(n=100)  
t=1:n  
a=10  
eps=runif(n,0,5)   
eps=rnorm(n,0,10)  
X=a+eps   
plot.ts(X)   
acf(X)  
acf(X^2)  
Y=X[2:n]-X[1:(n-1)]  
plot.ts(Y)   
acf(Y)  
Box.test(X, lag = 5, type = c("Box-Pierce"))  
Box.test(X, lag = 5, type = c("Ljung-Box"))  
Box.test(Y, lag = 5, type = c("Box-Pierce"))  
Box.test(Y, lag = 5, type = c("Ljung-Box"))  
Box.test(X, lag = 10, type = c("Box-Pierce"))  
Box.test(Y, lag = 10, type = c("Ljung-Box"))  
  
a=10  
eps=10\*rt(n,1)   
X=a+eps   
plot.ts(X)   
acf(X)  
Box.test(X, lag = 15, type = c("Box-Pierce"))  
Box.test(X, lag = 5, type = c("Ljung-Box"))  
  
t=1:n   
a=10\*sqrt(t+200/log(t+2))  
plot.ts(a)  
eps=10\*rnorm(n)   
X=a+eps   
plot.ts(X)   
acf(X)  
Box.test(X, lag = 5, type = c("Box-Pierce"))  
Box.test(X, lag = 5, type = c("Ljung-Box"))  
  
n=120  
t=1:n   
a=10+50\*cos(2\*pi\*t/12)  
#b=10\*sqrt(t+200/log(t+2))  
eps=10\*rnorm(n)   
X=a+eps   
plot.ts(a)  
plot.ts(X)   
acf(X)  
Box.test(X, lag = 5, type = c("Box-Pierce"))  
Box.test(X, lag = 5, type = c("Ljung-Box"))  
n=4000  
x=rnorm(n)   
r=acf(x)   
y=x[2:(n)]-1\*x[1:(n-1)]   
ts.plot(y)  
acf(y)  
Box.test(y, lag = 5, type = c("Box-Pierce"))  
Box.test(y, lag = 5, type = c("Ljung-Box"))  
Box.test(y, lag = 1, type = c("Ljung-Box"))  
Box.test(y, lag = 10, type = c("Ljung-Box"))  
  
SP500 <- read.csv("SP500\_9020.csv")  
V=SP500$Close  
ts.plot(V)  
nV=length(V)  
temps=1990.75+c(1:nV)\*30/nV  
plot(temps,V,'l')  
  
X=log(V[2:nV]/V[1:(nV-1)])  
plot(temps[2:nV],X,'l')  
acf(X)  
X=X[5500:6500]  
X=as.numeric(X)  
ts.plot(X)  
n=length(X)  
acf(X)  
acf(X^2)  
acf(abs(X))  
Box.test(X^2, lag = 5, type = c("Box-Pierce"))  
Box.test(X, lag = 10, type = c("Ljung-Box"))  
  
London <- read.delim("London.txt")  
View(London)  
X=London$jul  
Temp=ts(X,1659)  
Temp  
ts.plot(Temp)  
acf(Temp)  
acf(Temp^2)  
Box.test(Temp, lag = 5, type = c("Box-Pierce"))  
Box.test(Temp, lag = 5, type = c("Ljung-Box"))  
  
hstart <- read.table("hstart.txt")  
head(hstart)  
Dem=ts(hstart$V2,1966,frequency = 12)  
ts.plot(Dem)  
time(Dem)  
acf(Dem)  
Dem2=stl(Dem,s.window=12)  
plot(Dem2)  
ts.plot(Dem2$time[,"remainder"])  
plot(as.vector(Dem2$time[,"remainder"]))  
acf(Dem2$time[,"remainder"])  
Box.test(Dem2$time[,"remainder"], lag = 5, type = c("Ljung-Box"))  
Box.test(Dem2$time[,"remainder"], lag = 5, type = c("Box-Pierce"))  
  
London=London[,-1]  
TempL=as.matrix(London)  
TempL1659=matrix(t(TempL),ncol=1)  
TempL=ts(TempL1659,1659,frequency = 12)  
ts.plot(TempL)  
time(TempL)  
TempL1970=ts(TempL,1970,1998+11/12,frequency = 12)  
ts.plot(TempL1970)  
TempSTL=stl(TempL1970,s.window="periodic")  
plot(TempSTL)  
ts.plot(TempSTL$time[,"trend"])  
TempSTL=stl(TempL1970,s.window="periodic",t.window=250)  
plot(TempSTL)  
ts.plot(TempSTL$time[,"trend"])  
ts.plot(TempL1970)  
lines(TempSTL$time[,"trend"],col='red')  
ts.plot(TempL1970)  
lines(TempSTL$time[,"trend"]+TempSTL$time[,"seasonal"],col='red')  
ts.plot(TempSTL$time[,"remainder"])  
acf(TempSTL$time[,"remainder"])  
  
#### Estimation de tendance et saisonnalite ####  
################################################  
  
t=1:120  
tt=c(121:132)  
a=10\*sqrt(t+200/log(t+2))  
s=rep(c(10,-5,-12,7),30)  
aa=10\*sqrt(tt+200/log(tt+2))  
ss=rep(c(10,-5,-12,7),3)  
eps=10\*rnorm(120)   
X=a+s+eps   
plot.ts(c(a,aa), xlim=c(0,140),ylim=c(50,180))  
lines(t,X)  
  
