STAT 661: Project

Chapter 10: Nonconjugate priors and Metropolis-Hastings algorithms

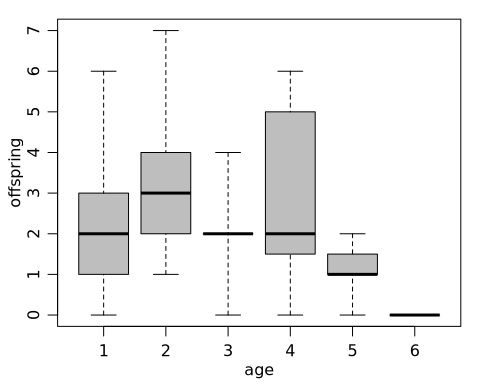
Geovani Bautista, Simon Lee, Shen Qu

Dec, 2019

## 10.1 Generalized linear models

Example: Song sparrow reproductive success

#### Sparrow data  
load("sparrows.RData")   
fledged<-sparrows[,1] ; age<-sparrows[,2] ; age2<-age^2  
  
  
  
#### Figure 10.1   
# pdf("fig10\_1.pdf",family="Times",height=3.5,width=7)  
par(mar=c(3,3,1,1),mgp=c(1.75,.75,0))  
plot(fledged~as.factor(age),range=0,xlab="age",ylab="offspring",  
 col="gray")



summary(glm(fledged~age+age2,family="poisson"))

##   
## Call:  
## glm(formula = fledged ~ age + age2, family = "poisson")  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.4650 -0.6355 -0.2298 0.4937 2.0429   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 0.27662 0.44219 0.626 0.5316   
## age 0.68174 0.33850 2.014 0.0440 \*  
## age2 -0.13451 0.05786 -2.325 0.0201 \*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for poisson family taken to be 1)  
##   
## Null deviance: 76.081 on 51 degrees of freedom  
## Residual deviance: 67.837 on 49 degrees of freedom  
## AIC: 198.78  
##   
## Number of Fisher Scoring iterations: 5

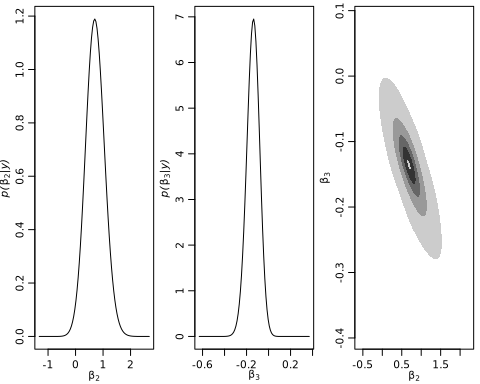
#### Grid-based posterior approximation  
p<-3  
beta0<-rep(0,p)  
S0<-diag( rep(100,3))  
gs<-100  
LPB<-array(0,dim=rep(gs,p))  
  
beta1<-seq(.27-1.75,.27+1.75,length=gs)  
beta2<-seq(.68-1.5,.68+1.5,length=gs)  
beta3<-seq(-.13-.25,-.13+.25,length=gs)  
  
beta1<-seq(.27-2.5,.27+2.5,length=gs)  
beta2<-seq(.68-2,.68+2,length=gs)  
beta3<-seq(-.13-.5,-.13+.5,length=gs)  
  
  
  
for(i in 1:gs) { for(j in 1:gs) { for(k in 1:gs) {  
 theta<-beta1[i]+beta2[j]\*age+beta3[k]\*age^2  
 LPB[i,j,k]<-dnorm(beta1[i],beta0[1],sqrt(S0[1,1]),log=TRUE) +  
 dnorm(beta2[j],beta0[2],sqrt(S0[2,2]),log=TRUE) +  
 dnorm(beta3[k],beta0[3],sqrt(S0[3,3]),log=TRUE) +  
 sum( dpois(fledged,exp(theta),log=TRUE ) )  
 }}   
 }  
cat(i,"\n")

## 100

PB<-exp( LPB - max(LPB) )  
PB<-PB/sum(PB)  
  
PB1<-apply(PB,1,sum)  
PB2<-apply(PB,2,sum)  
PB3<-apply(PB,3,sum)  
PB23<-apply(PB,c(2,3),sum)  
  
