Correction du TP1 : Estimation dans le modèle de bruit blanc gaussien

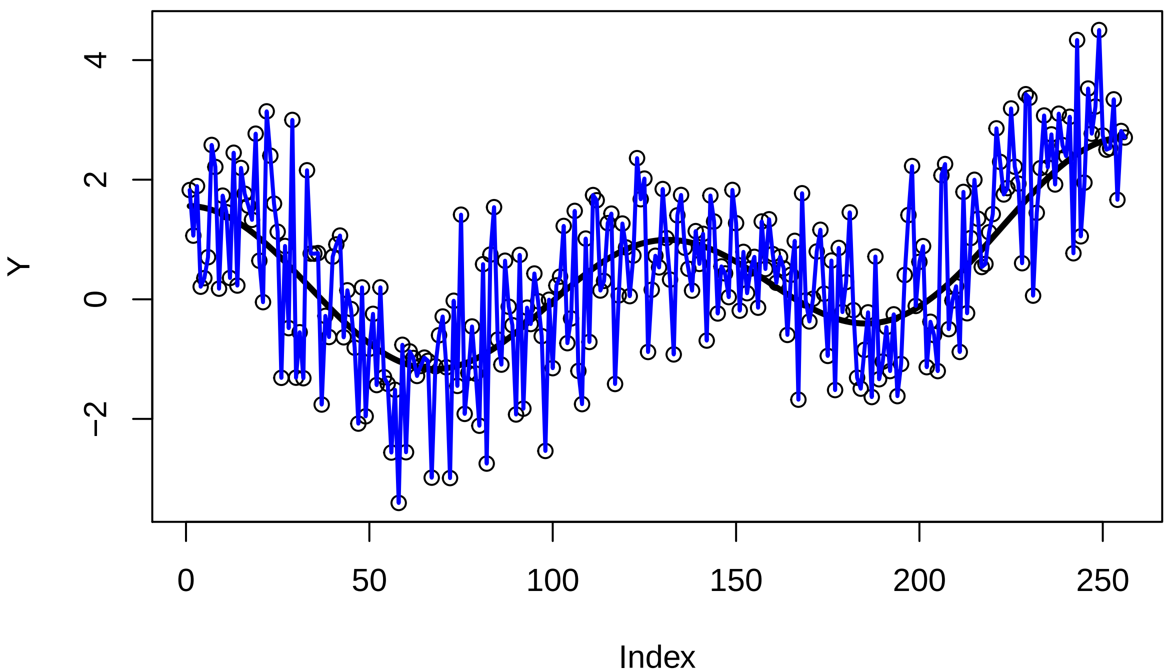
M2 Statistique, 2019-2020

Statistique mathématique en grande dimension et applications, cours de M. Roquain

Exercice 1 : Application du seuillage au débruitage d'un signal

1)

```
load("dataforTP1")
f=freg
n=length(f)
Y = f + rnorm(n)
plot(Y)
lines(f,lwd=3)
lines(Y,col="blue",lwd=2)
```