#### Tendance seule par LSE ####  
X=a+eps  
plot.ts(c(a,aa), xlim=c(0,140),ylim=c(50,180))  
lines(t,X)  
reg1=lm(X~ t)   
lines(t,reg1$fit,col='red')   
summary(reg1)  
plot(reg1)   
names(reg1)   
reg1$coef   
summary(reg1)   
segments(t,reg1$fit,t,X)   
reg1$res   
ts.plot(reg1$res)  
win.graph()   
par(mfrow = c(2,2))   
plot(reg1, las = 1)   
R2=1-var(reg1$res)/var(X)   
R2  
  
new=data.frame(t=tt)  
pred1=predict.lm(reg1,newdata=new)  
lines(tt,pred1,col='red')  
  
reg2=lm(X~poly(t,2,raw=TRUE))   
lines(t,reg2$fit,col='blue')   
summary(reg2)   
pred2=predict.lm(reg2,newdata=new)  
lines(tt,pred2,col='blue')  
  
reg5=lm(X~poly(t,5,raw=TRUE))  
lines(t,reg5$fit,col='green')  
summary(reg5)  
pred5=predict.lm(reg5,newdata=new)  
lines(tt,pred5,col='green')  
  
library(MASS)   
long=length(t)   
kmax=10  
Z=matrix(nrow=long,ncol=kmax+1)   
for (i in 1:long)  
for (j in 1:(kmax+1))  
Z[i,j]=i^(j-1)  
ZZ=as.data.frame(Z)   
x.lm=lm(X~. ,data=ZZ)   
x.AIC=stepAIC(x.lm,k=2)   
x.BIC=stepAIC(x.lm,k=log(long),trace='true')  
  
regBIC=lm(X~poly(t,4),raw=TRUE)   
lines(t,regBIC$fit,col='brown')  
summary(regBIC)  
predBIC=predict.lm(regBIC,newdata=new)  
lines(tt,predBIC,col='brown')  
  
#### Estimation de saisonnalite seule par LSE ####  
X=s+eps  
X=ts(X,1,frequency=4)  
X  
ts.plot(X)  
monthplot(X)  
  
long=length(t)   
T=4  
Z=matrix(nrow=long,ncol=(T-1))  
for (i in 1:long)  
 for (j in 1:(T-1))  
 Z[i,j]=floor(abs(cos(pi\*(i-j)/T)))  
Part=floor(long/T)  
for (i in 1:Part)  
 for (j in 1:(T-1))  
 {ii=i\*(T)  
 Z[ii,j]=-1}  
Z  
regs=lm(X~Z)  
summary(regs)  
regs$coefficients  
cT=-sum(regs$coefficients[2:4])  
cT  
  
#### Estimation de tendance polynomiale et saisonnalite par LSE ####  
X=a+s+eps   
ts.plot(X)  
long=length(t)   
T=4  
kmax=5  
Zs=matrix(nrow=long,ncol=(T-1))  
for (i in 1:long)  
 for (j in 1:(T-1))  
 Zs[i,j]=floor(abs(cos(pi\*(i-j)/T)))  
Part=floor(long/T)  
for (i in 1:Part)  
 for (j in 1:(T-1))  
 {ii=i\*(T)  
 Zs[ii,j]=-1}  
Zt=matrix(nrow=long,ncol=kmax+1)   
for (i in 1:long)  
 for (j in 1:(kmax+1))  
 Zt[i,j]=i^(j-1)  
Z=cbind(Zs,Zt)  
Z=as.data.frame(Z)   
x.lm=lm(X~.-1 ,data=Z)   
summary(x.lm)  
#x.lm=lm(X~.-1-V1-V2-V3,data=Z)   
#summary(x.lm)  
x.AIC=stepAIC(x.lm,k=2)   
x.BIC=stepAIC(x.lm,k=log(long))  
plot.ts(c(a,aa), xlim=c(0,140),ylim=c(50,180))  
lines(t,X)  
lines(x.lm$fit,col='red')  
ts.plot(x.lm$res)  
acf(x.lm$res)  
Box.test(x.lm$res,lag=10)  
  
Zsn=matrix(nrow=length(tt),ncol=(T-1)) # Pour la prediction  
for (i in c(1:length(tt)))  
 for (j in 1:(T-1))  
 Zsn[i,j]=floor(abs(cos(pi\*(i-j)/T)))  
Part=floor(length(tt)/T)  
for (i in 1:Part)  
 for (j in 1:(T-1))  
 {ii=i\*(T)  
 Zsn[ii,j]=-1}  
  
Ztn=matrix(nrow=length(tt),ncol=kmax+1)   
for (i in c(1:length(tt)))  
 for (j in 1:(kmax+1))  
 Ztn[i,j]=(120+i)^(j-1)  
Zn=cbind(Zsn,Ztn)  
Zn=as.data.frame(Zn)   
  
pred1=predict.lm(x.lm,newdata=Zn)  
plot.ts(c(a,aa), xlim=c(0,140),ylim=c(50,180))  
lines(t,X)  
lines(x.lm$fit,col='red')  
lines(tt,pred1,col='blue')  
  
