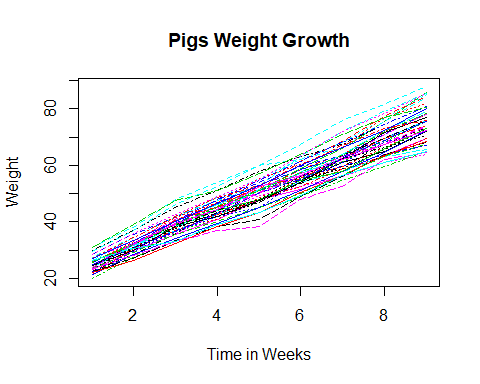
week1\_EDA\_Sample\_code.R

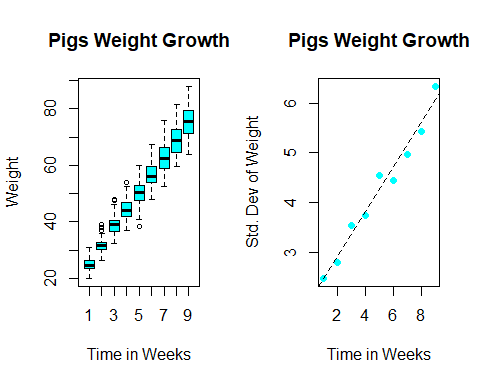
81983

2019-09-30

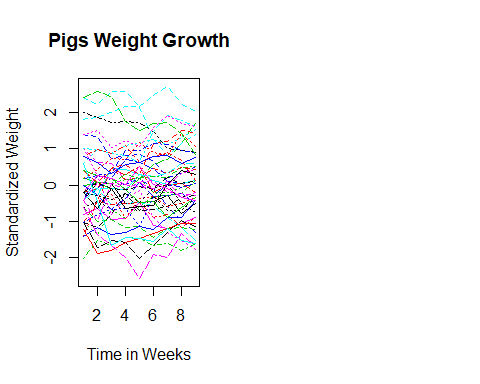
# exploring pig growth data  
pigs<-read.table("./PIGS.txt",sep="\t",header=TRUE)  
colnames(pigs)<-1:9  
matplot(t(pigs),type="l",xlab="Time in Weeks",ylab="Weight",lwd=1,main="Pigs Weight Growth")



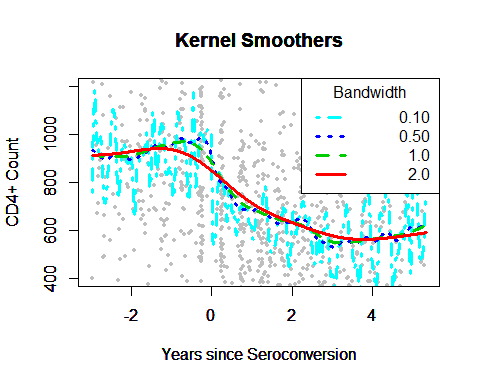
sd1<-apply(pigs,2,sd)  
pigs2<-data.frame(pigs)  
colnames(pigs2)<-1:9  
par(mfcol=c(1,2))  
boxplot(pigs2,xlab="Time in Weeks",ylab="Weight",col=5,main="Pigs Weight Growth",cex=0.75)  
plot(1:9,sd1,xlab="Time in Weeks",ylab="Std. Dev of Weight",main="Pigs Weight Growth",col=5,pch=19)  
abline(lm(sd1~c(1:9)),lty=2)



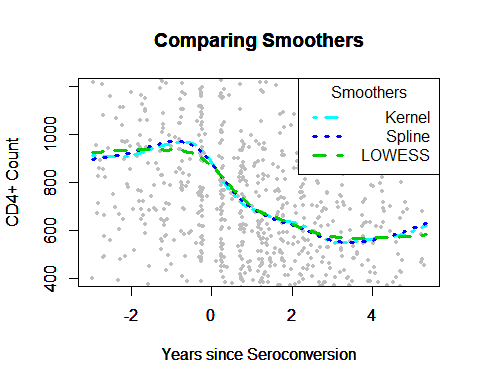
pigs3<-sweep(pigs,2,apply(pigs,2,mean))  
pigs3<-sweep(pigs3,2,sd1,FUN="/")  
matplot(t(pigs3),type="l",xlab="Time in Weeks",ylab="Standardized Weight",lwd=1,main="Pigs Weight Growth")  
pigs4<-as.vector(t(pigs))  
times<-rep(1:9,48)  
fit4<-loess.smooth(times,pigs4)  
fit5<-apply(pigs,2,mean)