  
## Simulation from grid approximation  
S<-50000  
BETAg<-matrix(nrow=S,ncol=3)  
for(s in 1:S) {  
i<-sample(1:gs,1,prob=PB2)  
j<-sample(1:gs,1,prob=PB23[i,] )  
k<-sample(1:gs,1,prob=PB[,i,j] )  
BETAg[s,]<-c(beta1[k],beta2[i],beta3[j]) }

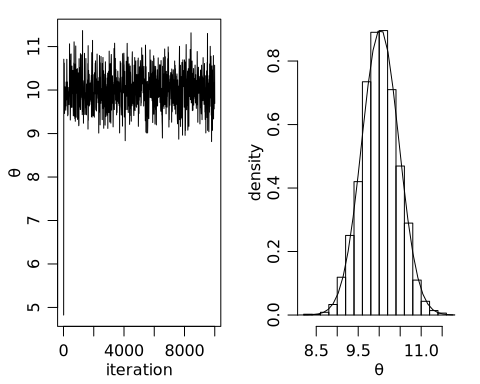
## 10.2 The Metropolis algorithm

#### Figure 10.2  
# pdf("fig10\_2.pdf",family="Times",height=1.75,width=5)  
par(mfrow=c(1,3),mar=c(2.75,2.75,.5,.5),mgp=c(1.7,.7,0))  
plot(beta2,PB2\*length(beta2)/(max(beta2)-min(beta2)) ,type="l",xlab=expression(beta[2]),ylab=expression(paste(italic("p("),beta[2],"|",italic("y)"),sep="") ) )  
plot(beta3,PB3\*length(beta3)/(max(beta3)-min(beta3)),type="l",xlab=expression(beta[3]),ylab=expression(paste(italic("p("),beta[3],"|",italic("y)"),sep="") ))  
  
Xs<-cbind(rep(1,6),1:6,(1:6)^2)  
eXB.post<- exp(t(Xs%\*%t(BETAg )) )  
qE<-apply( eXB.post,2,quantile,probs=c(.025,.5,.975))  
  
source("hdr2d.r")  
library(ash)  
plot.hdr2d(BETAg[,2:3],bw=c(15,15),xlab=expression(beta[2]),  
 ylab=expression(beta[3]))

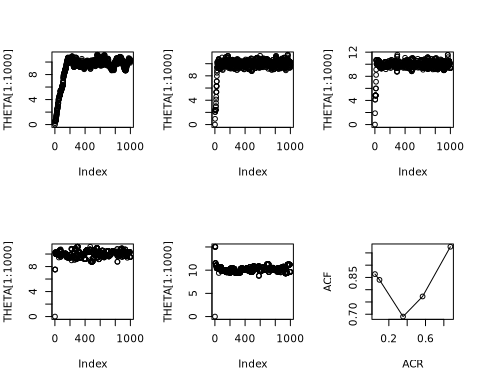


#### MH algorithm for one-sample normal problem with   
  
## Setup  
s2<-1   
t2<-10 ; mu<-5  
  
set.seed(1)  
n<-5  
y<-round(rnorm(n,10,1),2)  
  
mu.n<-( mean(y)\*n/s2 + mu/t2 )/( n/s2+1/t2)   
t2.n<-1/(n/s2+1/t2)  
  
## MCMC  
s2<-1 ; t2<-10 ; mu<-5   
y<-c(9.37, 10.18, 9.16, 11.60, 10.33)  
theta<-0 ; delta<-2 ; S<-10000 ; THETA<-NULL ; set.seed(1)  
  
for(s in 1:S)  
{  
  
 theta.star<-rnorm(1,theta,sqrt(delta))  
  
 log.r<-( sum(dnorm(y,theta.star,sqrt(s2),log=TRUE)) +  
 dnorm(theta.star,mu,sqrt(t2),log=TRUE) ) -  
 ( sum(dnorm(y,theta,sqrt(s2),log=TRUE)) +  
 dnorm(theta,mu,sqrt(t2),log=TRUE) )   
  