#### Estimation de tendance polynomiale, saisonnalite avec variables exogènes par LSE ####  
View(ConsoElec\_2022)  
attach(ConsoElec\_2022)  
ts.plot(Cons)  
ts.plot(Tm)  
ts.plot(TM)  
Consa=Cons[1:357]  
Tma=Tm[1:357]  
TMa=TM[1:357]  
Reg1=lm(Consa~Tma+TMa)  
summary(Reg1)  
  
new=data.frame(Tma=Tm[358:364],TMa=TM[358:364])  
pred1=predict.lm(Reg1,newdata=new)  
  
ts.plot(Cons)  
lines(Reg1$fit,col='red')  
lines(c(358:364),pred1,col='blue')  
RMSE1=(mean((pred1-Cons[358:364])^2))^0.5  
  
long=length(Consa)   
T=7  
Zsa=matrix(nrow=long,ncol=(T-1))  
for (i in 1:long)  
 for (j in 1:(T-1))  
 Zsa[i,j]=floor(abs(cos(pi\*(i-j)/T)))  
Part=floor(long/T)  
for (i in 1:Part)  
 for (j in 1:(T-1))  
 {ii=i\*(T)  
 Zsa[ii,j]=-1}  
Za=cbind(Zsa,Tma,TMa)  
 Za=as.data.frame(Za)   
 Reg2=lm(Consa~.,data=Za)   
summary(Reg2)  
x.AIC=stepAIC(Reg2,k=2)   
x.BIC=stepAIC(Reg2,k=log(long))  
  
tt=c(358:364)  
Zsn=matrix(nrow=length(tt),ncol=(T-1)) # Pour la prediction  
for (i in c(1:length(tt)))  
 for (j in 1:(T-1))  
 Zsn[i,j]=floor(abs(cos(pi\*(i-j)/T)))  
Part=floor(length(tt)/T)  
for (i in 1:Part)  
 for (j in 1:(T-1))  
 {ii=i\*(T)  
 Zsn[ii,j]=-1}  
Zn=as.data.frame(cbind(Zsn,Tma=Tm[tt],TMa=TM[tt]))  
pred2=predict.lm(Reg2,newdata=Zn)  
lines(Reg2$fit,col='cyan')  
lines(c(358:364),pred2,col='cyan')  
RMSE2=(mean((pred2-Cons[358:364])^2))^0.5  
  
#### Estimation non-parametrique de tendance et saisonnalite ####  
#################################################################  
  
#### Moyenne mobile ####  
t=1:120  
tt=c(121:132)  
a=10\*sqrt(t+200/log(t+2))  
s=rep(c(10,-5,-12,7),30)  
aa=10\*sqrt(tt+200/log(tt+2))  
ss=rep(c(10,-5,-12,7),3)  
eps=10\*rnorm(120)   
X=a+s+eps   
plot.ts(c(a,aa), xlim=c(0,140),ylim=c(50,180))  
lines(t,X)  
lines(tt,aa+ss,col='blue')  
  
X=a+eps  
ts.plot(X)  
plot.new()  
par(mfrow=c(2,2))  
for (m in c(3,8,33,50))  
{if (m%%2==1)  
am=filter(X, rep(1/m,m))  
if (m%%2==0)  
 am=filter(X, c(1/(2\*m),rep(1/m,m-1),1/(2\*m)))  
ts.plot(X)  
lines(am,col='red')  
}  
  
#### Estimateur a noyau ####  
plot.new()  
par(mfrow=c(1,1))  
ks1=ksmooth(t,X, "normal", bandwidth=4)  
ks1   
plot.ts(X)   
lines(ks1,col=2)  
ks2=ksmooth(t,X, "normal", bandwidth=30)  
lines(ks2,col='blue')   
ks2=ksmooth(t,X, "normal", bandwidth=15)  
lines(ks2,col='cyan')   
library(lokern)   
kerada=glkerns(t,X)   
lines(kerada,col='yellow')  
kerada$bandwidth  
plot(kerada)  
  
#### Estimateur par regression locale LOESS LOWESS ####  
plot.new()  
par(mfrow=c(1,1))  
plot.ts(c(a,aa), xlim=c(0,140),ylim=c(50,180))  
#points(tt,aa)  
lines(t,X)  
regl2=loess(X~t,span =0.9, t.degree=0)  
regl2=loess(X~t)  
summary(regl2)  
names(regl2)   
lo2=ts(regl2$fitted)  
plot(t,regl2$residuals,'l')  
acf(regl2$residuals)  
plot.ts(c(a,aa), xlim=c(0,140),ylim=c(50,180))  
lines(t,lo2,col='blue')   
  
R2=1-var(regl2$res)/var(X)  
R2  
  
regl=loess(X~t) # d?faut span=0.75  
summary(regl)  
lo=ts(regl$fitted)  
plot.ts(c(a,aa), xlim=c(0,140),ylim=c(50,180))  
lines(t,lo,col='blue')   
  
regl1=loess(X~t,span=0.6)   
names(regl1)   
ts.plot(X)   
lines(t,a)  
lines(t,regl1$fit,col='red')   
points(t,X)   
title("Resultats de loess(X t)")  
R2=1-var(regl1$res)/var(X)  
R2  
  
library(fANCOVA,quietly = T)  
regloa <- loess.as(t,X, degree = 0, criterion = c("aicc","gcv")[2], user.span = NULL, family ="gaussian",plot="TRUE")  
summary(regloa)  
names(regloa)  
regloa$pars$span  
  