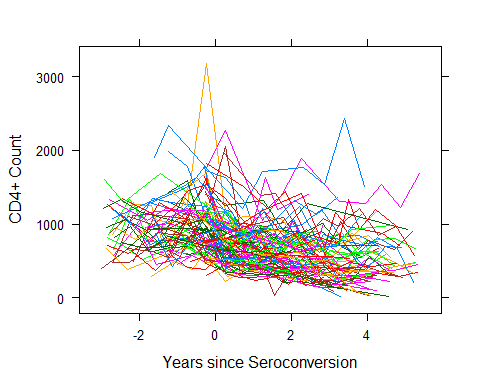
# CD4+ data exploration  
macs<-read.table("./MACS.txt",sep="\t",header=TRUE)  
# SMoothing kernel  
x<-macs$time  
y<-macs$cd4  
sid<-macs$id  
  
len<-sapply(split(x,sid),length)  
id<-which(len>=7)  
id<-sample(id,100,replace=FALSE)  
pid<-unique(sid)  
pid<-pid[id]  
pid<-sid%in%pid  
x<-x[pid]  
y<-y[pid]  
sid<-sid[pid]  
macs<-macs[pid,]  
  
  
k1<-ksmooth(x,y,kernel="normal",bandwidth=0.1)  
k2<-ksmooth(x,y,kernel="normal",bandwidth=0.5)  
k3<-ksmooth(x,y,kernel="normal",bandwidth=1.0)  
k4<-ksmooth(x,y,kernel="normal",bandwidth=2.0)  
ylims<-c(400,1200)  
xlims<-range(x)  
plot(x,y,pch=19,cex=0.5,xlim=xlims,ylim=ylims,col="gray",xlab="Years since Seroconversion",ylab="CD4+ Count",main="Kernel Smoothers")  
par(new=T)  
plot(k1,type="l",col=5,xlim=xlims,ylim=ylims,lwd=3,lty=4,xlab="Years since Seroconversion",ylab="CD4+ Count",main="Kernel Smoothers")  
par(new=T)  
plot(k2,type="l",col=4,xlim=xlims,ylim=ylims,lwd=3,lty=3,xlab="Years since Seroconversion",ylab="CD4+ Count",main="Kernel Smoothers")  
par(new=T)  
plot(k3,type="l",col=3,xlim=xlims,ylim=ylims,lwd=3,lty=2,xlab="Years since Seroconversion",ylab="CD4+ Count",main="Kernel Smoothers")  
par(new=T)  
plot(k4,type="l",col=2,xlim=xlims,ylim=ylims,lwd=3,lty=1,xlab="Years since Seroconversion",ylab="CD4+ Count",main="Kernel Smoothers")  
temp <- legend("topright", legend = c(" ", " "," "," "),  
 text.width = strwidth("Bandwidth"),  
 lty = c(4,3,2,1), col=c(5,4,3,2), lwd=3,xjust = 1, yjust = 1,  
 title = "Bandwidth")  
text(temp$rect$left + temp$rect$w, temp$text$y,  
 c("0.10", "0.50","1.0","2.0"), pos=2)