Il y a overfitting

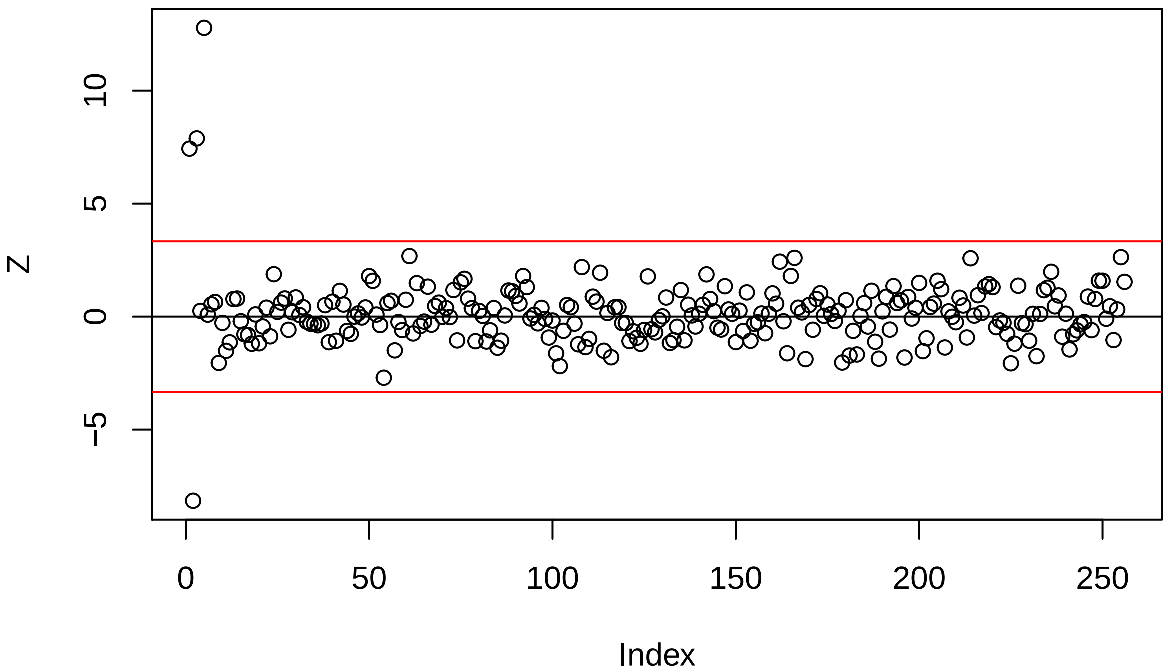
2)

```
X=ComputeCosMat(n)
(t(X)%*%X)[1:4,1:4]
```

```
##               [,1]          [,2]          [,3]          [,4]
## [1,]  1.000000e+00  1.413800e-16 -1.587272e-16  7.025630e-17
## [2,]  1.413800e-16  1.000000e+00 -2.862294e-17 -1.431147e-16
## [3,] -1.587272e-16 -2.862294e-17  1.000000e+00 -1.977585e-16
## [4,]  7.025630e-17 -1.431147e-16 -1.977585e-16  1.000000e+00
```

3)

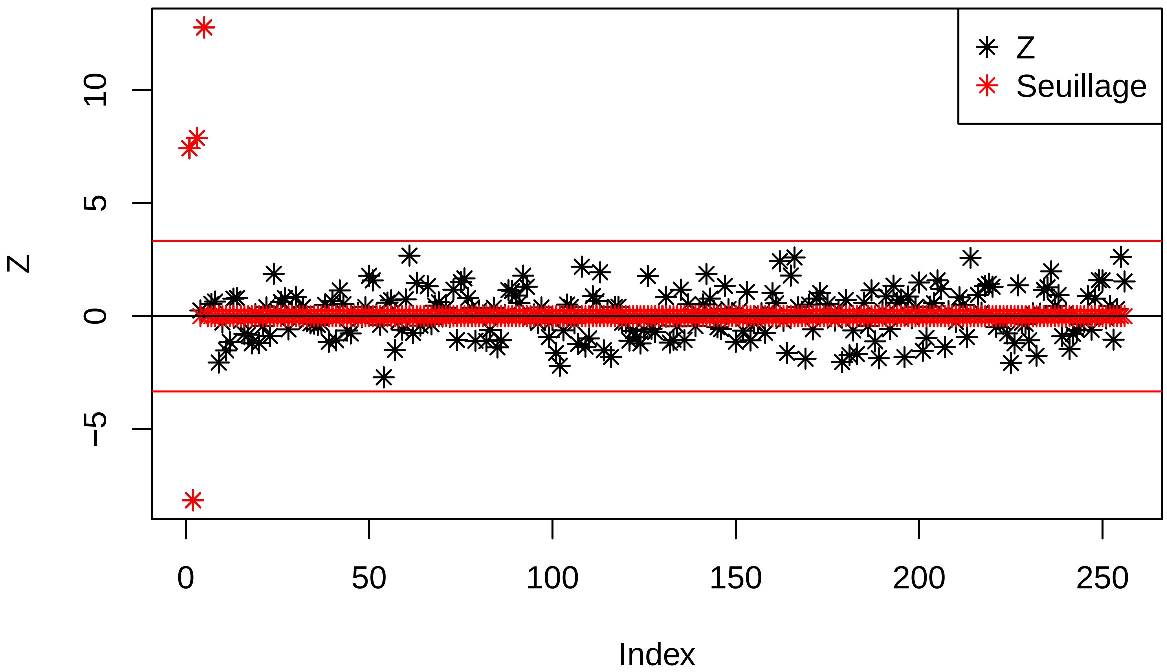
```
Z=t(X)%*%Y
plot(Z)
abline(h=0)
abline(h=sqrt(2*log(n)),col="red")
abline(h=-sqrt(2*log(n)),col="red")
```



Nous voyons quelques coefficients qui ont l'air “grand“, les autres étant petits et concentrés autour de 0. Donc $\theta = \mathbb{E}Z$ a l'air sparse (en tout cas plus que le vecteur f lui-même).