X=a+s+eps  
X=ts(X,frequency=4)  
ts.plot(X)  
ma=decompose(X)  
plot(ma)  
ma=decompose(X, filter=c(1,1,1,1)/4)  
plot(ma)  
ma=decompose(X, filter=c(1/8,1/4,1/4,1/4,1/8))  
plot(ma)  
ma=decompose(X, filter=c(1/16,1/8,1/8,1/8,1/8,1/8,1/8,1/8,1/16))  
plot(ma)  
ma=decompose(X, filter=c(1/24,rep(1/12,11),1/24))  
plot(ma)  
ma=decompose(X, filter=c(1/32,rep(1/16,15),1/32))  
plot(ma)  
lo=stl(X,s.window="periodic")  
plot(lo)  
lo=stl(X,s.window="periodic",t.window=11)  
plot(lo)  
  
######################################  
#### Exercice sur la serie hstart ####  
  
hstart <- read.table("hstart.txt")  
Dem=ts(hstart$V2,1966,frequency = 12)  
ts.plot(Dem)  
t=time(Dem)  
  
#### Questions ####  
# 1. Estimer la tendance et la saisonnalite suivant   
# la methode de regression par moindres carres et   
# predire les valeurs de l'annee 1975  
# 2. Estimer la tendance et la saisonnalite suivant   
# les 3 methodes non-parametriques vues en choisissant le   
# le parametre de lissage   
  
  
# 1.   
Z=0  
ZZ=0  
long=length(Dem)   
T=12  
Zs=matrix(nrow=long,ncol=(T-1))  
for (i in 1:long)  
 for (j in 1:(T-1))  
 Zs[i,j]=floor(abs(cos(pi\*(i-j)/T)))  
Part=floor(long/T)  
for (i in 1:Part)  
 for (j in 1:(T-1))  
 {ii=i\*(T)  
 Zs[ii,j]=-1}  
kmax=4  
Zt=matrix(nrow=long,ncol=kmax+1)   
for (i in 1:long)  
 for (j in 1:(kmax+1))  
 Zt[i,j]=i^(j-1)  
Z=cbind(Zs,Zt)  
Z=as.data.frame(Z)   
x.lm=lm(Dem~.-1 ,data=Z)   
summary(x.lm)  
Dem2=as.numeric(Dem)  
t2=as.numeric(t)  
plot(t2,x.lm$fitted.values,'l',col='red')  
lines(t2,Dem2)  
  
library(MASS)   
x.BIC=stepAIC(x.lm,k=log(length(t)))  
BIC(x.lm)  
BIC(lm(Dem~.-1-V18 ,data=Z))  
acf(x.lm$residuals)  
Box.test(x.lm$residuals, lag = 8, type = c("Ljung-Box"))  
  
a1=0  
s1=as.matrix(Z[,1:11])%\*%x.lm$coefficients[1:11]  
plot(t2,s1,'l')  
a1=t(as.matrix(Z[,c(12:(12+kmax))]))%\*%as.numeric(as.matrix(x.lm$coefficients[12:(12+kmax)]))  
plot(t2,a1,'l')  
  
ZZs=matrix(nrow=12,ncol=(T-1))  
for (i in 1:12)  
 for (j in 1:(T-1))  
 ZZs[i,j]=floor(abs(cos(pi\*(i-j)/T)))  
Part=floor(12/T)  
for (i in 1:Part)  
 for (j in 1:(T-1))  
 {ii=i\*(T)  
 ZZs[ii,j]=-1}  
ZZt=matrix(nrow=12,ncol=kmax+1)   
for (i in 1:12)  
 for (j in 1:(kmax+1))  
 ZZt[i,j]=(long+i)^(j-1)  
ZZ=cbind(ZZs,ZZt)  
ZZ=as.data.frame(ZZ)   
  
pred1=predict.lm(x.lm,newdata=ZZ)  
tt=1975+c(0:11)/12  
t3=c(t2,tt)  
Dem3=c(x.lm$fitted.values,pred1)  
plot.new()  
plot(t2,Dem2,'l',col='black',xlim=c(1965,1976))  
lines(t3,Dem3,'l',col='blue')  
lines(t2,Dem2,col='black')  
  
# 2. Moyenne mobile  
m=12  
am=filter(Dem2, c(1/(2\*m),rep(1/m,m-1),1/(2\*m)))  
ts.plot(Dem)  
lines(t2,am,col='red')  
  
m=24  
am=filter(Dem2, c(1/(2\*m),rep(1/m,m-1),1/(2\*m)))  
ts.plot(Dem)  
lines(t2,am,col='red')  
  
DemMA=decompose(Dem)  
names(DemMA)  
DemMA$trend  
plot(DemMA)  
sMA=DemMA$seasonal  
ts.plot(sMA)  
lines(t2,s1,col='red')  
max(abs(sMA-s1))  
acf(DemMA$random[7:102])  
Box.test(DemMA$random[7:102], lag = 5, type = c("Ljung-Box"))  
  