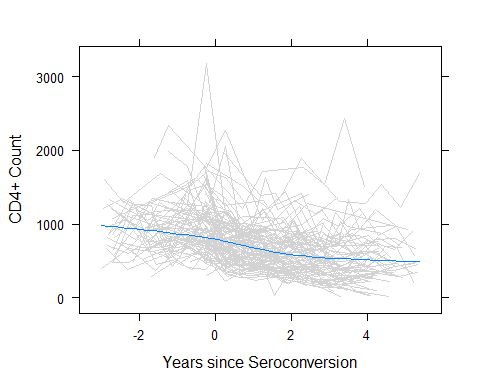
# comparing smoothers  
k1<-ksmooth(x,y,kernel="normal",bandwidth=1.0)  
s1<-smooth.spline(x,y)  
l1<-loess.smooth(x,y,span=0.5,family="gaussian",evaluation=length(x))  
ylims<-c(400,1200)  
xlims<-range(x)  
plot(x,y,pch=19,cex=0.5,xlim=xlims,ylim=ylims,col="gray",xlab="Years since Seroconversion",ylab="CD4+ Count",main="Comparing Smoothers")  
par(new=T)  
plot(k1,type="l",col=5,xlim=xlims,ylim=ylims,lwd=3,lty=4,xlab="Years since Seroconversion",ylab="CD4+ Count",main="Comparing Smoothers")  
par(new=T)  
plot(s1,type="l",col=4,xlim=xlims,ylim=ylims,lwd=3,lty=3,xlab="Years since Seroconversion",ylab="CD4+ Count",main="Comparing Smoothers")  
par(new=T)  
plot(l1,type="l",col=3,xlim=xlims,ylim=ylims,lwd=3,lty=2,xlab="Years since Seroconversion",ylab="CD4+ Count",main="Comparing Smoothers")  
temp <- legend("topright", legend = c(" ", " "," "),  
 text.width = strwidth("Smoothers"),  
 lty = c(4,3,2), col=c(5,4,3), lwd=3,xjust = 1, yjust = 1,  
 title = "Smoothers")  
text(temp$rect$left + temp$rect$w, temp$text$y,  
 c("Kernel", "Spline","LOWESS"), pos=2)  
# Time plot / profile plot  
library(lattice)



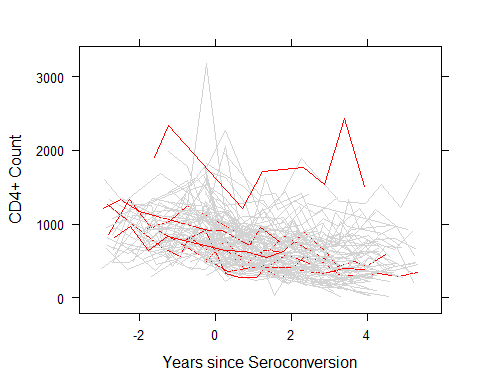
xyplot(cd4~time,groups=id,data=macs,type="l",xlab="Years since Seroconversion",ylab="CD4+ Count")



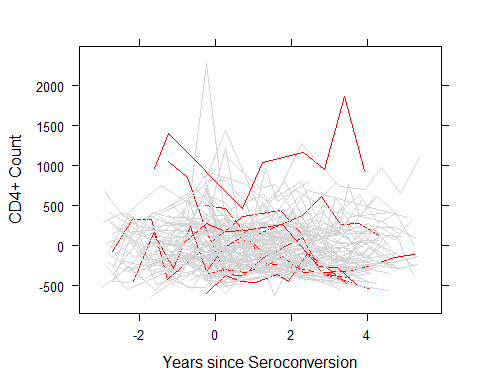
xyplot(cd4~time,groups=id,data=macs,type="l",xlab="Years since Seroconversion",ylab="CD4+ Count",col="lightgray",  
 panel=function(...){  
 panel.xyplot(...)  
 panel.loess(x,y)  
 }  
)



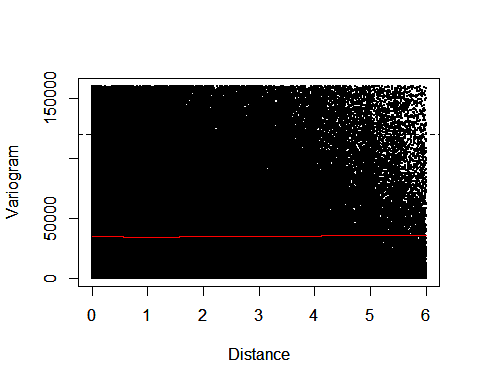
len<-length(unique(macs$id))  
id2<-rbinom(len,1,p=0.05)  
id2[id2==0]<-"lightgray"  
id2[id2==1]<-"red"  
xyplot(cd4~time,groups=id,data=macs,type="l",xlab="Years since Seroconversion",ylab="CD4+ Count",col=id2)