 if(log(runif(1))<log.r) { theta<-theta.star }  
  
 THETA<-c(THETA,theta)  
}

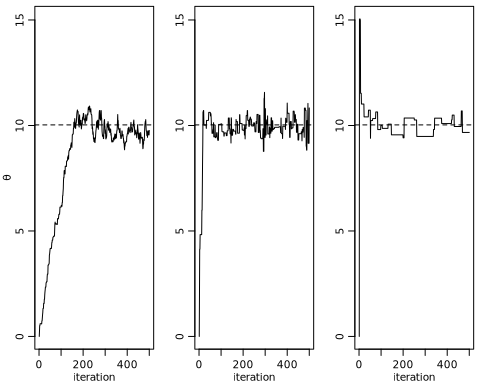
#### Figure 10.3  
# pdf("fig10\_3.pdf",family="Times",height=3.5,width=7)  
par(mar=c(3,3,1,1),mgp=c(1.75,.75,0))  
par(mfrow=c(1,2))  
  
skeep<-seq(10,S,by=10)  
plot(skeep,THETA[skeep],type="l",xlab="iteration",ylab=expression(theta))  
  
hist(THETA[-(1:50)],prob=TRUE,main="",xlab=expression(theta),ylab="density")  
th<-seq(min(THETA),max(THETA),length=100)  
lines(th,dnorm(th,mu.n,sqrt(t2.n)) )



#### MH algorithm with different proposal distributions  
par(mfrow=c(2,3))  
ACR<-ACF<-NULL  
THETAA<-NULL  
for(delta2 in 2^c(-5,-1,1,5,7) ) {  
set.seed(1)  
THETA<-NULL  
S<-10000  
theta<-0  
acs<-0  
delta<-2  
  
for(s in 1:S)   
{  
 theta.star<-rnorm(1,theta,sqrt(delta2))  
 log.r<-sum( dnorm(y,theta.star,sqrt(s2),log=TRUE)-  
 dnorm(y,theta,sqrt(s2),log=TRUE) ) +  
 dnorm(theta.star,mu,sqrt(t2),log=TRUE)-dnorm(theta,mu,sqrt(t2),log=TRUE)   
  
 if(log(runif(1))<log.r) { theta<-theta.star ; acs<-acs+1 }  
 THETA<-c(THETA,theta)   
  
}  
plot(THETA[1:1000])  
  
ACR<-c(ACR,acs/s)   
ACF<-c(ACF,acf(THETA,plot=FALSE)$acf[2] )  
THETAA<-cbind(THETAA,THETA)  
}  
plot(ACR,ACF) ; lines(ACR,ACF)



#### Figure 10.4   
# pdf("fig10\_4.pdf",family="Times",height=1.75,width=5)  
par(mfrow=c(1,3),mar=c(2.75,2.75,.5,.5),mgp=c(1.7,.7,0))  
laby<-c(expression(theta),"","","","")  
  
for(k in c(1,3,5)) {  
plot(THETAA[1:500,k],type="l",xlab="iteration",ylab=laby[k],   
 ylim=range(THETAA) )  
abline(h=mu.n,lty=2)  
 }



THCM<-apply(THETAA,2,cumsum)  
THCM<- THCM/(1:dim(THCM)[1])   
  
  
  
#### Back to sparrow data  
fit.mle<-glm(fledged~age+age2,family="poisson")  
summary(fit.mle)

##   
## Call:  
## glm(formula = fledged ~ age + age2, family = "poisson")  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.4650 -0.6355 -0.2298 0.4937 2.0429   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 0.27662 0.44219 0.626 0.5316   
## age 0.68174 0.33850 2.014 0.0440 \*  
## age2 -0.13451 0.05786 -2.325 0.0201 \*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for poisson family taken to be 1)  
##   
## Null deviance: 76.081 on 51 degrees of freedom  
## Residual deviance: 67.837 on 49 degrees of freedom  
## AIC: 198.78  
##   
## Number of Fisher Scoring iterations: 5

y<-fledged ; X<-cbind(rep(1,length(y)),age,age^2)  
yX<-cbind(y,X)  
colnames(yX)<-c("fledged","intercept","age","age2")   
  
n<-length(y) ; p<-dim(X)[2]  
  
pmn.beta<-rep(0,p)  
psd.beta<-rep(10,p)  
  
var.prop<- var(log(y+1/2))\*solve( t(X)%\*%X )  
beta<-rep(0,p)  
S<-10000  
BETA<-matrix(0,nrow=S,ncol=p)  
ac<-0  
set.seed(1)  
  