  
# 2. Noyau   
DemDes=Dem2-s1  
DemDes=ts(DemDes,1966,frequency = 12)  
plot.new()  
plot.ts(DemDes)   
lines(t2,a1,col='blue')  
ks1=ksmooth(t2,DemDes, "normal", bandwidth=4)  
ks1   
lines(ks1,col='red')  
ks2=ksmooth(t2,Dem, "normal", bandwidth=3)  
lines(ks2,col='cyan')   
library(lokern)   
kerada=glkerns(t2,DemDes)   
lines(kerada,col='yellow')  
kerada$bandwidth  
plot(kerada)  
  
# 2. LOESS et STL   
  
library(fANCOVA,quietly = T)  
regloa <- loess.as(t2,DemDes, degree = 2, criterion = c("aicc","gcv")[2], user.span = NULL, family ="gaussian",plot="TRUE")  
summary(regloa)  
names(regloa)  
regloa$pars$span  
  
DemSTL=stl(Dem,s.window=12,t.window=17)  
plot(DemSTL)  
lines(t2,DemSTL$time[,"trend"],col='red')  
acf(DemSTL$time[,"remainder"])  
Box.test(DemSTL$time[,"remainder"], lag = 5, type = c("Ljung-Box"))  
s2=DemSTL$time[,"seasonal"]  
ts.plot(s2)  
lines(t2,s1,col='red')  
max(abs(s2-s1))  
  
  
###########################################  
#### Exemple des temp?ratures globales ####  
Temp=read.table("TGlobe1.txt",header=TRUE)  
t=Temp$An  
X=Temp$Temp  
  
plot(t,X,'l')  
reg1=lm(X~ t)   
abline(reg1)   
names(reg1)   
reg1$coef   
summary(reg1)   
segments(t,reg1$fit,t,X)   
reg1$res   
win.graph()   
par(mfrow = c(2,2))   
plot(reg1, las = 1)   
R2=1-var(reg1$res)/var(X)   
R2  
  
reg2=lm(X~poly(t,2,raw=TRUE ))   
plot(t,X,'l')   
lines(t,reg2$fit,col='red')   
summary(reg2)   
segments(t,reg2$fit,t,X)   
  
reg9=lm(X~poly(t,9,raw=TRUE) )   
plot(t,X,'l')   
lines(t,reg9$fit,col='blue')  
summary(reg9)  
  
library(MASS)   
long=length(t)   
kmax=10   
Z=matrix(nrow=long,ncol=kmax+1)   
for (i in 1:long)  
 for (j in 1:(kmax+1))  
 Z[i,j]=i^(j-1)  
ZZ=as.data.frame(Z)   
x.lm=lm(X~. ,data=ZZ)   
x.AIC=stepAIC(x.lm,k=2)   
x.BIC=stepAIC(x.lm,k=log(long))   
  
  
regl2=lowess(t,X)  
names(regl2)   
summary(regl2)  
low2=ts(regl2$y)  
ts.plot(X,low2)   
regl3=lowess(t,X,f=0.1)  
low3=ts(regl3$y)  
ts.plot(X,low3)   
ts.plot(low2,low3)  
  
reglo=loess(X~t,span = 0.5)   
summary(reglo)  
names(reglo)   
plot(t,X,'l')  
lines(t,reglo$fit,col='red')   
points(t,X)   
title("Resultats de loess(X t)")  
R2=1-var(reglo$res)/var(X)  
R2  
  
reglo1=loess(X~t,span = 0.22)   
names(reglol)   
plot(t,X,'l')  
plot(t,reglo1$fit,'l',col='red')   
lines(t,X)   
points(t,X)   
title("Resultats de loess(X t)")  
R2=1-var(reglo1$res)/var(X)  
R2  
acf(reglo1$res)  
  
library(fANCOVA,quietly = T)  
regloa <- loess.as(t,X, degree = 2, criterion = c("aicc","gcv")[1], user.span = NULL, family ="gaussian",plot="TRUE")  
summary(regloa)  
names(regloa)  
regloa$pars$span  
  
ks1=ksmooth(t,X, "normal", bandwidth=2)  
ks1   
plot(t,X,'l')   
lines(ks1,col=2)  
ks2=ksmooth(t,X, "normal", bandwidth=10)  
lines(ks2,col=3)   
library(lokern)   
kerada=glkerns(t,X)   
plot(kerada)  
  
  
London <- read.table("London.txt")  
head(London)  
attach(London)  
TempLondon=data.frame(jan,feb,mar,apr,may,jun,jul,aou,sep,oct,nov,dec)  
TempLond=0  
for (i in c(1:340))  
 for (j in c(1:12))  
 TempLond=c(TempLond,TempLondon[i,j])  
TemLondon=TempLond[2:4081]  
TempLon=ts(TemLondon,frequency = 12,1659)  
TempLon  
ts.plot(TempLon)  
t=time(TempLon)  
  
  
monthplot(TempLon-mean(TempLon))  
s=stl(TempLon,s.win="perio")  
plot(s)  
monthplot(s)  
names(s)  
s$time[,"seasonal"]  
ts.plot(s$time[,"seasonal"])  
s$time[,"trend"]  
ts.plot(s$time[,"trend"])  
s$time[,"remainder"]  
ts.plot(s$time[,"remainder"])  
acf(s$time[,"remainder"])  
Box.test(s$time[,"remainder"], lag = 10, type = c("Ljung-Box"))  
  