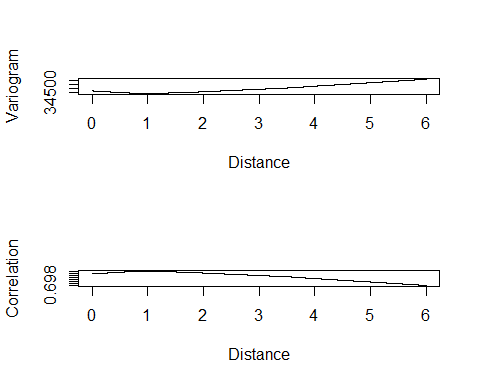
# PLotting residuals  
ind<-sapply(x,function(z){order(abs(z-l1$x))[1]})  
fit<-l1$y[ind]  
res<-y-fit  
res2<-split(res,sid)  
id<-NULL  
z1<-sapply(res2,median)  
for(i in c(0,0.05,0.1,0.25,0.5,0.75,0.90,0.95,1)){  
 z2<-order(abs(z1-quantile(z1,i)))[1]  
 id<-c(id,z2)  
}  
len<-length(unique(macs$id))  
id2<-rep(0,len)  
id2[id]<-1  
id2[id2==0]<-"lightgray"  
id2[id2==1]<-"red"  
macs2<-cbind(macs,res)  
xyplot(res~time,groups=id,data=macs2,type="l",lwd=0.5,xlab="Years since Seroconversion",ylab="CD4+ Count",col=id2)



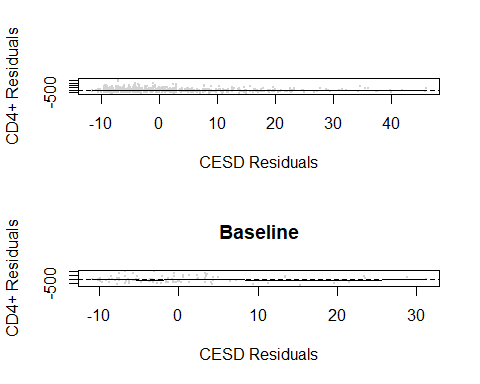
# variogram  
library(nlme)  
u<-lapply(1:(length(x)-1),function(i){abs(x[i]-x[-c(1:i)])})  
gam1<-Variogram(res,unlist(u))  
gam2<-gam1$variog  
u2<-gam1$dist  
id<-which(u2<=6)  
u2<-u2[id]  
gam2<-gam2[id]  
id<-which(gam2<=160000)  
gam2<-gam2[id]  
u2<-u2[id]  
ylims=range(gam2)  
xlims<-range(u2)  
junk<-loess.smooth(u2,gam2,evaluation=length(u2),span=0.5)  
sid<-macs$id  
sigma<-lapply(1:(length(y)-1),function(i){res[i]-res[sid!=sid[i]]})  
sigma<-mean(((unlist(sigma))^2)/2)  
plot(u2,gam2,cex=0.001,pch=19,ylim=ylims,xlim=xlims,xlab="Distance",ylab="Variogram")  
par(new=T)  
plot(junk,ylim=ylims,xlim=xlims,type="l",col=2,xlab="Distance",ylab="Variogram")  
abline(h=sigma,lty=2)



rho<-1-(junk$y/sigma)  
par(mfcol=c(2,1))  
plot(junk,type="l",xlab="Distance",ylab="Variogram")  
plot(junk$x,rho,type="l",xlab="Distance",ylab="Correlation")