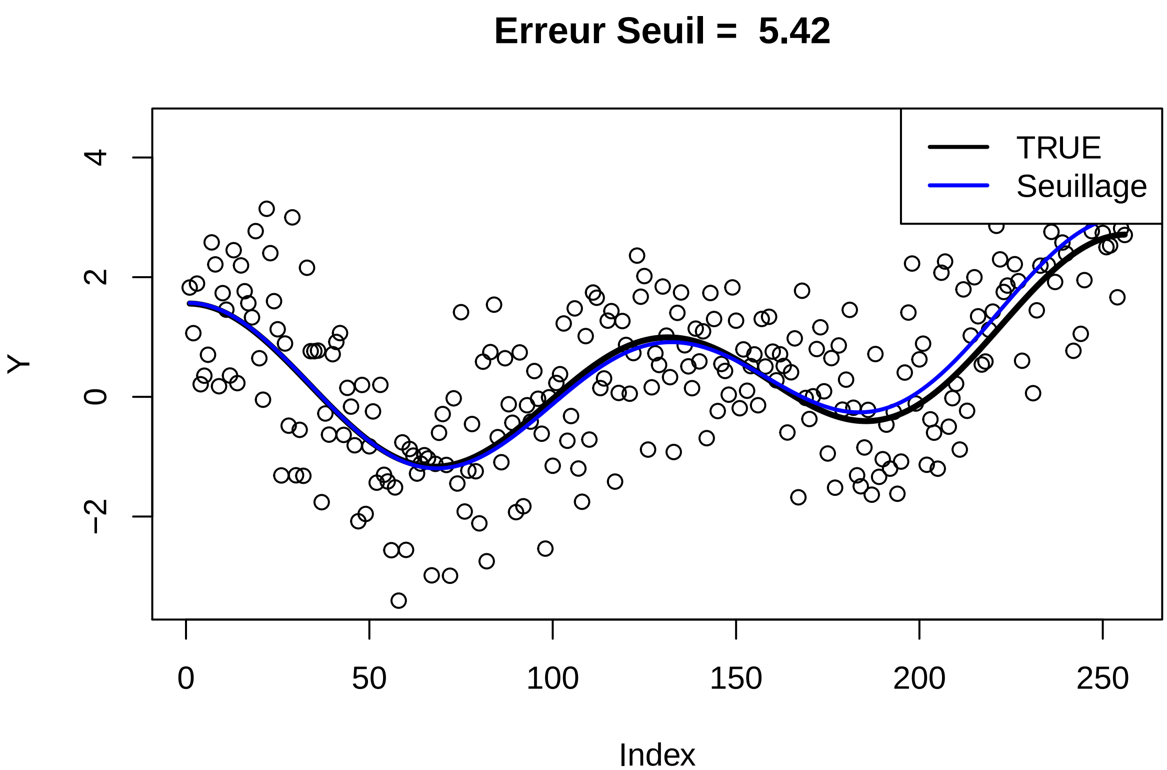
4)

```
p=n
t=sqrt(2*log(p))
thetachap=Z*(abs(Z)>=t)
plot(Z,pch=8)
abline(h=t,col="red")
abline(h=-t,col="red")
points(thetachap,col="red",pch=8)
abline(h=0)
legend("topright",c("Z","Seuillage"),col=c("black","red"),
       pch=c(8,8),bg="white")
```



5)