## rmvnorm function for proposals  
rmvnorm<-function(n,mu,Sigma)  
{ # samples from the multivariate normal distribution  
 E<-matrix(rnorm(n\*length(mu)),n,length(mu))  
 t( t(E%\*%chol(Sigma)) +c(mu))  
}  
  
## MCMC  
for(s in 1:S) {  
  
#propose a new beta  
  
beta.p<- t(rmvnorm(1, beta, var.prop ))  
  
lhr<- sum(dpois(y,exp(X%\*%beta.p),log=T)) -  
 sum(dpois(y,exp(X%\*%beta),log=T)) +  
 sum(dnorm(beta.p,pmn.beta,psd.beta,log=T)) -  
 sum(dnorm(beta,pmn.beta,psd.beta,log=T))  
  
if( log(runif(1))< lhr ) { beta<-beta.p ; ac<-ac+1 }  
  
BETA[s,]<-beta  
 }  
cat(ac/S,"\n")

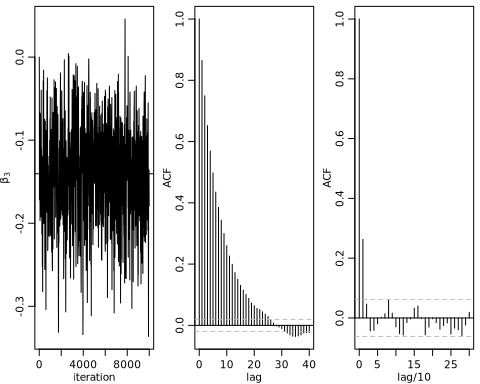
## 0.4293

library(coda)  
apply(BETA,2,effectiveSize)

## [1] 818.4049 778.4707 726.3633

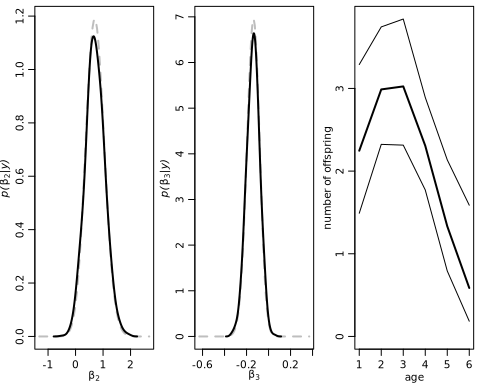
## 10.3 The Metropolis algorithm for Poisson regression

#### Figure 10.5   
# pdf("fig10\_5.pdf",family="Times",height=1.75,width=5)  
par(mar=c(2.75,2.75,.5,.5),mgp=c(1.7,.7,0))  
par(mfrow=c(1,3))  
blabs<-c(expression(beta[1]),expression(beta[2]),expression(beta[3]))  
thin<-c(1,(1:1000)\*(S/1000))  
j<-3  
plot(thin,BETA[thin,j],type="l",xlab="iteration",ylab=blabs[j])  
abline(h=mean(BETA[,j]) )  
  
acf(BETA[,j],ci.col="gray",xlab="lag")  
acf(BETA[thin,j],xlab="lag/10",ci.col="gray")



## 10.4 Metropolis, Metropolis-Hastings and Gibbs

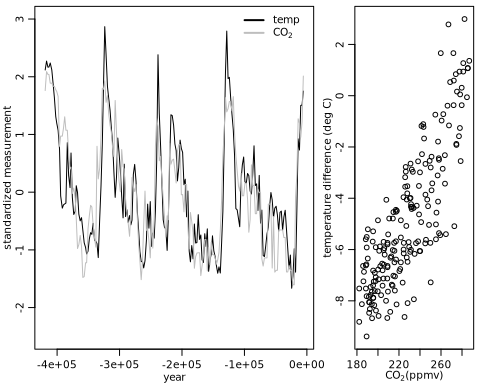
#### Figure 10.6  
# pdf("fig10\_6.pdf",family="Times",height=1.75,width=5)  
par(mar=c(2.75,2.75,.5,.5),mgp=c(1.7,.7,0))  
par(mfrow=c(1,3))  
  