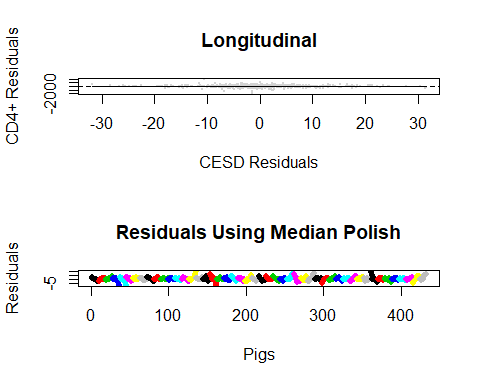
# CESD  
y2<-macs$cesd  
l2<-loess.smooth(x,y2,span=0.5,family="gaussian",evaluation=length(x))  
ind<-sapply(x,function(z){order(abs(z-l2$x))[1]})  
fit2<-l2$y[ind]  
res2<-y2-fit2  
scatter.smooth(res2,res,pch=19,cex=0.2,col="lightgray",xlab="CESD Residuals",ylab="CD4+ Residuals")  
abline(h=0,lty=2)  
# Seperate baseline and longitudinal effects  
x4<-split(macs$time,macs$id)  
y41<-split(macs$cd4,macs$id)  
y42<-split(macs$cesd,macs$id)  
# baseline effects  
x4b<-sapply(x4,function(z){z[1]})  
y41b<-sapply(y41,function(z){z[1]})  
y42b<-sapply(y42,function(z){z[1]})  
l41<-loess.smooth(x4b,y41b,span=0.5,family="gaussian",evaluation=length(x4b))  
l42<-loess.smooth(x4b,y42b,span=0.5,family="gaussian",evaluation=length(x4b))  
ind<-sapply(x4b,function(z){order(abs(z-l41$x))[1]})  
fit41<-l41$y[ind]  
res41<-y41b-fit41  
ind<-sapply(x4b,function(z){order(abs(z-l42$x))[1]})  
fit42<-l42$y[ind]  
res42<-y42b-fit42  
scatter.smooth(res42,res41,pch=19,cex=0.2,col="lightgray",xlab="CESD Residuals",ylab="CD4+ Residuals",main="Baseline")  
abline(h=0,lty=2)



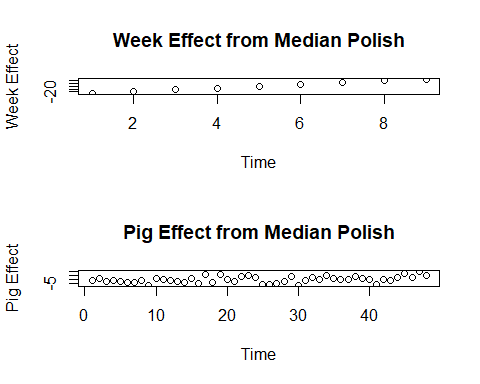
# longitudinal effects  
x4b<-unlist(sapply(x4,function(z){diff(z)}))  
y41b<-unlist(sapply(y41,function(z){diff(z)}))  
y42b<-unlist(sapply(y42,function(z){diff(z)}))  
l41<-loess.smooth(x4b,y41b,span=0.5,family="gaussian",evaluation=length(x4b))  
l42<-loess.smooth(x4b,y42b,span=0.5,family="gaussian",evaluation=length(x4b))  
ind<-sapply(x4b,function(z){order(abs(z-l41$x))[1]})  
fit41<-l41$y[ind]  
res41<-y41b-fit41  
ind<-sapply(x4b,function(z){order(abs(z-l42$x))[1]})  
fit42<-l42$y[ind]  
res42<-y42b-fit42  
scatter.smooth(res42,res41,pch=19,cex=0.2,col="lightgray",xlab="CESD Residuals",ylab="CD4+ Residuals",main="Longitudinal")  
abline(h=0,lty=2)  
  
# Exploring correlation  
pigs<-read.table("PIGS.txt",sep="\t",header=TRUE)  
colnames(pigs)<-1:9  
junk1<-medpolish(pigs)