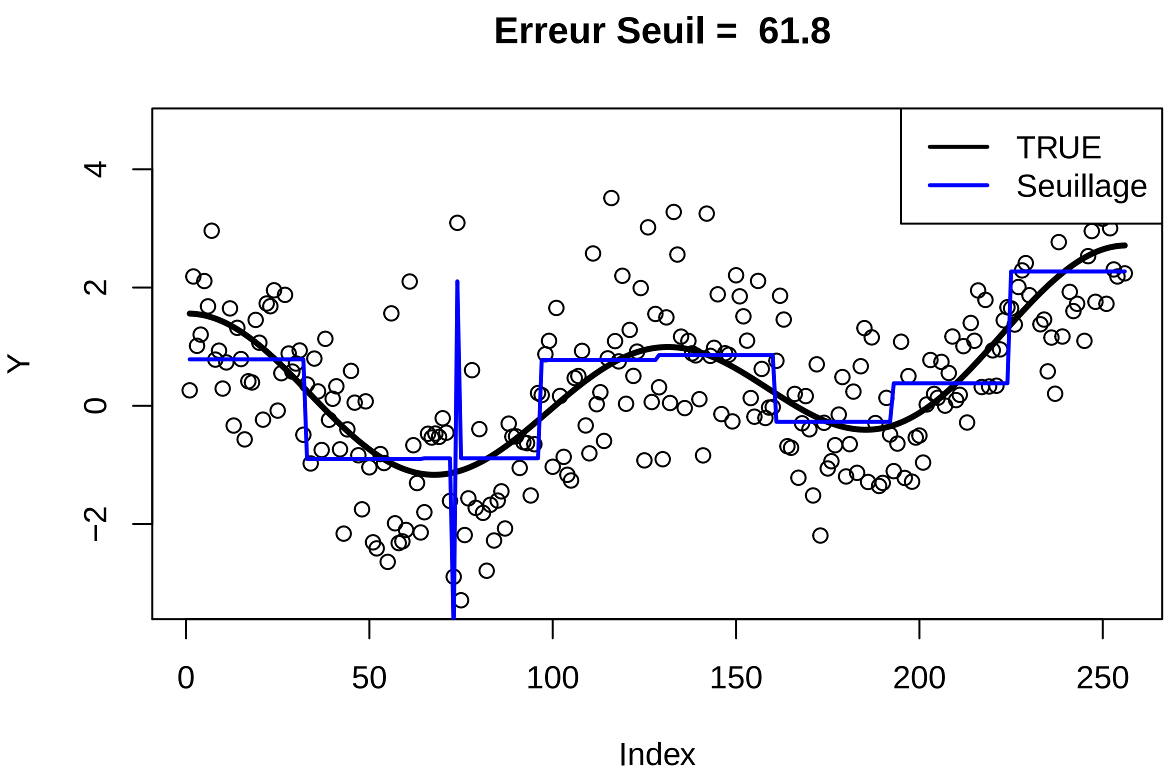
```
fhat=X%*%thetachap
erreurseuil=sum((fhat-f)^2)
plot(Y)
lines(f,lwd=3)
lines(fhat,col="blue",lwd=2)
legend("topright",c("TRUE","Seuillage"),col=c("black","blue"),
       lwd=c(2,2),bg="white")
title(paste(" Erreur Seuil = ",signif(erreurseuil,3),sep=" "))
```



L'estimation est bien meilleure. On voit que l'erreur est pas très loin de ce que prédit la théorie $2 * s * \log(n)$.

6)

```
f=freg
n=length(f)
Y = f + rnorm(n)
X=ComputeHaarMat(n)
Z=t(X)%*%Y
p=n
t=sqrt(2*log(p))
thetachap=Z*(abs(Z)>=t)
fhat=X%*%thetachap
erreurseuil=sum((fhat-f)^2)
plot(Y)
lines(f,lwd=3)
lines(fhat,col="blue",lwd=2)
legend("topright",c("TRUE","Seuillage"),col=c("black","blue"),
       lwd=c(2,2),bg="white")
title(paste(" Erreur Seuil = ",signif(erreurseuil,3),sep=" "))
```



On voit que la base de Haar est peu adaptée. Mais c'est quand même mieux que $\hat{f} = Y$

```
sum((Y-f)^2)
```

