

```

# Exercice 13 : efficacite asymptotique et maximum de vraisemblance
#
#a) On s'interesse dans un ler temps ...
voir.biais.efficacite = fonction(tab.n)
{
  # on considere une loi N(0,1)
  tab.biais=1:length(tab.n)
  tab.variance=1:length(tab.n)
  for (i in tab.n)
  {
    theta.n=1:1000
    for(j in 1:1000)
    {
      x=rnorm(tab.n[i])
      theta.n[j]=mean(x)
    }
    tab.biais[i]=mean(theta.n)
    tab.variance[i]=var(theta.n)
  }
  split.screen(c(2,1))
  screen(1)
  plot(tab.n,tab.biais,type="l",col="blue",main="Biais")
  lines(tab.n,0*tab.n,lty="dotted",col="red")
  screen(2)
  plot(tab.n,tab.n*tab.variance,type="l",col="blue",main="Efficacite")
  lines(tab.n,0*tab.n+1,lty="dotted",col="red") # borne Cramer-Rao ...
}
voir.biais.efficacite(1:20)
voir.biais.efficacite(1:200)
# b) On s'interesse maintenant a l'estimation ...
voir.biais.efficacite = fonction(tab.n)
{
  # on considere une loi N(0,1)
  tab.biais=1:length(tab.n)
  tab.variance=1:length(tab.n)
  for (i in tab.n)
  {
    estim.n=1:1000
    for(j in 1:1000)
    {
      x=rnorm(tab.n[i])
      estim.n[j]=pnorm(mean(x))
    }
    tab.biais[i]=mean(estim.n)-0.5
    tab.variance[i]=var(estim.n)
  }
  split.screen(c(2,1))
  screen(1)
  plot(tab.n,tab.biais,type="l",col="blue",main="Biais")
  lines(tab.n,0*tab.n,lty="dotted",col="red")
  screen(2)

  plot(tab.n,2*pi*tab.n*tab.variance,type="l",col="blue",main="Efficacite

```

```

    lines(tab.n,0*tab.n+1,lty="dotted",col="red") # borne Cramer-Rao ...
}
voir.biais.efficacite(1:50)
voir.biais.efficacite(1:500)
#
# Exercice 14 : estimation non parametrique par noyaux
#
plot(density(c(-20, rep(0,98), 20)), xlim = c(-4, 4))
#a) Ecrire la fonction get.sample ...
get.sample = fonction(n)
{
  #generer un echantillon uniforme sur [0,2] de taille n/2
  x.unif=runif(n/2,max=2)
  #generer un echantillon N(5,1) de taille n/2
  x.norm=rnorm(n/2,mean=5)
  c(x.unif,x.norm)
}
#b) Ecrire la fonction show all ...
show.all = fonction(x,h)
{
  # tracer la densite estimee avec un noyau uniforme
  f.unif=density(x,bw=h,kernel="rectangular")
  plot(f.unif$x,f.unif$y,type="l")
  # tracer la densite estimee avec noyau gaussien
  f.norm=density(x,bw=h)
  lines(f.norm$x,f.norm$y,type="l",col="blue")
  # tracer la vraie densite
  f.true=0.5*dunif(f.norm$x,max=2)+0.5*dnorm(f.norm$x,mean=5)
  lines(f.norm$x,f.true,type="l",col="red")
}
x=get.sample(1000); hist(x, probability = T, br = 50)
h = 0.5
f.unif=density(x,bw=h,kernel="rectangular")
lines(f.unif$x,f.unif$y,type="l", col = "orange")
f.norm=density(x,bw=h)
lines(f.norm$x,f.norm$y,type="l",col="green")

f.true=0.5*dunif(f.norm$x,max=2)+0.5*dnorm(f.norm$x,mean=5)
lines(f.norm$x,f.true,type="l",col="red")

show.all(x,0.1)
#####
x=rnorm(1000); hist(x, probability = T, br = 20)

hist(x, probability = T, br = 200)

# xx = seq(-3,3,0.1)
# lines(xx, dnorm(xx), col = "red")

h = 0.1
f.unif=density(x,bw=h,kernel="rectangular")
lines(f.unif$x,f.unif$y,col="blue")
f.norm=density(x,bw=h)
lines(f.norm$x,f.norm$y,col="green")
#c) Fenetre optimale

```

```
h=bw.nrd0(x)
```