plot(beta2,PB2\*length(beta2)/(max(beta2)-min(beta2)) ,type="l",xlab=expression(beta[2]),ylab=expression(paste(italic("p("),beta[2],"|",italic("y)"),sep="") ) ,lwd=2,lty=2,col="gray")  
lines(density(BETA[,2],adj=2),lwd=2)  
  
plot(beta3,PB3\*length(beta3)/(max(beta3)-min(beta3)),type="l",xlab=expression(beta[3]),ylab=expression(paste(italic("p("),beta[3],"|",italic("y)"),sep="") ),lwd=2,col="gray",lty=2)  
lines(density(BETA[,3],adj=2),lwd=2)  
  
Xs<-cbind(rep(1,6),1:6,(1:6)^2)   
eXB.post<- exp(t(Xs%\*%t(BETA )) )  
qE<-apply( eXB.post,2,quantile,probs=c(.025,.5,.975))  
  
plot( c(1,6),range(c(0,qE)),type="n",xlab="age",  
 ylab="number of offspring")  
lines( qE[1,],col="black",lwd=1)  
lines( qE[2,],col="black",lwd=2)  
lines( qE[3,],col="black",lwd=1)



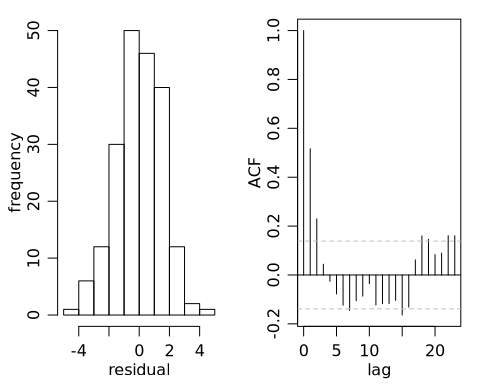
## 10.4 Metropolis, Metropolis-Hastings and Gibbs

Example: Historical CO2 and temperature data

#### Ice core example  
load("icecore.RData")   
  
# pdf("fig10\_7.pdf",family="Times",height=1.75,width=5)  
  
par(mar=c(2.75,2.75,.5,.5),mgp=c(1.7,.7,0))  
layout(matrix( c(1,1,2),nrow=1,ncol=3) )  
  
plot(icecore[,1], (icecore[,3]-mean(icecore[,3]))/sd(icecore[,3]) ,  
 type="l",col="black",  
 xlab="year",ylab="standardized measurement",ylim=c(-2.5,3))  
legend(-115000,3.2,legend=c("temp",expression(CO[2])),bty="n",  
 lwd=c(2,2),col=c("black","gray"))  
lines(icecore[,1], (icecore[,2]-mean(icecore[,2]))/sd(icecore[,2]),  
 type="l",col="gray")  
  
plot(icecore[,2], icecore[,3],xlab=expression(paste(CO[2],"(ppmv)")),ylab="temperature difference (deg C)")



#### Figure 10.8  
# pdf("fig10\_8.pdf",family="Times",height=3.5,width=7)  
par(mar=c(3,3,1,1),mgp=c(1.75,.75,0))  
par(mfrow=c(1,2))  
  
lmfit<-lm(icecore$tmp~icecore$co2)  
hist(lmfit$res,main="",xlab="residual",ylab="frequency")  
acf(lmfit$res,ci.col="gray",xlab="lag")



#### Starting values for MCMC  
n<-dim(icecore)[1]  
y<-icecore[,3]  
X<-cbind(rep(1,n),icecore[,2])  
DY<-abs(outer( (1:n),(1:n) ,"-"))  
  
lmfit<-lm(y~-1+X)  
beta<-lmfit$coef  
s2<-summary(lmfit)$sigma^2  
phi<-acf(lmfit$res,plot=FALSE)$acf[2]  
nu0<-1 ; s20<-1 ; T0<-diag(1/1000,nrow=2)  
  