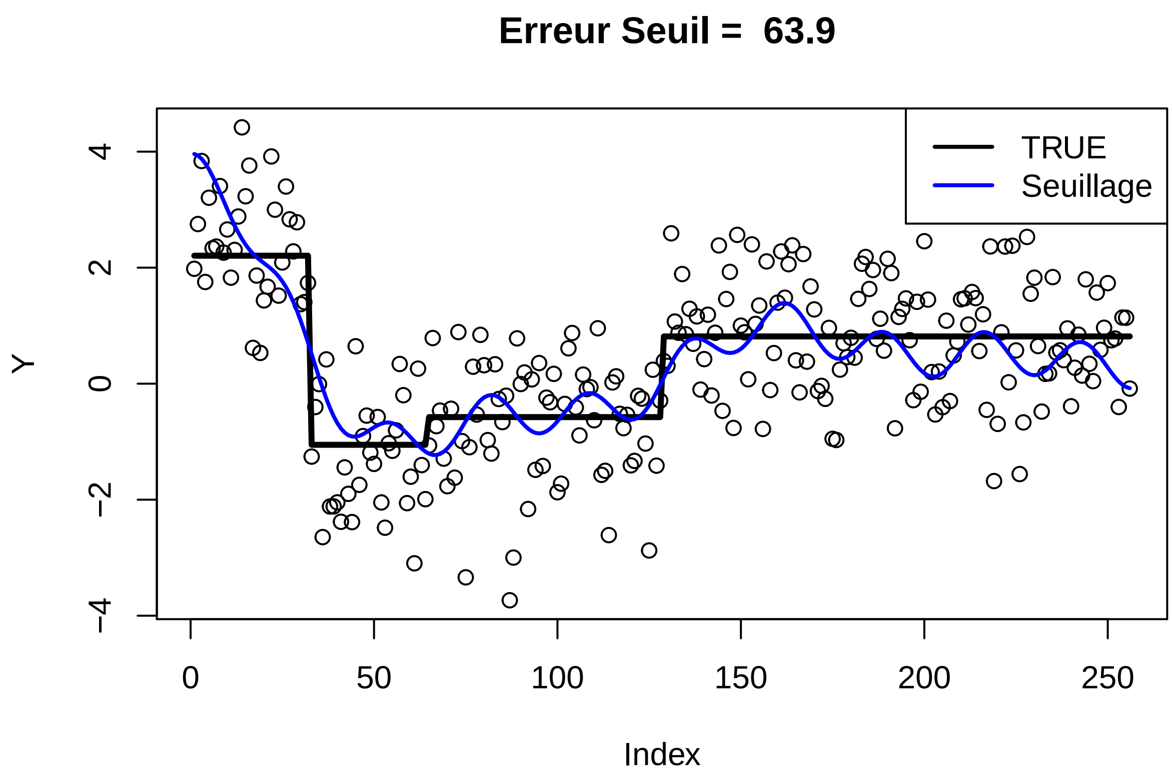
```
## [1] 263.6142
```

7)

Voyons pour un signal non régulier.

Avec la base Cosine

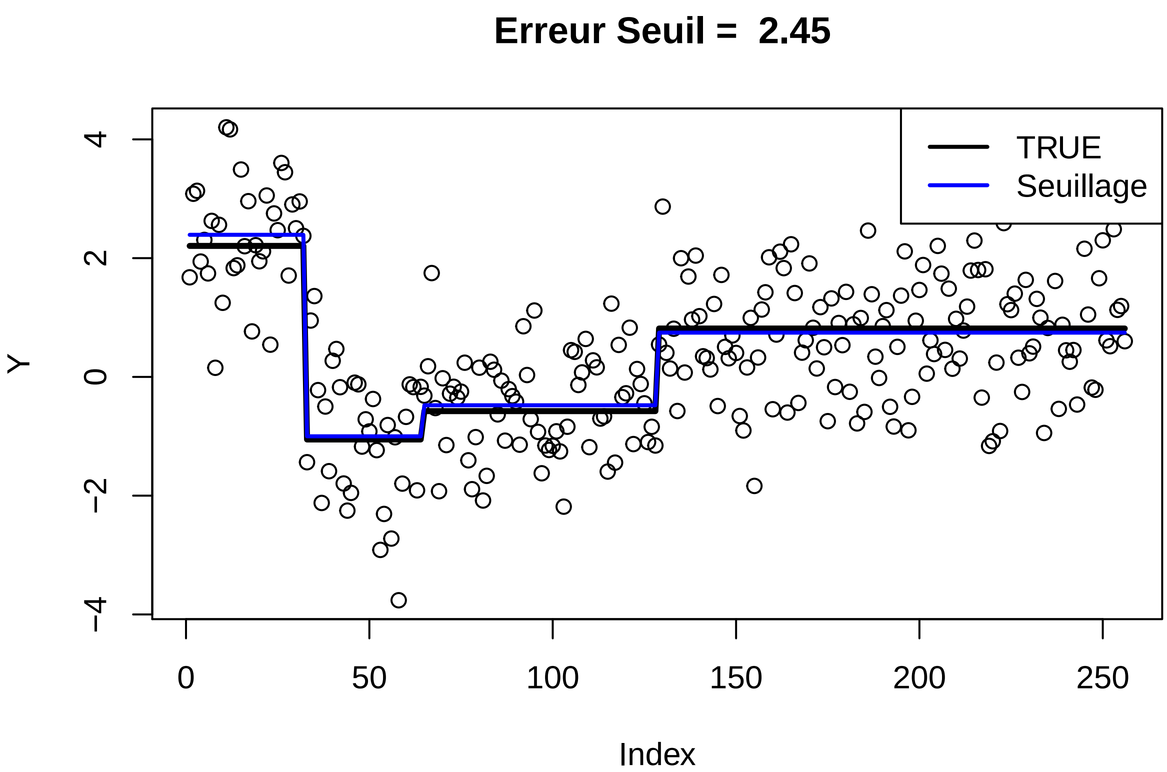
```
f=fstep
n=length(f)
Y = f + rnorm(n)
X=ComputeCosMat(n)
Z=t(X)%*%Y
p=n
t=sqrt(2*log(p))
thetachap=Z*(abs(Z)>=t)
fhat=X%*%thetachap
erreurseuil=sum((fhat-f)^2)
plot(Y)
lines(f,lwd=3)
lines(fhat,col="blue",lwd=2)
legend("topright",c("TRUE","Seuillage"),col=c("black","blue"),
       lwd=c(2,2),bg="white")
title(paste(" Erreur Seuil = ",signif(erreurseuil,3),sep=" "))
```