s2=stl(TempLon,s.win="perio",t.win=600)  
monthplot(s2)  
plot(s2)  
ts.plot(s2$time[,"trend"])  
acf(s2$time[,"remainder"])  
Box.test(s2$time[,"remainder"], lag = 10, type = c("Ljung-Box"))  
  
s3=stl(TempLon,s.win=600,t.win=600)  
monthplot(s3)  
ts.plot(s3$time[,"trend"])  
  
s4=stl(TempLon,s.win="perio",t.win=1200)  
monthplot(s4)  
ts.plot(s4$time[,"trend"])  
acf(s4$time[,"remainder"])  
  
s5=stl(TempLon,s.win="perio",t.win=300)  
monthplot(s5)  
ts.plot(s5$time[,"trend"])  
acf(s5$time[,"remainder"])  
  
s6=stl(TempLon,s.win="perio")  
plot(s6)  
  
library(fANCOVA,quietly = T)  
slo3 <- loess.as(t,TempLon, degree = 2, criterion = c("aicc","gcv")[2], user.span = NULL, family ="gaussian",plot="TRUE")  
summary(slo3)  
names(slo3)  
slo3$pars$span  
plot.new()  
ts.plot(TempLon)  
plot(t,slo3$fitted,col='red')  
  
  
########################  
#### Processus ARMA ####  
  
n=1000000  
m=100  
eps=runif(n+m,-2,2)  
eps=rt(n+m,1)  
Y=0; X=rep(0,n)  
a1=-0.7  
b1=-0.5  
for (j in c(1:(n+m-1)))  
Y[j+1]=-a1\*Y[j]+eps[j+1]+b1\*eps[j]  
Y  
X=Y[(m+1):(n+m)]  
ts.plot(X)  
acf(X)  
  
a1=c(0.6,0.1); b1=c(0.3,-0.6)  
X=arima.sim(n+m,model=list(ar=a1,ma=b1))  
X=X[(m+1):(n+m)]  
plot.ts(X)   
acf(X)  
  
b1=c(-0.2)  
X=arima.sim(1000000,model=list(ma=b1))  
plot.ts(X)   
acf(X)  
  
#########################  
#### Processus GARCH ####  
  
n=10000  
m=100  
eps=rnorm(n+m,0,1)  
Y=0; X=rep(0,n)  
a0=2  
a1=0.6  
b1=0.3  
sigma=0  
for (j in c(1:(n+m-1)))  
 {sigma[j+1]=sqrt(a0+a1\*Y[j]^2+b1\*sigma[j]^2)  
 Y[j+1]=sigma[j+1]\*eps[j+1]}  
Y  
X=Y[(m+1):(n+m)]  
ts.plot(X)  
acf(X)  
acf(X^2)  
acf(abs(X))  
  
library(fGarch)  
n=1000  
spec = garchSpec(model = list(omega = 1, alpha = c(0.2), beta = 0.75))  
X=garchSim(spec,n+100)  
X=X[101:1100]  
ts.plot(X)  
  
##########################  
#### Processus APARCH ####  
  
### un APARCH(delta,2,1)$   
n=1000  
spec = garchSpec(model = list(omega = 2, alpha = c(0.10, 0.05), gamma = c(0.5, 0.8),   
 beta = 0.8, delta = 1.5), cond.dist = "snorm")   
X=garchSim(spec,n)  
ts.plot(X)  
acf(X)  
  
##########################  
#### Processus FARIMA ####  
  
### FARIMA(0,d,0)$   
n=1000  
  
d=0.45;  
sigma=4  
N=20000  
x1= ((0:N)^(d-1)/gamma(d\*rep(1,N+1)))   
y1 = (gamma(d\*rep(1,N+1)+(0:N))/gamma(1:(N+1))/gamma(d\*rep(1,N+1)))   
m1=sum(is.na(y1>0)==FALSE)-1  
y=c(y1[1:m1], x1[(m1+1):N])  
eps=rnorm(n+N,0,sigma)  
X=filter(eps,y)  
X=X[is.na(X)==FALSE]  
ts.plot(X)  
acf(X)  
  
par(mfrow = c(2,2))   
Nil=read.table("Nil.txt", header=TRUE)  
Nil  
plot(Nil$Annee,Nil$Niveau,'l')  
acf(Nil$Niveau)  
  
n=560  
d=0.42;  
sigma=sd(Nil$Niveau)  
N=20000  
x1= ((0:N)^(d-1)/gamma(d\*rep(1,N+1)))   
y1 = (gamma(d\*rep(1,N+1)+(0:N))/gamma(1:(N+1))/gamma(d\*rep(1,N+1)))   
m1=sum(is.na(y1>0)==FALSE)-1  
y=c(y1[1:m1], x1[(m1+1):N])  
eps=rnorm(n+N,0,sigma)  
eps=sigma\*rt(n+N,4)  
X=filter(eps,y)  
X=mean(Nil$Niveau)+X[is.na(X)==FALSE]  
plot(tnil, X, 'l')  
acf(X)  
tnil=722+c(0:560)  
  
  
par(mfrow = c(1,1))   
library(arfima)  
X=arfima.sim(560, model = list(dfrac = .49))  
ts.plot(X)  
acf(X)  
X=arfima.sim(1000, model = list(phi=0.2, theta=-0.4, dfrac = 0.4))  
ts.plot(X)  
  