  
## MCMC - 1000 scans saving every scan  
set.seed(1)  
S<-1000 ; odens<-S/5  
OUT<-NULL ; ac<-0 ; par(mfrow=c(1,2))  
for(s in 1:S)  
{  
  
 Cor<-phi^DY ; iCor<-solve(Cor)  
 V.beta<- solve( t(X)%\*%iCor%\*%X/s2 + T0)  
 E.beta<- V.beta%\*%( t(X)%\*%iCor%\*%y/s2 )  
 beta<-t(rmvnorm(1,E.beta,V.beta) )  
  
 s2<-1/rgamma(1,(nu0+n)/2,(nu0\*s20+t(y-X%\*%beta)%\*%iCor%\*%(y-X%\*%beta)) /2 )  
  
 phi.p<-abs(runif(1,phi-.1,phi+.1))  
 phi.p<- min( phi.p, 2-phi.p)  
 lr<- -.5\*( determinant(phi.p^DY,log=TRUE)$mod -  
 determinant(phi^DY,log=TRUE)$mod +  
 sum(diag( (y-X%\*%beta)%\*%t(y-X%\*%beta)%\*%(solve(phi.p^DY) -solve(phi^DY)) ) )/s2 )  
  
 if( log(runif(1)) < lr ) { phi<-phi.p ; ac<-ac+1 }  
  
 if(s%%odens==0)  
 {  
 cat(s,ac/s,beta,s2,phi,"\n") ; OUT<-rbind(OUT,c(beta,s2,phi))  
 }  
}

## 200 0.305 -9.895297 0.02147806 5.895741 0.878887   
## 400 0.3 -9.3182 0.0226351 6.040352 0.8576138   
## 600 0.315 -12.56879 0.03438303 4.589399 0.7856489   
## 800 0.32625 -12.35531 0.03255924 3.638383 0.7821035   
## 1000 0.322 -9.741086 0.02318024 4.732336 0.8357122

OUT.1000<-OUT  
library(coda)  
apply(OUT.1000,2,effectiveSize )

## [1] 5 5 5 5

## MCMC - 25000 scans saving every 25th scan  
set.seed(1)  
S<-25000 ; odens<-S/10  
OUT<-NULL ; ac<-0 ; par(mfrow=c(1,2))  
for(s in 1:S)  
{  
  
 Cor<-phi^DY ; iCor<-solve(Cor)  
 V.beta<- solve( t(X)%\*%iCor%\*%X/s2 + T0)  
 E.beta<- V.beta%\*%( t(X)%\*%iCor%\*%y/s2 )  
 beta<-t(rmvnorm(1,E.beta,V.beta) )  
  
 s2<-1/rgamma(1,(nu0+n)/2,(nu0\*s20+t(y-X%\*%beta)%\*%iCor%\*%(y-X%\*%beta)) /2 )  
  
 phi.p<-abs(runif(1,phi-.1,phi+.1))  
 phi.p<- min( phi.p, 2-phi.p)  
 lr<- -.5\*( determinant(phi.p^DY,log=TRUE)$mod -  
 determinant(phi^DY,log=TRUE)$mod +  
 sum(diag( (y-X%\*%beta)%\*%t(y-X%\*%beta)%\*%(solve(phi.p^DY) -solve(phi^DY)) ) )/s2 )  
  
 if( log(runif(1)) < lr ) { phi<-phi.p ; ac<-ac+1 }  
  
 if(s%%odens==0)  
 {  
 cat(s,ac/s,beta,s2,phi,"\n") ; OUT<-rbind(OUT,c(beta,s2,phi))  
 }  
}

## 2500 0.2892 -10.21004 0.02628094 5.250891 0.8574818   
## 5000 0.2754 -11.26445 0.02519835 4.547171 0.7997519   
## 7500 0.2812 -12.62779 0.03587274 3.875924 0.7210603   
## 10000 0.2734 -11.36281 0.02955854 4.203981 0.8091944   
## 12500 0.26744 -15.35529 0.04472815 3.48839 0.7874637   
## 15000 0.2669333 -8.681444 0.01881463 6.808671 0.8727924   
## 17500 0.2726857 -12.20131 0.0312813 5.625345 0.850547   
## 20000 0.27405 -11.46528 0.03096424 4.074319 0.8177968   
## 22500 0.2733778 -9.620781 0.02367434 3.265798 0.7566786   
## 25000 0.27632 -13.30138 0.03596505 3.927558 0.7798107