## 1: 736.5  
## 2: 591  
## Final: 591

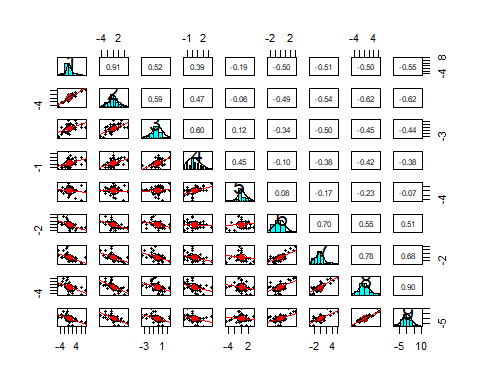
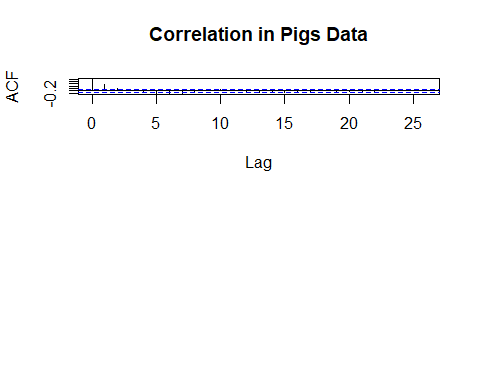
res<-junk1$res  
cols<-rep(1:48,rep(9,48))  
plot(as.vector(t(res)),col=cols,pch=19,cex=0.8,xlab="Pigs",ylab="Residuals",main="Residuals Using Median Polish")



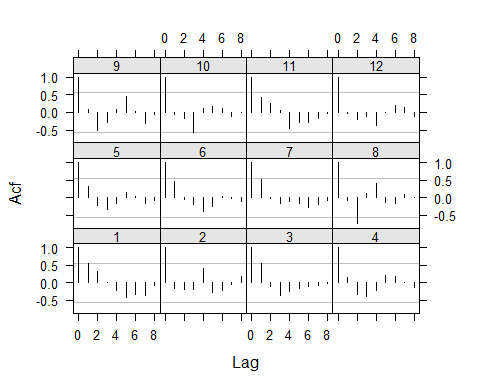
plot(junk1$col,xlab="Time",ylab="Week Effect",main="Week Effect from Median Polish")  
plot(junk1$row,xlab="Time",ylab="Pig Effect",main="Pig Effect from Median Polish")



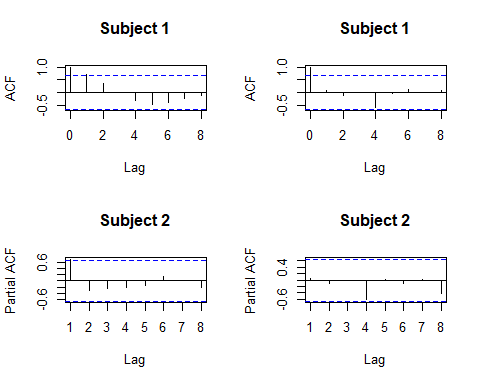
acf(as.vector(t(res)),main="Correlation in Pigs Data")  
library(psych)  
pairs.panels(res,lm=TRUE)



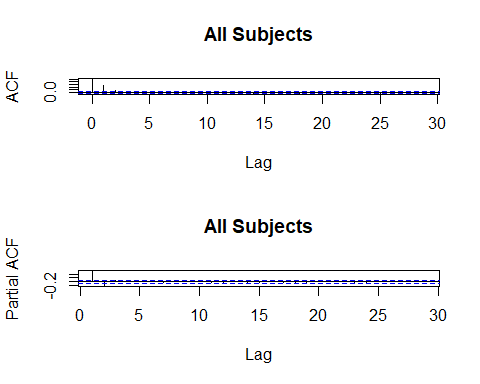
mu<-apply(pigs,2,mean)  
res<-sweep(pigs,2,mu)  
cols<-rep(1:48,rep(9,48))  
plot(as.vector(t(res)),col=cols,pch=19,cex=0.8,xlab="Pigs",ylab="Residuals",main="Residuals Using Simple Mean")  
  
  
# ACF  
library(languageR)  
a<-t(sapply(1:100,function(z){arima.sim(9,model=list(ar=c(0.8,-0.3)))}))  
b<-as.vector(t(a))  
id<-rep(1:100,rep(9,100))  
times<-rep(1:9,100)  
zz<-cbind(times,b,id)  
zz<-as.data.frame(zz)  
zz2<-zz[1:(12\*9),]  
zz2[,3]<-as.factor(zz2[,3])  
zz[,3]<-as.factor(zz[,3])  
colnames(zz)<-c("times","b","id")  
xyplot(b~times|id,data=zz[1:(12\*9),],pch=20,xlab="Time",ylab="Residuals")  
acf.fnc(zz2, group="id",time="times",x="b")  
  
par(mfcol=c(2,2))