Assez peu adapté

Avec la base Haar

```
f=fstep
n=length(f)
Y = f + rnorm(n)
X=ComputeHaarMat(n)
Z=t(X)%*%Y
p=n
t=sqrt(2*log(p))
thetachap=Z*(abs(Z)>=t)
fhat=X%*%thetachap
erreurseuil=sum((fhat-f)^2)
plot(Y)
lines(f,lwd=3)
lines(fhat,col="blue",lwd=2)
legend("topright",c("TRUE","Seuillage"),col=c("black","blue"),
       lwd=c(2,2),bg="white")
title(paste(" Erreur Seuil = ",signif(erreurseuil,3),sep=" "))
```



beaucoup mieux.

Exercice 2 : Application du seuillage au débruitage de l'image `lena`

1)

```
F=lena
n=dim(F)[1]
Y = F + rnorm(n^2)
ranges=range(c(F))
cols=gray.colors(255,0,1)
par(mfrow=c(1,2),mai=c(0,0,0,0))
image(F, zlim=ranges,col=cols,xaxt="n",yaxt="n")
image(Y, zlim=ranges,col=cols,xaxt="n",yaxt="n")
```

2) avec la base Cosine

```r
error=mean((Y-F)^2)
X=ComputeCosMat(n)
Z=t(X)%*%Y%*%X
seuil=sqrt(2*log(n^2))
FhatCosine=X%*%(Z*(abs(Z)>seuil))%*%t(X)
error_seuillageCosine=mean((FhatCosine-F)^2)

c(error,error_seuillageCosine)
```

```
## [1] 0.9997996 0.2012169
```

```r
ranges=range(c(FhatCosine))
cols=gray.colors(255,0,1)
par(mfrow=c(1,3),mai=c(0,0,0,0))
image(F, zlim=ranges,col=cols,xaxt="n",yaxt="n")
image(Y, zlim=ranges,col=cols,xaxt="n",yaxt="n")
image(FhatCosine, zlim=ranges,col=cols,xaxt="n",yaxt="n")
```

avec la base de Haar, cela donne

```
error=mean((Y-F)^2)
X=ComputeHaarMat(n)
Z=t(X)%*%Y%*%X
seuil=sqrt(2*log(n^2))
FhatCosine=X%*%(Z*(abs(Z)>seuil))%*%t(X)
error_seuillageCosine=mean((FhatCosine-F)^2)

c(error,error_seuillageCosine)
```

```
## [1] 0.9997996 0.2268506
```

```
ranges=range(c(FhatCosine))
cols=gray.colors(255,0,1)
par(mfrow=c(1,3),mai=c(0,0,0,0))
image(F, zlim=ranges,col=cols,xaxt="n",yaxt="n")
image(Y, zlim=ranges,col=cols,xaxt="n",yaxt="n")
image(FhatCosine, zlim=ranges,col=cols,xaxt="n",yaxt="n")
```

Remarquons que le risque L^2 est bien meilleur avec le seuillage (Haar ou Cosine), alors que visuellement, on préférerait presque la version bruitée, car elle restitue les détails. Donc attention aux jugements de nature graphique !