###############################################  
#### Estimation param?trique ARMA et GARCH ####  
  
a1=c(0.3,0.5); b1=c(-0.6)  
n=1000  
m=100  
Y=arima.sim(n+m,model=list(ar=a1,ma=b1))  
X=Y[(m+1):(n+m)]  
ts.plot(X)  
  
library(forecast)  
  
ord=c(2,0,1)   
EstiARMA=arima(X,ord)   
EstiARMA  
  
acf(X)  
Box.test(X, lag = 5, type = c("Ljung-Box"))  
ts.plot(EstiARMA$res)  
acf(EstiARMA$res^2)  
Porte=Box.test(EstiARMA$res, lag = 10, type = c("Ljung-Box"))  
Porte  
qchisq(0.95,6)  
  
  
#############################  
### EStimation pour GARCH ###  
  
library(fGarch)  
n=1000  
spec = garchSpec(model = list(omega = 0.0005, alpha = c(0.18), beta = 0.8))  
X=garchSim(spec,n)  
ts.plot(X)  
acf(X)  
acf(X^2)  
acf(abs(X))  
  
FitX1=garchFit(~ garch(1,1), data = X, trace = FALSE)   
coef(FitX1)  
summary(FitX1)  
  
########################################  
#### Estimation param?trique APARCH ####  
  
spec = garchSpec(model = list(omega = 2, alpha = c(0.1), gamma = c(0.8),   
 beta = 0.4, delta = 1.5), cond.dist = "snorm")   
n=10000  
X=garchSim(spec,n)  
ts.plot(X)  
FitA=garchFit(~ aparch(1,1), data = X, trace = FALSE)   
coef(FitA)  
summary(FitA)  
FitA=garchFit(~ aparch(2,1), data = X, trace = FALSE)   
coef(FitA)  
summary(FitA)  
FitA=garchFit(~ arma(1,1)+garch(2,1), data = X, trace = FALSE)   
coef(FitA)  
summary(FitA)  
  
  
########################################  
#### Estimation param?trique FARIMA ####  
  
library(arfima)  
X=arfima.sim(20000, model = list(phi=0.2, theta=-0.4, dfrac = 0.3))  
ts.plot(X)  
acf(X)  
fit=arfima(X, order = c(1, 0, 1))  
summary(fit)  
  
######################################  
#### S?lection de mod?les et test ####  
  
  
###########################################  
#### Selection de mod?les et test ARMA ####  
  
library(forecast)  
m=100  
n=2000  
X=arima.sim(n+m,model=list(ar=c(-0.3,0.4),ma=.7))  
#spec = garchSpec(model = list(omega = 0.000005, alpha = c(0.18), beta = 0.75))  
#X=garchSim(spec,n+m)  
X=X[(m+1):(n+m)]  
ts.plot(X)  
fitaic=auto.arima(X,max.p=5,max.q=5,ic="aic")  
fit=auto.arima(X,max.p=5,max.q=5,ic="bic")  
fit  
fit$residuals  
Box.test(fit$residuals^2, lag = 10)  
Box.test(fitaic$residuals^2, lag = 10)  
predict(fit,n.ahead=20)  
  
  
X=arima.sim(10000,model=list(ar=c(0.8),ma=-.5))  
plot.ts(X)   
acf(X)  
ord=c(0,0,4)   
EstiARMA=arima(X,ord)   
EstiARMA  
Box.test(X, lag = 10, type = c("Ljung-Box"))  
Porte2=Box.test(EstiARMA$res^2, lag = 10, type = c("Ljung-Box"))  
Porte2$stat  
Porte2$p.value  
qchisq(0.95,10)  
  
  
library(tseries)  
data(LakeHuron)  
LakeHuron=LakeHuron-mean(LakeHuron)  
plot.ts(LakeHuron)  
acf(LakeHuron)  
acf(LakeHuron)[1]  
ord=c(1,0,0)   
EstiHuron=arima(LakeHuron,ord, method="CSS-ML")   
EstiHuron  
PorteHuron=Box.test(EstiHuron$res^2, lag = 5, type = c("Ljung-Box"))  
PorteHuron$stat  
PorteHuron$p.value  
qchisq(0.95,5)  
auto.arima(LakeHuron,max.p=5,max.q=5,ic="bic")  
  
predict(EstiHuron,n.ahead=5)  
  
  
############################################  
#### Selection de mod?les et test GARCH ####  
  
library(fGarch)  
n=1000  
spec = garchSpec(model = list(omega = 1, alpha = c(0.3), beta = 0.66000))  
X=garchSim(spec,n)  
ts.plot(X)  
acf(X)  
acf(X^2)  
acf(abs(X))  
  
FitX=garchFit(~ garch(1,1), data = X, trace = FALSE)   
summary(FitX)  
predict(FitX,n.ahead=5)  
FitXAG=garchFit(~ arma(1,1)+garch(2,1), data = X, trace = FALSE)   
predict(FitXAG,n.ahead=5)  
  
