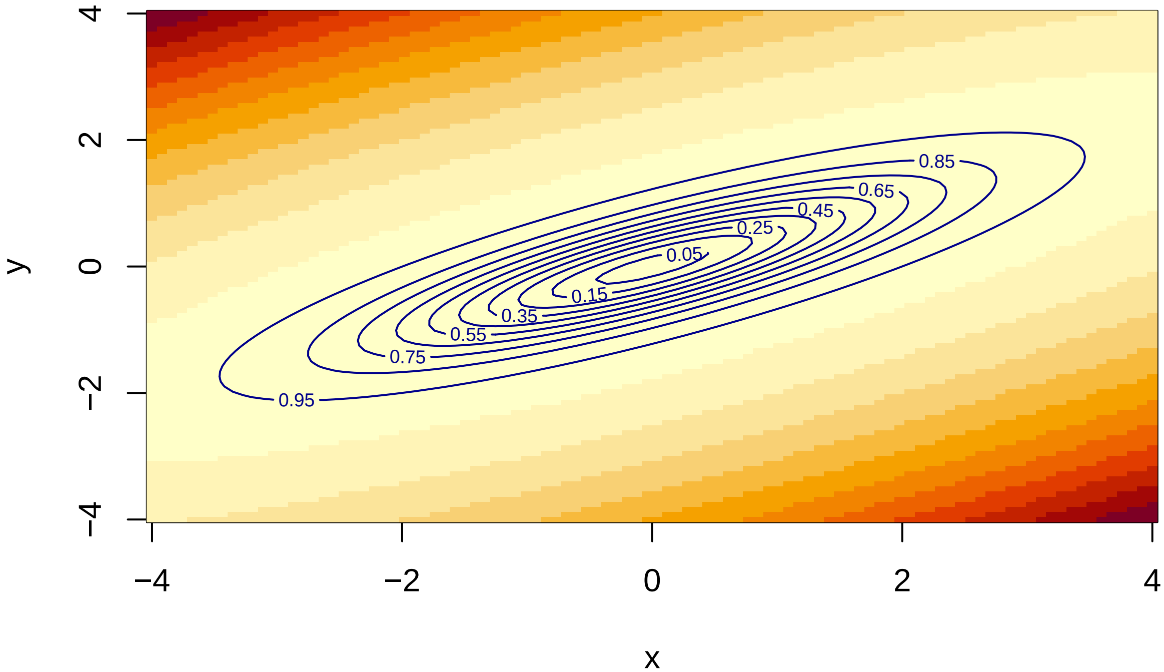
```
##      [,1] [,2]
## [1,]    2 1.00
## [2,]    1 0.75
```

```
Y <- matrix(rnorm(2000),1000,2) %*% A # une matrice avec 1000 lignes et 2 colonnes
# le grapghe de Y
plot(Y, xlab = "x", ylab = "y", pch = ".", col="2")
```



Draw the ellipses of equiprobability of the multiples of 5%

```
x <- seq(-4,4,length = 100)
y <- seq(-4,4,length = 100)
sigmainv <- solve(sigma) # inverse matrix of sigma
a <- sigmainv[1,1] # THE ELEMENT OF 1ST ROW AND 1ST COLUMN
b <- sigmainv[2,2]
c <- sigmainv[1,2]
z <- outer(x,y,function(x,y) (a*x^2 + b*y^2 + 2*c*x*y)) # the function of an ellipse
image(x,y,z)
p <- seq(0.05,0.95,by=0.1)
Q <- qchisq(p, df=2)
contour(x,y,z,col = "blue4", levels = Q, labels = p, add=T)
```




```
persp(x,y,1/(2*pi)*det(sigmainv)^(-1/2)*exp(-0.5*z), col = "cornflowerblue", theta = 5, phi = 10, zlab =
```