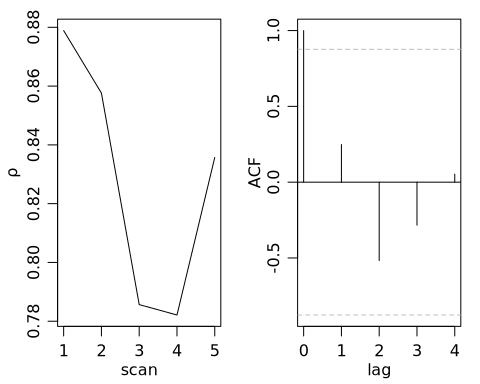
OUT.25000<-OUT  
library(coda)  
apply(OUT.25000,2,effectiveSize )

## [1] 27.60457 26.03660 10.00000 10.00000

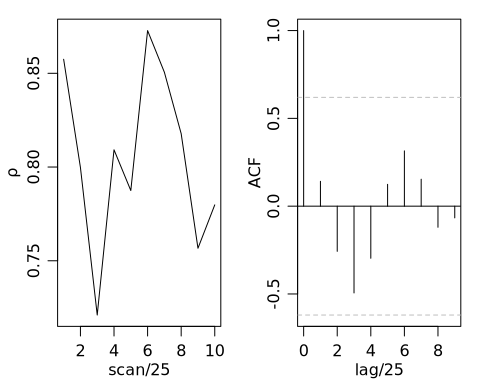
### 10.5.1 A regression model with correlated errors

### 10.5.2 Analysis of the ice core data

#### Figure 10.9  
# pdf("fig10\_9.pdf",family="Times",height=3.5,width=7)  
par(mar=c(3,3,1,1),mgp=c(1.75,.75,0))  
par(mfrow=c(1,2))  
plot(OUT.1000[,4],xlab="scan",ylab=expression(rho),type="l")  
acf(OUT.1000[,4],ci.col="gray",xlab="lag")

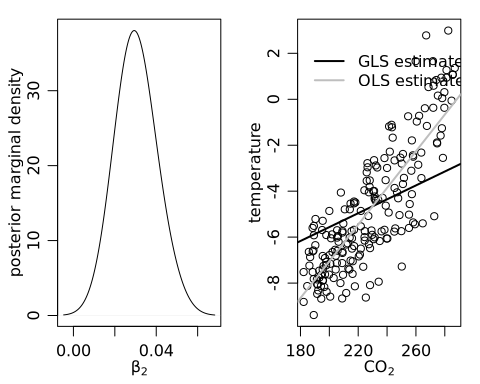


#### Figure 10.10  
# pdf("fig10\_10.pdf",family="Times",height=3.5,width=7)  
par(mar=c(3,3,1,1),mgp=c(1.75,.75,0))  
par(mfrow=c(1,2))  
plot(OUT.25000[,4],xlab="scan/25",ylab=expression(rho),type="l")  
acf(OUT.25000[,4],ci.col="gray",xlab="lag/25")



## 10.6 Discussion and further references

#### Figure 10.11  
# pdf("fig10\_11.pdf",family="Times",height=3.5,width=7)  
par(mar=c(3,3,1,1),mgp=c(1.75,.75,0))  
par(mfrow=c(1,2))  
  
plot(density(OUT.25000[,2],adj=2),xlab=expression(beta[2]),  
 ylab="posterior marginal density",main="")  
  
plot(y~X[,2],xlab=expression(CO[2]),ylab="temperature")  
abline(mean(OUT.25000[,1]),mean(OUT.25000[,2]),lwd=2)  
abline(lmfit$coef,col="gray",lwd=2)  
legend(180,2.5,legend=c("GLS estimate","OLS estimate"),bty="n",  
 lwd=c(2,2),col=c("black","gray"))



# Reference

Hoff, P. D. (2009). A first course in Bayesian statistical methods (Chapter 10). New York: Springer.