  
FitX1=garchFit(~ garch(1,1), data = X, trace = FALSE)   
coef(FitX1)  
BIC1=FitX1@fit$ics[2]; BIC1  
FitX2=garchFit(~ garch(2,1), data = X, trace = FALSE)  
BIC2=FitX2@fit$ics[2]; BIC2  
FitX3=garchFit(~ garch(3,3), data = X, trace = FALSE)  
BIC3=FitX3@fit$ics[2]; BIC3  
  
spec = garchSpec(model = list(omega = 1, alpha = c(0.4,0.1,0.3), beta = 0.16000))  
X=garchSim(spec,n)  
  
FitBIC=GARCH.fit(X,pBIC,qBIC)   
lag=10  
GARCH.test(X,FitBIC$par,pBIC,qBIC,lag) # Statistique de test et p-value pour le test portemanteau  
  
#####################################  
#### Pr?diction avec Holt-Winter ####  
  
n=100  
t=1:n   
a=10\*sqrt(t+200/log(t+2))  
eps=10\*rnorm(n)   
X=a+eps   
X=ts(X)  
plot.ts(X, xlim=c(0,115),ylim=c(60,150))  
Holt=HoltWinters(X,alpha=0.3,beta=FALSE, gamma=FALSE)  
Holt  
names(Holt)  
lines(Holt$fitted[,1],col='red')  
predict(Holt,n.ahead = 10)  
lines(predict(Holt,n.ahead = 10),col='blue')  
  
plot.ts(X, xlim=c(0,115),ylim=c(60,150))   
Holt=HoltWinters(X,beta=FALSE, gamma=FALSE)  
Holt  
lines(Holt$fitted[,1],col='red')  
predict(Holt,n.ahead = 10)  
lines(predict(Holt,n.ahead = 10),col='blue')  
  
plot.ts(X, xlim=c(0,115),ylim=c(60,150))   
Holt=HoltWinters(X, gamma=FALSE)  
Holt  
lines(Holt$fitted[,1],col='red')  
predict(Holt,n.ahead = 10)  
lines(predict(Holt,n.ahead = 10),col='blue')  
  
################################  
#### Exercice de pr?diction ####  
  
London <- read.delim("C:/Users/bardet/Dropbox/Donnees/London.txt")  
attach(London)  
TempLondon=data.frame(jan,feb,mar,apr,may,jun,jul,aou,sep,oct,nov,dec)  
TempLond=0  
for (i in c(1:340))  
 for (j in c(1:12))  
 TempLond=c(TempLond,TempLondon[i,j])  
TemLondon=TempLond[2:4081]  
TempLon=ts(TemLondon,frequency = 12,1659)  
TempLon  
ts.plot(TempLon)  
  
  
HoltLondon=HoltWinters(TempLon)  
HoltLondon  
lines(HoltLondon$fitted[,1],col='red')  
PredLond=predict(HoltLondon,n.ahead = 12)  
TempLon1978=ts(TempLon[3829:4068],frequency=12,1978, 1997+11/12)  
ts.plot(TempLon1978,xlim=c(1977,2000),ylim=c(-2,20))  
  
##################################################  
#### On veut pr?dire les temperatures de 1998 ####  
  
HoltLondon=HoltWinters(TempLon1978)  
HoltLondon  
lines(HoltLondon$fitted[,1],col='red')  
PredLond1=predict(HoltLondon,n.ahead = 12)  
lines(PredLond1,col='blue')  
  
tt=50  
BLond=c()  
for (tt in (20+5\*c(0:20)))  
{stlLond=stl(TempLon1978,s.window="periodic",t.window=tt)  
plot(stlLond)  
acf(stlLond$time.series[,"remainder"])  
BoxLond=Box.test(stlLond$time.series[,"remainder"],lag=8)  
BLond=c(BLond,BoxLond$statistic)}  
BLond  
which.min(BLond)  
  
stlLond=stl(TempLon1978,s.window="periodic",t.window=105)  
sea=stlLond$time.series[,"seasonal"]  
ts.plot(sea)  
tre=stlLond$time.series[,"trend"]  
ts.plot(tre)  
acf(stlLond$time.series[,"remainder"])  
Box.test(stlLond$time.series[,"remainder"],lag=5)  
sea98=sea[1:12]  
detLond=TempLon1978-sea  
time(TempLon1978)  
TempLon1994=as.numeric(TempLon1978[(16\*12+1):(20\*12)])  
time94=1994+c(1:48)/12  
reg94=lm(TempLon1994~time94)  
new98=data.frame(time94=1998+c(1:12)/12)  
tend98=predict.lm(reg94,newdata = new98)  
temp98=tend98+sea98  
temp98=ts(temp98,frequency = 12,1998)  
  
LondDes=TempLon1978-ts(sea,frequency = 12,1978)  
ts.plot(LondDes)  
HoltLondon2=HoltWinters(LondDes,gamma=FALSE)  
HoltLondon2  
lines(HoltLondon2$fitted[,1],col='red')  
PredLond2=predict(HoltLondon2,n.ahead = 12)  
lines(PredLond2,col='blue')  
Predsea=ts(sea[1:12],frequency=12,1998)  
Predsea  
PredLond2=PredLond2+Predsea  
  
TempLon1998=ts(TempLon[4069:4080],frequency=12,1998, 1998+11/12)  
ts.plot(TempLon1998,xlim=c(1997.9,1999.1),ylim=c(3,18))  
lines(PredLond1,col='blue')  
lines(PredLond2,col='red')  
lines(temp98,col='cyan')