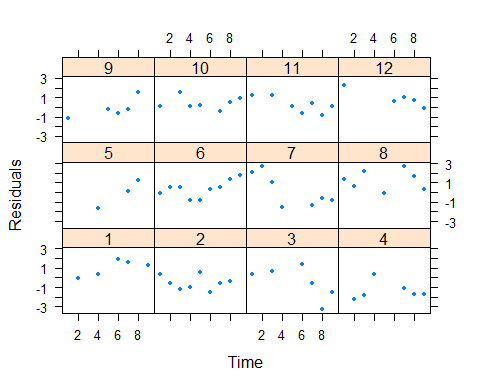
id<-sample(1:100,2)  
for(i in 1:2){  
 acf(a[id[i],],main="Subject 1")  
 pacf(a[id[i],],main="Subject 2")  
}



par(mfcol=c(2,1))  
acf(b,main="All Subjects")  
pacf(b,main="All Subjects")



#variogram  
id<-as.logical(rbinom(900,1,p=2/3))  
naid<-matrix(id,ncol=9,byrow=FALSE)  
b2<-a  
b2[!naid]<-NA  
b3<-as.vector(t(b2))  
id<-rep(1:100,rep(9,100))  
zz<-cbind(times,b3,id)  
zz<-as.data.frame(zz)  
zz[,3]<-as.factor(zz[,3])  
colnames(zz)<-c("times","b","id")  
xyplot(b~times|id,data=zz[1:(12\*9),],pch=20,xlab="Time",ylab="Residuals")



library(nlme)  
ind<-which(!is.na(b3))  
times<-times[ind]  
b3<-b3[ind]  
u<-lapply(1:(length(times)-1),function(i){abs(times[i]-times[-c(1:i)])})  
gam1<-Variogram(b3,unlist(u))  
gam2<-gam1$variog  
u2<-gam1$dist  
ylims=range(gam2)  
xlims<-range(u2)  
junk<-loess.smooth(u2,gam2,evaluation=length(u2),span=0.25)

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =  
## FALSE, : pseudoinverse used at 2

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =  
## FALSE, : neighborhood radius 1

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =  
## FALSE, : reciprocal condition number 0

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =  
## FALSE, : There are other near singularities as well. 1

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =  
## FALSE, : pseudoinverse used at -0.04

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =  
## FALSE, : neighborhood radius 1.04

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =  
## FALSE, : reciprocal condition number 3.6147e-015

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =  
## FALSE, : There are other near singularities as well. 1

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =  
## FALSE, : pseudoinverse used at -0.04

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =  
## FALSE, : neighborhood radius 1.04

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =  
## FALSE, : reciprocal condition number 6.4922e-016

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =  
## FALSE, : There are other near singularities as well. 1

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =  
## FALSE, : pseudoinverse used at -0.04

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =  
## FALSE, : neighborhood radius 1.04

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =  
## FALSE, : reciprocal condition number 5.874e-016

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =  
## FALSE, : There are other near singularities as well. 1

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =  
## FALSE, : pseudoinverse used at 2

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =  
## FALSE, : neighborhood radius 1

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =  
## FALSE, : reciprocal condition number 0

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =  
## FALSE, : There are other near singularities as well. 1

sid<-id[ind]  
sigma<-lapply(1:(length(b3)-1),function(i){b3[i]-b3[sid!=sid[i]]})  
sigma<-mean(((unlist(sigma))^2)/2)  
junk<-split(gam2,u2)  
gam2<-sapply(junk,median)  
par(mfcol=c(2,1))  
plot(0:8,gam2,pch=20,xlab="Distance",ylab="Variogram",type="b")  
abline(h=sigma,lty=2)  
rho2<-1-(gam2/sigma)  
plot(0:8,rho2,pch=20,xlab="Distance",ylab="Correlation",type="b")

