#######basic dealing with data

usedata=cbind(age,wcmatrix1,fnlwgt,edumatrix,edunum,msmatrix,ocmatrix1,rsmatrix,rcmatrix,sexmatrix,capitalgain,capitalloss,hoursperweek,ncmatrix1,salary)

n=dim(usedata)[1]

m=dim(usedata)[2]

#1 stands for <=50K, 2 stands for >50

#delete the responce variable "salary"

df=data.frame(usedata)

traindata=df[(1:(n/2)),-m]

testdata=df[(n/2+1):n,-m]

#to test whether there are outliers

require("outliers")

chisq.out.test(fnlwgt, variance=var(fnlwgt), opposite = FALSE)

chisq.out.test(age, variance=var(age), opposite = FALSE)

#OK, no outlier nor missing data nor "?"of fnlwgt and age

##########pca############################################to centralize the traindata first

traindata\_centered=traindata-matrix(rep(colMeans(traindata),nrow(traindata)),byrow=T)

pca=prcomp(traindata\_centered,scale.=TRUE)

summary(pca)

pca.var=pca$sdev^2

pca.pervar=pca.var/sum(pca.var)

plot(pca.pervar , xlab=" Principal Component ", ylab=" Proportion of

Variance Explained ",xlim=c(0,60),ylim=c(0,0.04),type="line")

loadings=pca$rotation[,1:33]

scale=as.matrix(testdata)%\*%loadings

> pca.lr=lm(usedata[((n/2)+1):n,m]~scale)

> truetest=usedata[((n/2)+1):n,m]

> pca.predict.test=predict(pca.lr,testdata)

> pca.predict.test[which(pca.predict.test<=1.5)]=1

> pca.predict.test[which(pca.predict.test>1.5)]=2

> pca.misrate.test=sum(abs(pca.predict.test-truetest))/length(truetest)

> pca.misrate.test

[1] 0.1720708

> ###how about the mis-classified rate for traingset?

> pca.lr=lm(usedata[(1:(n/2)+1),m]~scale)

> truetest=usedata[(1:(n/2)+1),m]

> pca.predict.train=predict(pca.lr,traindata)

> pca.predict.train[which(pca.predict.train<=1.5)]=1

> pca.predict.train[which(pca.predict.train>1.5)]=2

> pca.misrate.train=sum(abs(pca.predict.train-truetest))/length(truetest)

> pca.misrate.train

[1] 0.2471985